سیستم‌های چندعاملی

درس ۲۸

طرح‌ریزی چندعاملی

Multiagent Planning

کاظم فولادی قلعه
دانشکده مهندسی، پردیس فارابی
دانشگاه تهران

http://courses.fouladi.ir/mas

طرح‌ریزی

PLANNING

طرح‌ریزی *Planning*	مسئله‌ی یافتن **دنباله‌ای از کنش‌ها** برای دست‌یابی به یک **هدف**

طرح‌ریزی چندعاملی

MULTIAGENT PLANNING

در طرح‌ریزی کلاسیک، فرض می‌شد که
تنها یک عامل وجود دارد که حس، طرح‌ریزی و کنش می‌کند.

وقتی چند عامل در محیط وجود دارد،
هر عامل با یک **مسئله‌ی طرح‌ریزی چندعاملی** مواجه می‌شود.

طرح‌ریزی چندعاملی

MULTIAGENT PLANNING

طرح‌ریزی چندعاملی
Multiagent Planning

روی‌کرد متمرکز
Centralized

پلان هر یک از عامل‌ها،
توسط یک هماهنگ‌کننده‌ی مرکزی تحلیل می‌شود
تا تعاملات لازم را شناسایی کند.

پلان متمرکز *Centralized Plan*	پلان توزیع‌شده *Distributed Plan*
پلان باید توسط یک عامل **اجرا** شود.	پلان باید توسط گروهی از عامل‌ها **اجرا** شود.
طرح‌ریزی کلاسیک تک‌عاملی	

روی‌کرد توزیع‌شده
Distributed

گروهی از عامل‌ها برای تهیه‌ی پلان
با هم همکاری می‌کنند.

پلان متمرکز *Centralized Plan*	پلان توزیع‌شده *Distributed Plan*
پلان باید توسط یک عامل **اجرا** شود.	پلان باید توسط گروهی از عامل‌ها **اجرا** شود.

طرح‌ریزی چندعاملی

طرح‌ریزی توزیع‌شده برای پلان‌های متمرکز

DISTRIBUTED PLANNING FOR CENTRALIZED PLANS

طرح‌ریزی متمرکز برای پلان‌های متمرکز
Centralized Planning for Centralized Plans

مثل کاربرد در حوزه‌ی کنترل ترافیک هوایی

هدف: قادر کردن هر هواپیما برای نگهداری یک پلان پرواز
که یک فاصله‌ی ایمن را از همه‌ی هواپیماها در همسایگی آن حفظ می‌کند.

هر هواپیما اطلاعاتی را در مورد کنش‌های از پیش‌تعیین شده‌ی خود به هماهنگ‌کننده‌ی مرکزی می‌فرستد.
هماهنگ کننده پلانی را ایجاد می‌کند که کنش همه‌ی عامل‌ها را تعیین می‌کند؛
که از جمله شامل این است که آنها برای اجتناب از برخورد باید چه کاری انجام دهند.

طرح‌ریزی چندعاملی

طرح‌ریزی متمرکز برای پلان‌های توزیع‌شده

CENTRALIZED PLANNING FOR DISTRIBUTED PLANS

طرح‌ریزی متمرکز برای پلان‌های توزیع‌شده
Centralized Planning for Distributed Plans

- ۱) با داشتن توصیف هدف، مجموعه‌ای از عملگرها و یک توصیف حالت اولیه، یک پلان ترتیب جزئی تولید می‌کنیم.
- ۲) پلان حاصل را به زیرمسئله‌هایی تجزیه می‌کنیم به‌گونه‌ای که روابط ترتیب بین پلان‌ها می‌نیمم شود.
- ۳) زیرپلان‌ها را به عامل‌ها تخصیص می‌دهیم.
- ۴) اگر تخصیص زیرپلان‌ها به شکست انجامید، تجزیه را دوباره انجام می‌دهیم (گام ۲)، اگر تخصیص زیرپلان‌ها با همه‌ی تجزیه‌ها شکست خورد، پلان دیگری تولید می‌کنیم (گام ۱).
- ۵) زیرپلان‌ها را اجرا می‌کنیم.

طرح‌ریزی چندعاملی

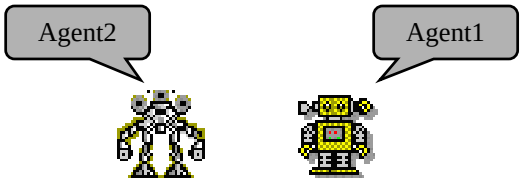
طرح‌ریزی متمرکز برای پلان‌های توزیع‌شده: مثال (۱ از ۴)

CENTRALIZED PLANNING FOR DISTRIBUTED PLANS

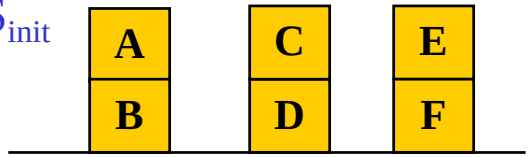
move(b,x,y) // move b from x to y
PRECOND: on(b,x) ∧ clear(b)
POSTCOND: on(b,y) ∧ clear(x) ∧ ¬on(b,x) ∧ ¬clear(y)

movetotable(b,x)
PRECOND: on(b,x) ∧ clear(b)
POSTCOND: on(b,T) ∧ clear(x) ∧ ¬on(b,x)

work backward from each "on" goal:



S_{init}



on(A,B) on(C,D) on(E,F)
on(B,T) on(D,T) on(F,T)

on(B,A)

S1: move(B,T,A)

on(B,T) clear(B) clear(A)

movetotable(A,B) move(A,B,y)

S2: move(A,B,E)

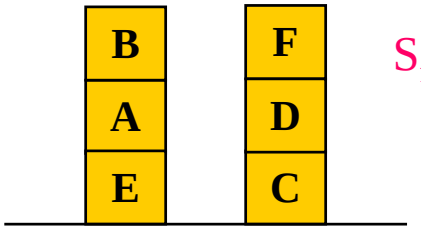
clear(A) clear(E) on(A,B)

……..

……

on(E,T)

S3: movetotable(E,F)



S_{final}

on(B,A) on(F,D)
on(A,E) on(D,C)
on(E,T) on(C,T)

طرح‌ریزی چندعاملی

طرح‌ریزی متمرکز برای پلان‌های توزیع‌شده: مثال (۲ از ۴)

CENTRALIZED PLANNING FOR DISTRIBUTED PLANS

S1: **move**(B,T,A)
S2: **move**(A,B,E)
S3: **movetotable**(E,F)
S4: **move**(F,T,D)
S5: **move**(D,T,C)
S6: **movetotable**(C,D)

To satisfy the preconditions, we have:

S2 < S1, S3 < S4
S6 < S4, S6 < S5

Also

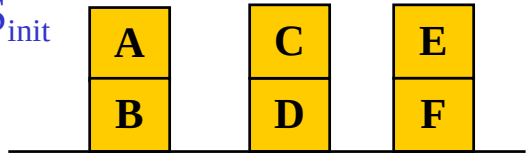
S2 threat to S3 ⇒ S3 < S2
S4 threat to S5 ⇒ S5 < S4

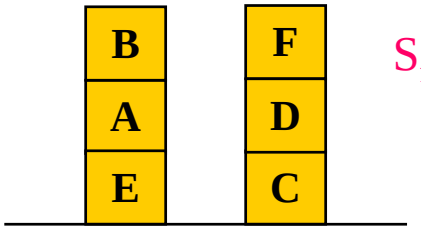
Then the **partial ordering** is:

S3 < S2 < S1
S6 < S5 < S4
S3 < S4

Any total ordering that satisfies this partial ordering is a good plan for Agent1

S_{init}





S_{final}

طرح‌ریزی چندعاملی

طرح‌ریزی متمرکز برای پلان‌های توزیع‌شده: مثال (۳ از ۴)

CENTRALIZED PLANNING FOR DISTRIBUTED PLANS

وقتی دو عامل داریم:

DECOMP1

SUBPLAN1: S3 < S2 < S1

SUBPLAN2: S6 < S5 < S4

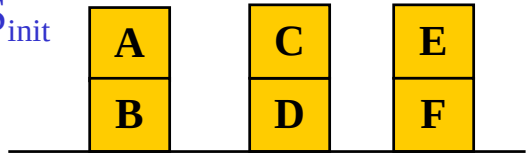
and S3 < S4

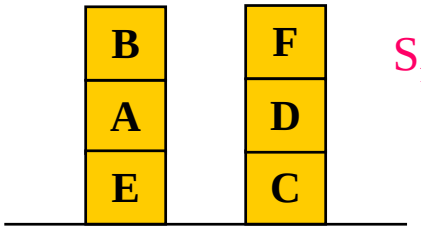
Agent1: S3 < *send*(clear(F)) < S2 < S1

Agent2: S6 < S5 < *wait*(clear(F)) < S4

{ S3: **movetotable**(E,F), S6: **movetotable**(C,D) } < { S2: **move**(A,B,E), S5: **move**(D,T,C) } < { S1: **move**(B,T,A), S4: **move**(F,T,D) }

S_{init}





S_{final}

طرح‌ریزی چندعاملی

طرح‌ریزی متمرکز برای پلان‌های توزیع‌شده: مثال (۴ از ۴)

CENTRALIZED PLANNING FOR DISTRIBUTED PLANS

وقتی دو عامل داریم:

DECOMP2

SUBPLAN1: S3 < S5 < S4

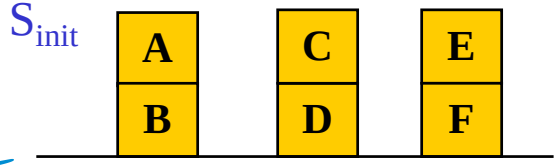
SUBPLAN2: S6 < S2 < S1

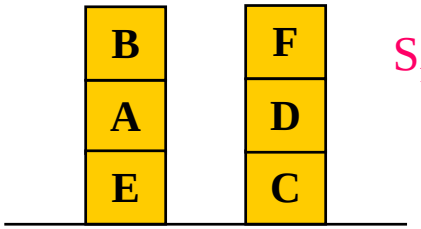
and S2 < S2 and S6 < S5

Agent1: S3 < *send*(don't_care(E)) < *wait*(clear(D)) < S5 < S4

Agent2: S6 < *send*(don't_care(E)) < *wait*(clear(F)) < S2 < S1

در این مثال، تجزیه‌ی ۲ تعداد روابط ترتیب بیشتری نسبت به تجزیه‌ی ۱ دارد،
پس تجزیه‌ی ۱ انتخاب می‌شود.





طرح‌ریزی چندعاملی

طرح‌ریزی توزیع‌شده برای پلان‌های متمرکز

DISTRIBUTED PLANNING FOR CENTRALIZED PLANS

طرح‌ریزی توزیع‌شده برای پلان‌های متمرکز
Distributed Planning for Centralized Plans

پلان‌های جداگانه‌ای را تولید می‌کنیم و آنها را با هم ادغام می‌کنیم

(تسهیم موازی نتیجه‌ها)
* ممکن است به مذاکره نیاز داشته باشیم.

طرح‌ریزی چندعاملی

طرح‌ریزی توزیع‌شده برای پلان‌های متمرکز: مثال

DISTRIBUTED PLANNING FOR CENTRALIZED PLANS

Agent 1 - is specialized in doing **movetotable**(b,x)

Agent 2 - is specialized in doing **move**(b,x,y)

P_{Agent1} = { S3: **movetotable**(E,F) satisfies **on**(E,T)
S6: **movetotable**(C,D) satisfies **on**(C,T)
no ordering }

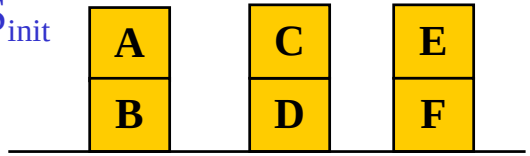
$P_{Agent\ 2}$ = { S1: **move**(B,T,A), S2: **move**(A,B,E) satisfies **on**(B,A) ∧ **on**(A,E)
S4: **move**(F,T,D), S5: **move**(D,T,C) satisfies **on**(F,D) ∧ **on**(D,C)
ordering S2 < S1 and S5 < S4 }

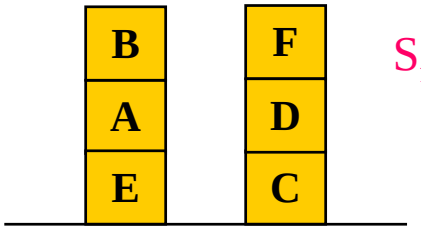
Merge P_{Agent1} with P_{Agent2} by checking preconditions and threats

S3 < S2, S6 < S5, S3 < S4, S2 < S1 and S5 < S4

one agent executes (as is centralized)

S_{init}





S_{final}

طرح‌ریزی چندعاملی

طرح‌ریزی توزیع‌شده برای پلان‌های توزیع‌شده

DISTRIBUTED PLANNING FOR DISTRIBUTED PLANS

طرح‌ریزی توزیع‌شده برای پلان‌های توزیع‌شده
Distributed Planning for Distributed Plans

پلان‌های تک تک عامل‌ها، به‌صورت پویا هماهنگ می‌شود.

هیچ فردی، دید کاملی از کنش‌های همه‌ی عامل‌ها ندارد.

* تشخیص و رفع اندرکنش‌های نامطلوب دشوارتر است.

رویکردها		
ادغام پلان‌ها *Plan Merging*	**تشکیل تکراری پلان‌ها** *Iterative Plan Formation*	**طرح‌ریزی توزیع‌شده‌ی سلسله‌مراتبی** *Hierarchical Distributed planning*

طرح‌ریزی چندعاملی

طیف

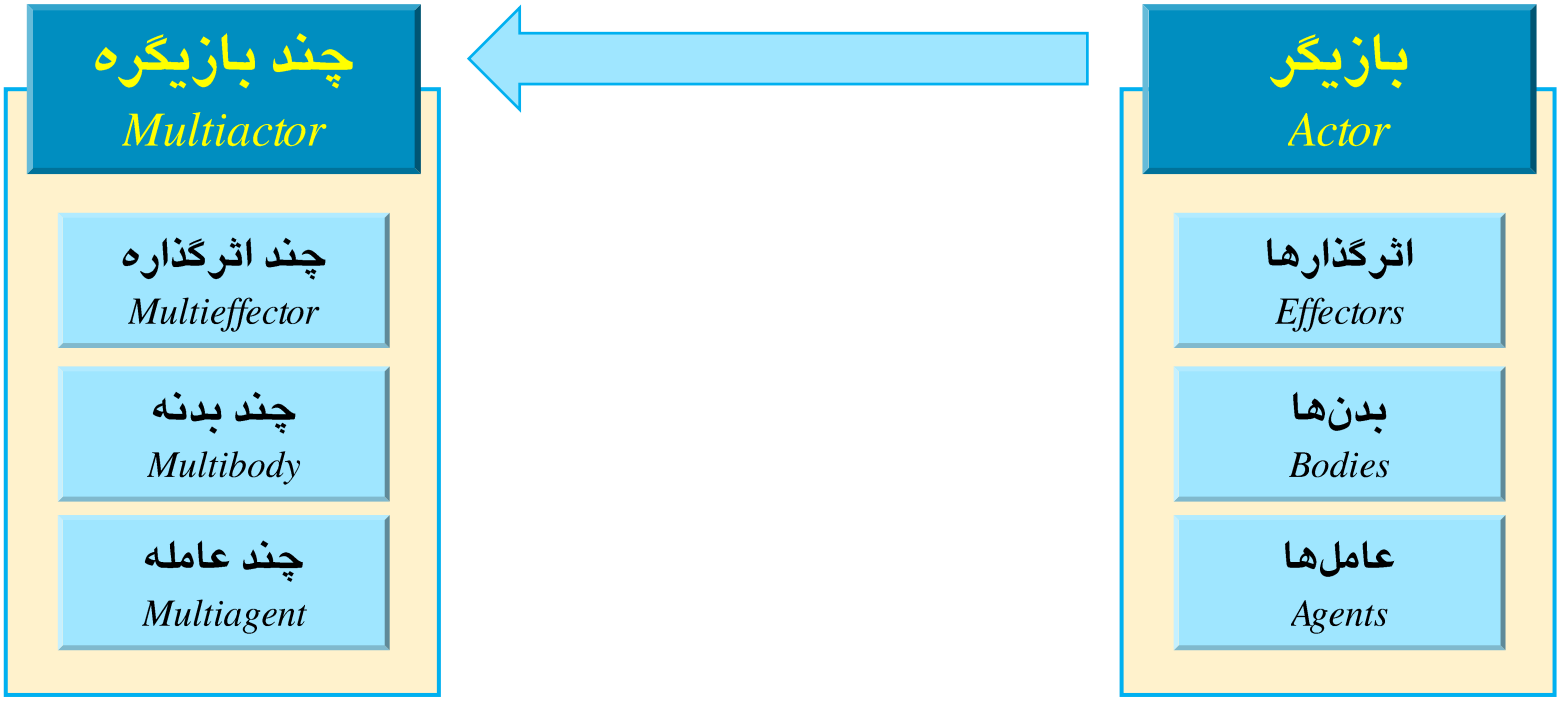
MULTIAGENT PLANNING

طیف گسترده‌ی مسائل بین حالت تک‌عاملی محض و حالت چندعاملی واقعی
(بر حسب درجات مختلف تجزیه‌ی عامل یک‌تکه)

طرح‌ریزی با کنش‌های چندگانه‌ی همزمان

بازیگر

ACTOR



بازیگر: یک اصطلاح کلی برای پوشش دادن مفاهیم اثرگذار، بدن و عامل

طرح‌ریزی با کنش‌های چندگانه‌ی همزمان

PLANNING WITH MULTIPLE SIMULTANEOUS ACTIONS

در وضعیت **چندبازیگره**، چگونه باید
مدل‌های گذار، پلان‌های درست و الگوریتم‌های طرح‌ریزی کارآمد را تعریف کنیم؟

پلان درست *Correct Plan*	پلانی که اگر توسط بازیگرها اجرا شود، به هدف می‌رسد.

البته: در وضعیت واقعاً چندعاملی، عامل‌ها ممکن است موافق اجرای یک پلان خاص نباشند،
اما حداقل می‌دانند که پلان‌ها کار خواهند کرد اگر آنها موافق اجرای آنها باشند.

طرح‌ریزی با کنش‌های چندگانه‌ی همزمان

مدل گذار

PLANNING WITH MULTIPLE SIMULTANEOUS ACTIONS: TRANSITION MODEL

مدل گذار	حالت قطعی	
	تک‌عاملی	$\text{RESULT}(s, a)$
	چند بازیگره	**joint action** $\langle a_1, \ldots, a_n \rangle$

n بازیگر: a_i کنشی است که بازیگر i انجام می‌دهد.

دو مشکل:

- مسئله‌ی توصیف یک مدل گذار برای b^n کنش توأم
- مسئله‌ی طرح‌ریزی توأم با فاکتور انشعاب b^n

همگامی کامل
Perfect Synchronization

فرض همگامی کامل:
- کنش‌ها مقدار زمان مشابهی مصرف می‌کنند.
- کنش‌ها در هر نقطه در پلان توأم، همزمان هستند.

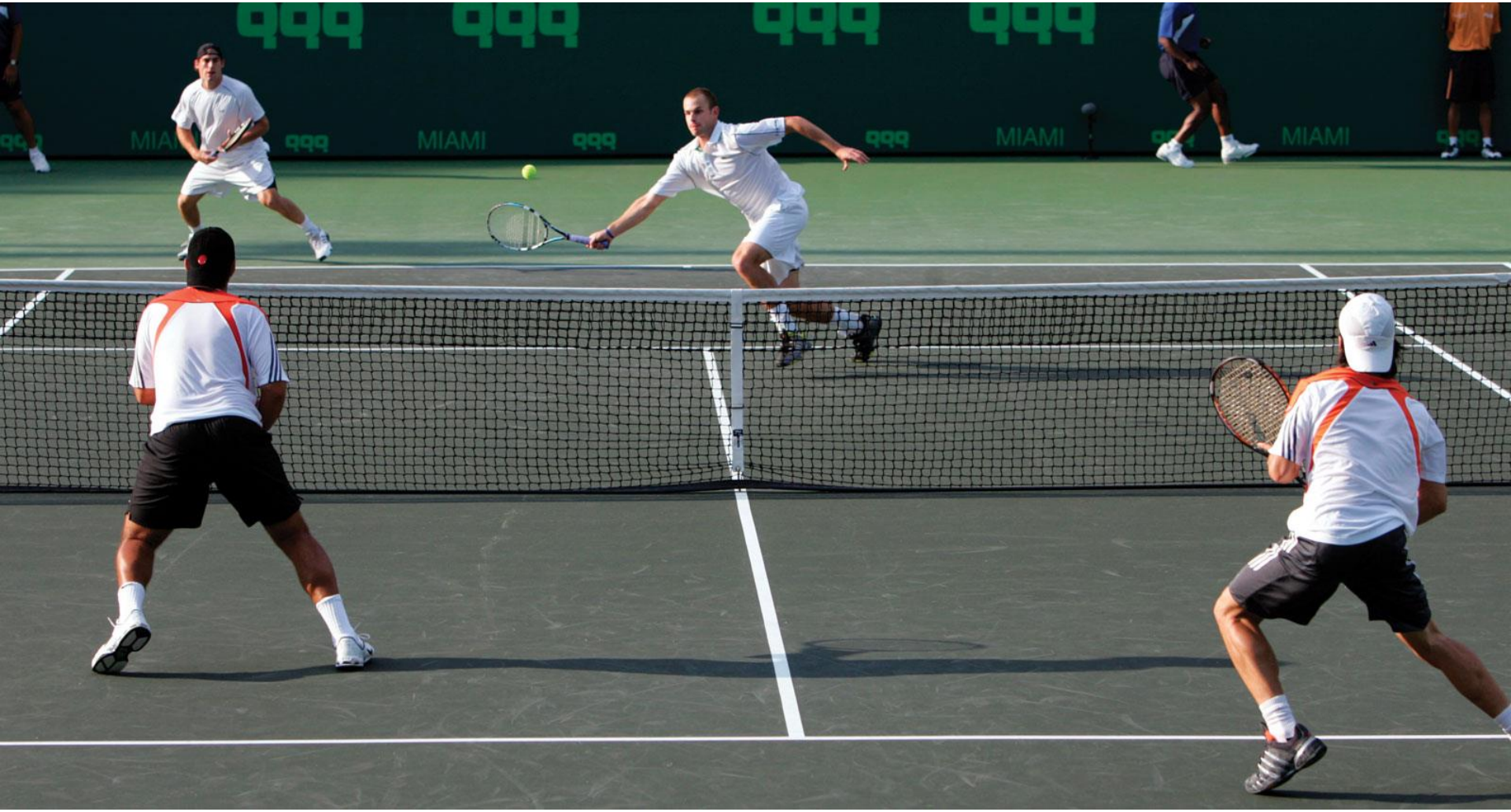
طرح‌ریزی با کنش‌های چندگانه‌ی همزمان

بازیگرهای دارای جفت‌شدگی سست

PLANNING WITH MULTIPLE SIMULTANEOUS ACTIONS: LOOSELY COUPLED ACTORS

روی‌کرد استاندارد برای مسائل دارای جفت‌شدگی سست:
فرض می‌کنیم مسائل کاملاً تفکیک شده‌اند و سپس تعاملات آنها را درست می‌کنیم.

یعنی: شِماهای کنش را به‌گونه‌ای می‌نویسیم که گویی بازیگرها **مستقل** کنش می‌کنند.



Tennis Doubles

طرح‌ریزی با کنش‌های چندگانه‌ی همزمان

مثال: مسئله‌ی تنیس دوبل

$$
\begin{aligned}
&Actors(A, B)\\
&Init(At(A, LeftBaseline) \wedge At(B, RightNet) \wedge\\
&\qquad Approaching(Ball, RightBaseline)) \wedge Partner(A, B) \wedge Partner(B, A)\\
&Goal(Returned(Ball) \wedge (At(a, RightNet) \vee At(a, LeftNet))\\
&Action(Hit(actor, Ball),\\
&\qquad \text{PRECOND:} Approaching(Ball, loc) \wedge At(actor, loc)\\
&\qquad \text{EFFECT:} Returned(Ball))\\
&Action(Go(actor, to),\\
&\qquad \text{PRECOND:} At(actor, loc) \wedge to \neq loc,\\
&\qquad \text{EFFECT:} At(actor, to) \wedge \neg At(actor, loc))
\end{aligned}
$$

مسئله‌ی تنیس دوبل:

دو بازیگر A و B کنار هم بازی می‌کنند و می‌توانند در یکی از چهار محل زیر باشند:
LeftBaseline, RightBaseline, LeftNet, RightNet
توپ می‌تواند برگردد فقط اگر یک بازیکن در مکان درست باشد.
(هر کنش باید شامل یک بازیگر به‌عنوان آرگومان آن باشد. / بازیکن‌های حریف را در نظر نگرفته‌ایم)

طرح‌ریزی با کنش‌های چندگانه‌ی همزمان

بازیگرهای دارای جفت‌شدگی سست: مثال

PLANNING WITH MULTIPLE SIMULTANEOUS ACTIONS: LOOSELY COUPLED ACTORS

فرض می‌کنیم در یک نقطه از بازی، هدف تیم برگشتن توپی است که به سمت آنها پرتاب شده است
و نیز اطمینان از پوشش تور توسط حداقل یکی از بازیگرهاست.

با در نظر گرفتن تعریف شمای کنش،
به‌سادگی دیده می‌شود که **کنش توأم** زیر کار می‌کند:

PLAN 1:

$$A: [Go(A, RightBaseline), Hit(A, Ball)]$$
$$B: [NoOp(B), NoOp(B)] .$$

اما: وقتی پلان این باشد که «هر دو عامل در یک زمان توپ را پرتاب کنند» مشکلاتی بروز می‌کند.

طرح‌ریزی با کنش‌های چندگانه‌ی همزمان

همکاری: اهداف و پلان‌های توأم

COOPERATION: JOINT GOALS AND PLANS

راه‌حل یک **پلان توأم** است که از کنش‌های هر دو عامل تشکیل شده است:

PLAN 1:

$$A: [Go(A, RightBaseline), Hit(A, Ball)]$$
$$B: [NoOp(B), NoOp(B)] .$$

PLAN 2:

$$A: [Go(A, LeftNet), NoOp(A)]$$
$$B: [Go(B, RightBaseline), Hit(B, Ball)] .$$

برای اینکه هر دو عامل به پلان‌های توأم یکسان برسند، **هماهنگی** لازم است.

طرح‌ریزی چندبدنی

MULTI-BODY PLANNING

طرح‌ریزی چندبدنی
Multi-body Planning

مسئله‌ی طرح‌ریزی چندبدنی، در مورد یک عامل متمرکز تنها مطرح می‌شود که می‌تواند کنش موجودیت‌های فیزیکی متعدد را دیکته کند.

پس در حقیقت، این نوع طرح‌ریزی واقعاً چندعاملی نیست،
با این وجود، همگام‌سازی کنش‌ها اهمیت دارد.

برای سادگی فرض می‌کنیم که هر کنش یک گام زمانی مصرف می‌کند
و در هر نقطه از پلان توأم، کنش‌ها به‌صورت همزمان انجام می‌شوند.

مثال:

$$[<Go(A,[Left,Net]),\ Go(B,[Right,Baseline]>;\ <NoOp(A),\ Hit(B,\ Ball)>]$$

می‌توان طرح‌ریزی را با الگوریتم POP که بر روی مجموعه‌ی همه‌ی کنش‌های توأم
ممکن اعمال می‌شود، انجام داد؛ اما اندازه‌ی این مجموعه بسیار بزرگ است ....

طرح‌ریزی چندبدنی

کنش‌های همروند

MULTI-BODY PLANNING

در طرح‌ریزی چندبدنی (با الگوریتم POP) به‌جای مجموعه‌ی همه‌ی کنش‌های توأم:
خطوط همروندی (concurrency) را به توصیف کنش اضافه می‌کنیم

کنش همروند
Concurrent Action

$$Action(Hit(a, Ball),$$
$$\text{CONCURRENT:} b \neq a \Rightarrow \neg Hit(b, Ball)$$
$$\text{PRECOND:} Approaching(Ball, loc) \wedge At(a, loc)$$
$$\text{EFFECT:} Returned(Ball)) .$$

کنش‌های ضروری (required actions) [مثال: حمل یک شیئ توسط دو عامل]

$$Action(Carry(a, cooler, here, there),$$
$$\text{CONCURRENT:} b \neq a \wedge Carry(b, cooler, here, there)$$
$$\text{PRECOND:} At(a, here) \wedge At(cooler, here) \wedge Cooler(cooler)$$
$$\text{EFFECT:} At(a, there) \wedge At(cooler, there) \wedge \neg At(a, here) \wedge \neg At(cooler, here)).$$

طرح‌ریزی کننده مشابه الگوریتم POP عمل می‌کند،
اما تغییرات کوچکی در الگوریتم برای رابطه‌های ترتیب ممکن ایجاد می‌شود.

طرح‌ریزی با عامل‌های چندگانه: همکاری و هماهنگی

مکانیسم‌های هماهنگی

PLANNING WITH MULTIPLE AGENTS: COOPERATION AND COORDINATION

برای اطمینان از توافق بر روی پلان توأم: از **قرارداد** استفاده می‌کنیم.

قرارداد *Convention*	یک قید بر روی انتخاب پلان‌های توأم (بدون این قید که پلان توأم باید کار کند اگر عامل‌ها آن را بپذیرند.)

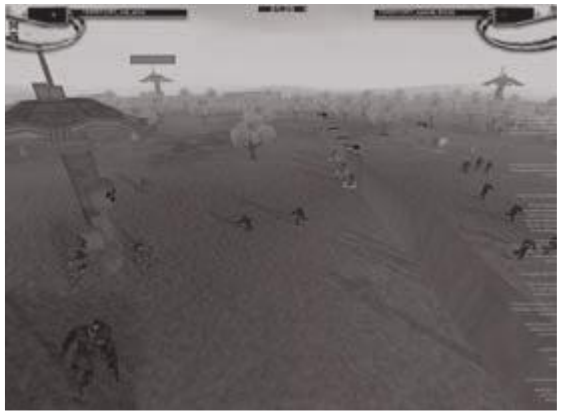
برای مثال: در زمین بازی خود بمانید یا یک بازیکن در تور بماند.

- قراردادی که به‌صورت گسترده پذیرفته شود = **قانون اجتماعی** (social law) مانند زبان
- قرارداد می‌تواند وابسته به دامنه یا مستقل از دامنه باشد.
- قرارداد می‌تواند در طول یک فرآیند تطوری شکل بگیرد (مثل رفتار گله‌ای: flocking behavior).

طرح‌ریزی با عامل‌های چندگانه: همکاری و هماهنگی

مکانیسم‌های هماهنگی: مثال از رفتار گله‌ای (هجوم پرندگان)

FLOCKING EXAMPLE



- Three rules:
  - **Separation:** Steer away from neighbors when you get too close

    جداسازی: از همسایه‌ها دور شوید وقتی خیلی به آنها نزدیک شدید.
  - **Cohesion:** Steer toward the average position of neighbors

    انسجام: به سمت موقعیت میانگین همسایه‌ها بروید.
  - **Alignment:** Steer toward average orientation (heading) of neighbors

    ترازبندی: به سمت جهت‌گیری میانگین همسایه‌ها متمایل شوید.
- Flock exhibits *emergent behavior* of flying as a pseudo-rigid body.

  گله، یک **رفتار بروزکننده** از پرواز را به‌عنوان یک بدن شبه–صلب به نمایش می‌گذارد.

طرح‌ریزی با عامل‌های چندگانه: همکاری و هماهنگی

مکانیسم‌های هماهنگی: ارتباطات

COORDINATION MECHANISM

در صورت عدم حضور قراردادها، باید از **ارتباطات** استفاده شود.

e.g. Mine! Or Yours! in tennis example

زحمت رسیدن به یک پلان توأم موفق می‌توان به عهده‌ی عامل یا طراح عامل گذاشته شود:

عامل *Agent*	**طراح عامل** *Agent Designer*
عامل‌ها تأملی (deliberative) هستند. مدل از سایر عامل‌ها لازم است.	عامل‌ها واکنشی (reactive) هستند. هیچ مدل صریحی از سایر عامل‌ها لازم نیست.

طرح‌ریزی در محیط‌های رقابتی

COMPETITIVE ENVIRONMENTS

عامل‌ها می‌توانند سودمندی‌های متضاد داشته باشند.
(مثل بازی‌های مجموع صفر شبیه شطرنج)

در محیط رقابتی عامل باید:

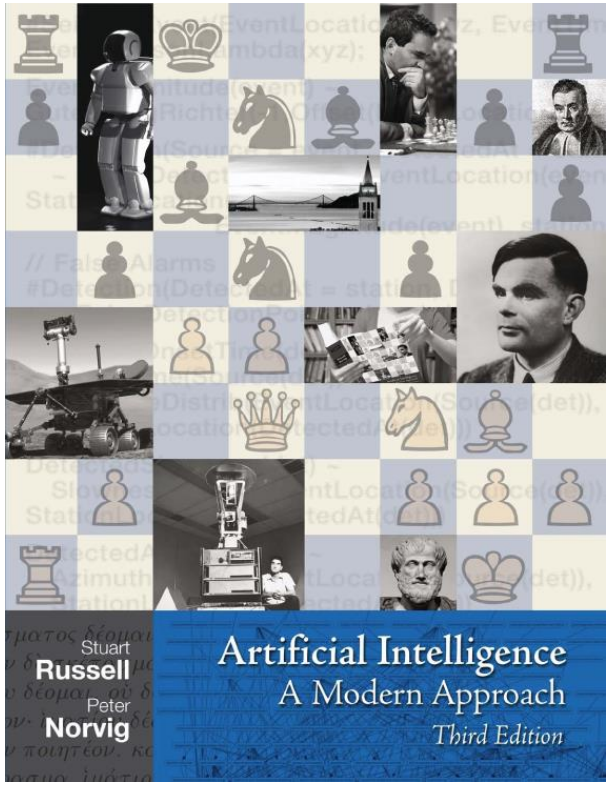
- تشخیص دهد که عامل‌های دیگری وجود دارند.
- پلان‌های برخی از عامل‌های دیگر را محاسبه کند.
- محاسبه کند که سایر عامل‌ها چگونه با پلان شخصی او تعامل می‌کنند.
- در مورد بهترین کنش از نگاه این تعاملات تصمیم بگیرد.

مدل سایر عامل‌ها ضروری است.
با این وجود، تعهدی برای اجرای کنش توأم از سوی عامل‌ها وجود ندارد (محیط رقابتی)

طرح‌ریزی چندعاملی

منابع

منبع اصلی



Stuart Russell and Peter Norvig,
Artificial Intelligence: A Modern Approach,
3rd Edition, Prentice Hall, 2010.

Chapters 11

11 PLANNING AND ACTING IN THE REAL WORLD

In which we see how more expressive representations and more interactive agent architectures lead to planners that are useful in the real world.

The previous chapter introduced the most basic concepts, representations, and algorithms for planning. Planners that are are used in the real world for planning and scheduling the operations of spacecraft, factories, and military campaigns are more complex; they extend both the representation language and the way the planner interacts with the environment. This chapter shows how. Section 11.1 extends the classical language for planning to talk about actions with durations and resource constraints. Section 11.2 describes methods for constructing plans that are organized hierarchically. This allows human experts to communicate to the planner what they know about how to solve the problem. Hierarchy also lends itself to efficient plan construction because the planner can solve a problem at an abstract level before delving into details. Section 11.3 presents agent architectures that can handle uncertain environments and interleave deliberation with execution, and gives some examples of real-world systems. Section 11.4 shows how to plan when the environment contains other agents.

11.1 TIME, SCHEDULES, AND RESOURCES

The classical planning representation talks about *what to do*, and in *what order*, but the representation cannot talk about time: *how long* an action takes and *when* it occurs. For example, the planners of Chapter 10 could produce a schedule for an airline that says which planes are assigned to which flights, but we really need to know departure and arrival times as well. This is the subject matter of **scheduling**. The real world also imposes many **resource constraints**; for example, an airline has a limited number of staff—and staff who are on one flight cannot be on another at the same time. This section covers methods for representing and solving planning problems that include temporal and resource constraints.

The approach we take in this section is "plan first, schedule later": that is, we divide the overall problem into a *planning* phase in which actions are selected, with some ordering constraints, to meet the goals of the problem, and a later *scheduling* phase, in which temporal information is added to the plan to ensure that it meets resource and deadline constraints.

401