



Agents et Systèmes Multi- agents

Adina Magda Florea

Professeur à l'Université "Politehnica" de Bucarest

adina@cs.pub.ro

URL du cours: <http://turing.cs.pub.ro/auf2/>



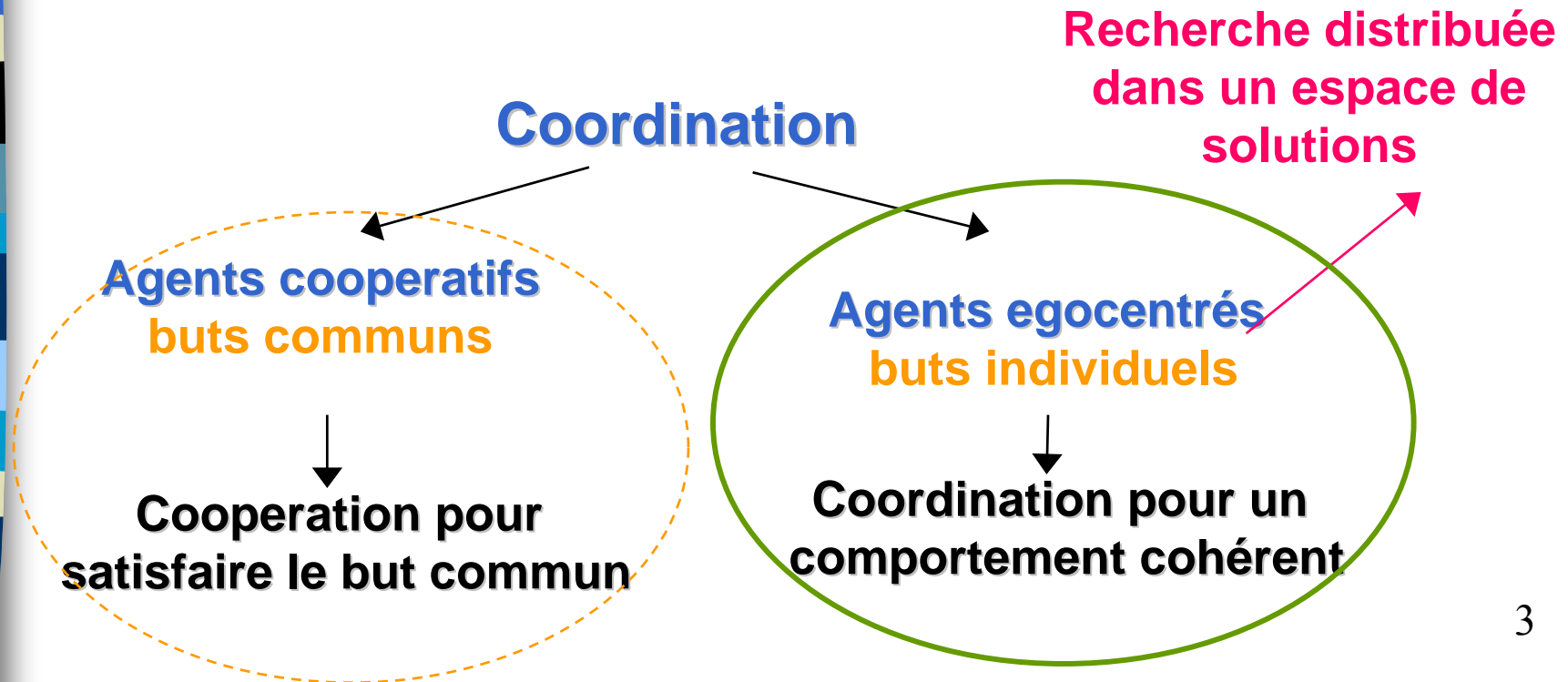
Techniques de négociation

Plan

- 1 Principes de la négociation**
- 2 Négociation basée sur la théorie des jeux**
 - 2.1 Critères d'évaluation**
 - 2.2 Enchères**
 - 2.3 Marchés avec un équilibre général**
 - 2.4 Réseaux contractuel(Contract Nets)**
- 3 Négociation basée sur des heuristiques**
- 4 Négociation basée sur l'argumentation**

1 Principes de la négociation

- **Négociation** = interaction entre agents basée sur la communication avec le but d'arriver à un accord.
- La résolution distribuée des conflits
- Prise de décision
- Offre → acceptée, raffinée, critiquée, ou refusée



■ **La négociation comprend:**

- un langage de communication
- un protocole de négociation
- un processus de décision par lequel un agent décide sa position, les concessions, les critères pour un accord, etc.

⊗ **négociation un-à-un**

⊗ **négociation un-à-plusieurs**

⊗ **négociation plusieurs-à-plusieurs** - eBay <http://www.ebay.com>

- Peut inclure un seul message envoyé par chaque partie ou une conversation avec plusieurs messages envoyés dans les deux directions

■ **Techniques de négociation**

- **Négociation basée sur la théorie des jeux**
- **Négociation basée sur des heuristiques**
- **Négociation basée sur l'argumentation**



2 Négociation basée sur la théorie des jeux

2.1 Critères d'évaluation

- Critères pour évaluer les protocoles de négociation entre plusieurs agents egocentrés
- Des agents sont rationaux
- **Comportement rationnel** = un agent préfère une utilité (un profit) plus grande à une plus petite
- **La maximisation du profit**: profits individuels, profits du group, ou bien-être (“welfare”) social
- **Bien-être social**
 - La somme des utilités (profits) des agents dans une solution.
 - Mesure le bien global des agents
 - Problème: comment comparer les utilités

■ Efficacité Pareto

- Une solution \mathbf{x} , i.e., un **vecteur de profit** $\mathbf{p}(\mathbf{x}_1, \dots, \mathbf{x}_n)$, est **efficace Pareto**, i.e., optimal Pareto, s'il n'y a pas un autre solution \mathbf{x}' tel qu'au moins un agent est mieux dans \mathbf{x}' que dans \mathbf{x} et aucun agent n'est pas pire dans \mathbf{x}' que dans \mathbf{x} .
- Mesure le bien-être global, demande pas la comparaison d'utilités
- Bien-être social \subset efficacité Pareto

■ Rationalité individuelle (RI)

- **RI de la participation d'un agent** = Le profit de l'agent dans la solution négociée n'est pas plus petit que celui obtenu si l'agent n'aurait pas négocié
- **Un mécanisme est RI** si la participation est RI pour tous les agents

■ Stabilité

- un protocole est stable si une fois que les agents ont trouvé une solution, ils ne la changent pas
- **Stratégie dominante** = l'agent obtient le profit maximal avec une stratégie indépendamment des stratégies utilisées par les autres

$r = f(\text{Act}_A, \text{Act}_B)$ le résultat (état) des actions Act_A de l'agent A et Act_B de l'agent B

On dit qu'une stratégie $S_1 = \{r_{11}, r_{12}, \dots, r_{1n}\}$ **domine** une autre stratégie $S_2 = \{r_{21}, r_{22}, \dots, r_{2m}\}$ si n'importe quel résultat $r \in S_1$ est préféré à n'importe quel résultat $r' \in S_2$.

■ Équilibre de Nash

- ❑ Deux stratégies, S_1 de l'agent A et S_2 de l'agent B, sont dans un **équilibre Nash** si:
 - ⑩ dans le cas où l'agent A adopterait S_1 l'agent B ne peut pas faire mieux que d'utiliser S_2 et
 - ⑩ dans le cas où l'agent B adopterait S_2 l'agent A ne peut pas faire mieux que de d'utiliser S_1 .
- ❑ La définition peut être généralisée pour plusieurs agent qui suivent les stratégies S_1, S_2, \dots, S_k . **L'ensemble de stratégies** $\{S_1, S_2, \dots, S_k\}$ suivies par les agents A_1, A_2, \dots, A_k est dans un **équilibre Nash** si, pour chaque agent A_i , la stratégie S_i est la meilleure stratégie à suivre par A_i pourvu que les autres agents suivent les stratégies $\{S_1, S_2, \dots, S_{i-1}, S_{i+1}, \dots, S_k\}$.

Problèmes:

- ❑ pas d'équilibre de Nash
- ❑ plusieurs équilibres de Nash
- ❑ garanti la stabilité seulement au début du jeu

La dilemme du prisonnier

- Bien-être social?
- L'équilibre de Nash ?

Matrice des gains dans le dilemme du prisonnier

		Jouer colonne	
		Coopération	Défection
Jouer ligne	Coopération	3 , 3	0 , 5
	Défection	5 , 0	2 , 2

■ Considérations sur l'efficacité

Pour obtenir une "rationalité parfaite"

- Le nombre d'options à évaluer est trop grand
- Parfois aucun algorithme ne trouve pas la solution optimale

Rationalité limitée

- limite le temps/ressources pour évaluer les options
- réduit l'espace de recherche
- impose des restrictions sur le type d'options

2.2 Enchères

(a) La théorie des enchères = les protocoles et les stratégies des agents dans les enchères

- L'initiateur veut vendre un objet au plus grand prix et les participants veulent l'acheter au plus petit prix possible
- Un protocole centralisé, inclut un initiateur et plusieurs participants
- L'initiateur annonce un objet pour la vente. Dans certains cas, l'objet peut être une combinaison d'autres objets, ou un objet avec plusieurs attributs
- Les participants font des soumissions (offres). Cela peut être fait plusieurs fois, en fonction du type d'enchère
- L'initiateur choisit le gagnant

(b) Les paramètres des enchères

- **Enchères avec valeur privée**: la valeur d'un agent pour un objet dépend seulement de ses préférences privées
- **Enchères avec valeur commune**: la valeur de l'objet dépend complètement de l'évaluation des autres
- **Enchères avec valeur corrélée**: la valeur de l'objet dépend des évaluations internes et externes

(c) Protocoles d'enchère

Enchère anglaise (premier-prix offre-publique) – chaque participant annonce publiquement son offre. Le participant avec la plus grande soumission gagne l'objet au prix de son offre.

Stratégie:

- Dans les enchères à valeurs privées la stratégie dominante est de toujours faire une offre avec un peu plus grande que la plus grande offre actuelle et s'arrêter quand la valeur privée est atteinte.
- Dans les enchères à valeurs corrélées, le participant augmente le prix à une rate constante ou à une rate qu'il considère appropriée.

Enchère premier-prix offre-cachée – chaque participant soumet une offre sans savoir les offres des autres. Celui qui fait la plus grande soumission gagne l'objet et paye le montant de son offre.

Stratégie:

- Pas de stratégie dominante
- Offrir moins que sa vraie évaluation, mais cela dépend des autres soumissions qui sont pas connues.

Enchère hollandaise (descendante) – l’initiateur diminue tout le temps le prix jusqu’à ce qu’un des participants achète l’objet au prix actuel.

Stratégie:

- Équivalente du point de vue stratégique avec l’enchère premier-prix offre-cachée.
- Efficiente en temps réel.

Enchère Vickery (deuxième-prix offre-cachée) – chaque participant soumet une offre sans savoir les offres des autres. Celui avec la plus grande offre gagne, mais au prix de la deuxième plus grande offre

Stratégie:

- La stratégie dominante du participant est d’offrir sa vraie évaluation

Enchères tous-payent – chaque participant doit payer le montant de son offre (ou un autre montant) à l’initiateur

(d) Problèmes avec les protocoles d'enchère

- Des blocages peuvent apparaître
- Initiateur menteur
 - Problème dans l'enchère Vickery
 - Problème dans l'enchère anglaise – l'initiateur utilise des faux participants dans l'enchère pour augmenter l'évaluation de l'objet par les autres participants
 - L'initiateur offre le deuxième plus grand prix pour obtenir son prix réservé – il est possible qu'il arrive à garder l'objet
 - Les enchères avec des valeurs communes peuvent être soumises à la **malédiction du gagnant** ("winner's curse
 - Des enchères inter-liées – le participant peut mentir sur la valeur d'un objet pour obtenir une combinaison d'objets à leur prix d'évaluation

2.4 Marchés avec un équilibre général

- La théorie de l'équilibre général = une théorie micro-économique
- Un nombre de n objets - $g = 1, n$
- prix $p=[p_1, \dots, p_n]$, où $p_g \in \mathbb{R}$ est le prix de l'objet g
- 2 types d'agents: consommateurs et producteurs

Consommateurs:

- une fonction d'utilité $u_i(x_i)$ qui code les préférences sur des différents paquets de consommation $x_i=[x_{i1}, \dots, x_{in}]$, où $x_{ig} \in \mathbb{R}^+$ et l'allocation d'un objet g par le consommateur i .
- une attribution initiale $e_i=[e_{i1}, \dots, e_{in}]$, où e_{ig} est l'attribution de l'objet g

Producteurs:

- vecteur de production $y_j=[y_{j1}, \dots, y_{jn}]$ où y_{jg} est le montant des objets g produits par le producteur j .
- ensemble de possibilités de production Y_j – l'ensemble des vecteurs de production possibles

- Le **profit** du producteur j est $p \cdot y_j$, où $y_j \in Y_j$.
- Les profits des producteurs sont divisés parmi les consommateurs selon des proportions pré-déterminées qui peuvent ne pas être égales.
- Soit θ_{ij} la fraction du producteur j que le consommateur i possède.
- Les profits des producteurs sont divisés parmi les consommateurs selon ces fractions
- Les prix peuvent changer et les agents peuvent changer leurs plans de production et consommation mais
 - les vraies productions et consommations apparaissent quand **le marché a atteint un équilibre général**

(p^*, x^*, y^*) est un **équilibre Walrasian** si:

- les marchés sont propres

$$\sum_i x_i^* = \sum_i e_i + \sum_j y_j^*$$

- chaque consommateur i maximise ses préférences avec les prix donnés

$$x_i^* = \arg \max_{x_i \in R_n^+, p^* \cdot x_i \leq p^* \cdot e_i + \sum_j \theta_{ij} p^* \cdot y_j} u_i(x_i)$$

- chaque producteur j maximise ses profits avec les prix donnés

$$y_j^* = \arg \max_{y_j \in Y_j} p^* \cdot y_j$$

L'algorithme de tâtonnement des prix distribués

Algorithme pour l'adaptation du prix:

$p_g = 1$ pour tout $g \in [1..n]$

Donner à λ_g une valeur positive pour tout $g \in [1..n-1]$

repeat

 envoie \mathbf{p} aux consommateurs et producteurs

 reçoit un plan de production y_j de chaque producteur j

 envoie les plans \mathbf{y}_j aux consommateurs

 reçoit un plan de consommation \mathbf{x}_i de chaque consommateur i

for $g=1$ to $n-1$ **do**

$$p_g = p_g + \lambda_g (\sum_i (x_{ig} - e_{ig}) - \sum_j y_{jg})$$

until $|\sum_i (x_{ig} - e_{ig}) - \sum_j y_{jg}| < \varepsilon$ pour tout $g \in [1..n-1]$

Informers les consommateurs et producteurs qu'un équilibre a été atteint

L'algorithme de tâtonnement des prix distribués

L'algorithme pour le consommateur i :

repeat

reçoit \mathbf{p} de l'adapteur

reçoit un plan de production \mathbf{y}_j de chaque j de l'adapteur

annonce à l'adapteur un plan de consommation $\mathbf{x}_i \in \mathbb{R}_+^n$
qui maximise $u_i(x_i)$ avec les contraintes du budget

$$\mathbf{p} \cdot \mathbf{x}_i \leq \mathbf{p} \cdot \mathbf{e}_i + \sum_j \theta_{ij} \mathbf{p} \cdot \mathbf{y}_j$$

until il a été informé qu'un équilibre à été atteint
echager et consumer

Algorithme pour le producteur j :

repeat

reçoit \mathbf{p} de l'adapteur

annonce à l'adapteur un plan de production $\mathbf{y}_j \in Y_j$ qui

maximise $\mathbf{p} \cdot \mathbf{y}_j$

until il a été informé qu'un équilibre à été atteint
echager et produire

2.5 Réseaux contractuel (contract nets)

Les marchés d'équilibre général utilisent

- des prix globaux
- un médiateur centralisé

Désavantages:

- pas tous les prix sont globaux
- surcharge du médiateur
- médiateur – point faible
- les agents n'ont pas un contrôle direct sur les agent à qui ils envoient des informations

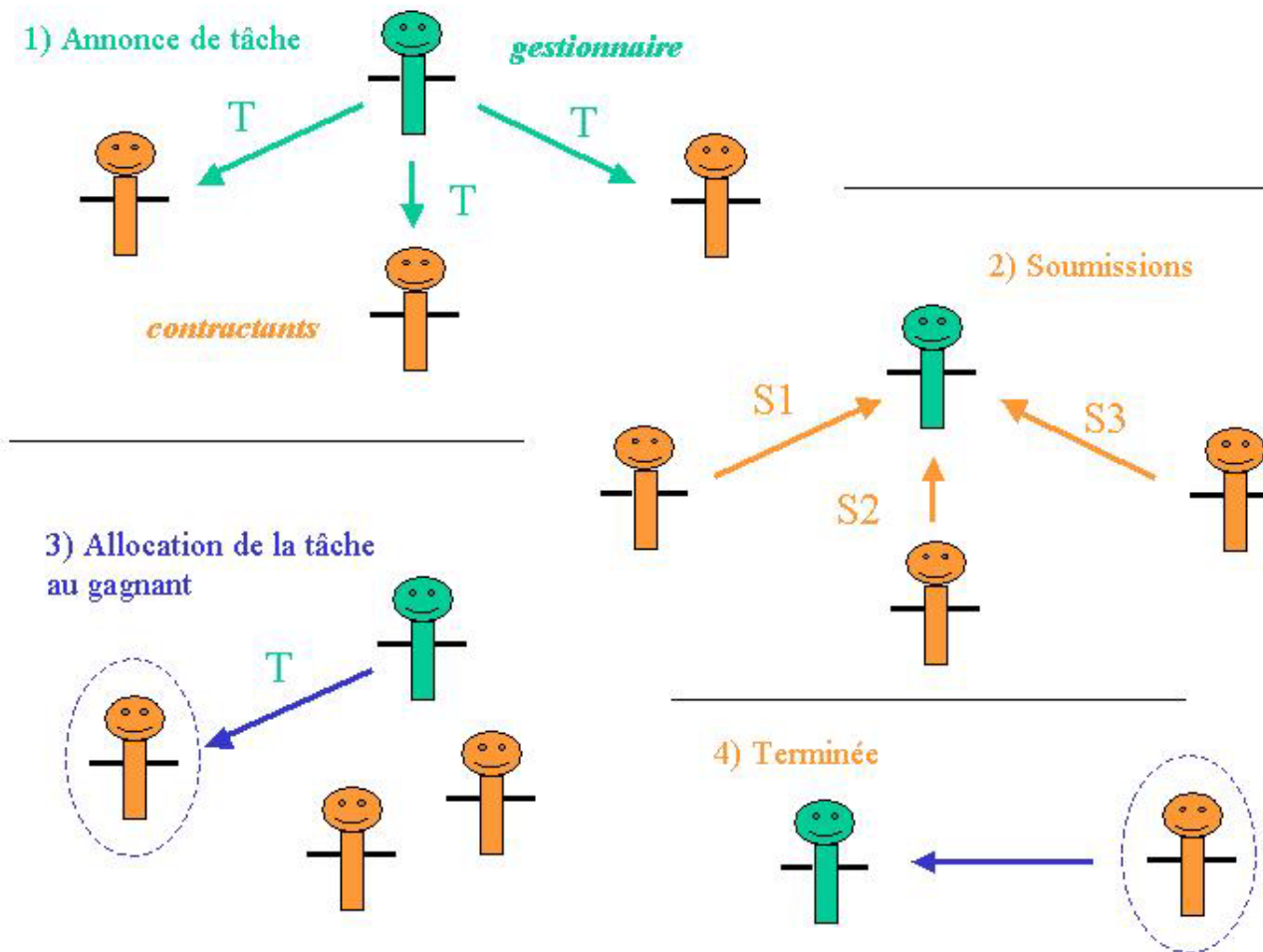
Besoin d'une solution plus distribuée

Allocation des tâches par négociation - Contract Net

- Un type de pont entre la négociation de la théorie des jeux et celle basée sur des heuristiques
- Modèle formel pour faire des offres et des décisions

(a) Allocation des tâches par réseau contractuel

- Dans le protocole réseau contractuel, les agents peuvent prendre **deux rôles**: **gestionnaire** et **contractant**.



(b) Allocation des tâches par redistribution

- Un **domaine orienté tâche** est un triple
- $\langle T, Ag, c \rangle$ où
 - T est un ensemble de tâches;
 - $Ag = \{1, \dots, n\}$ est un ensemble d'agents qui participent à la négociation;
 - $c: P(T) \rightarrow R^+$ est une fonction coût qui définit les coûts nécessaires pour exécuter chaque sous-ensemble de tâches.
- La fonction coût doit satisfaire deux contraintes: elle doit être monotone et le coût de ne pas exécuter une tâche doit être zéro, notamment $c(\Phi) = 0$. La première contrainte dit que, si on ajoute une tâche à un ensemble des tâches, le coût de l'ensemble résultant doit être toujours supérieur au coût de l'ensemble initial.
- Une **rencontre** dans un domaine orienté tâche est l'attribution d'une collection d'ensembles de tâches $R = \{E_1, \dots, E_n\}$, $E_i \subseteq T$, $i \in Ag$, aux agents Ag .

Dans une telle rencontre on se pose le problème si chaque agent ne peut pas être mieux (avoir des coûts moins élevés) par un reallocation des tâches à exécuter. Par exemple, si on a trois agents a_1, a_2 et a_3 ($Ag = \{a_1, a_2, a_3\}$) et les tâches $T = \{t_1, t_2, t_3, t_4, t_5\}$ on peut définir la rencontre

$$R = \{E_1, E_2, E_3\} \text{ avec } E_1 = \{t_1, t_3\}, E_2 = \{t_2\}, E_3 = \{t_4, t_5\}$$

Le **coût d'une affaire** pour l'agent a_1 est défini comme $c(D_1)$ alors que le coût pour l'agent a_2 est $c(D_2)$. **L'utilité d'une affaire** représente combien l'agent doit gagner de l'affaire et peut être définie comme

$$\text{utilité}_i(\alpha) = c_i(E1) - c_i(D_i), \text{ pour } i = 1, 2 \text{ correspondant aux agents } a_1 \text{ et } a_2$$

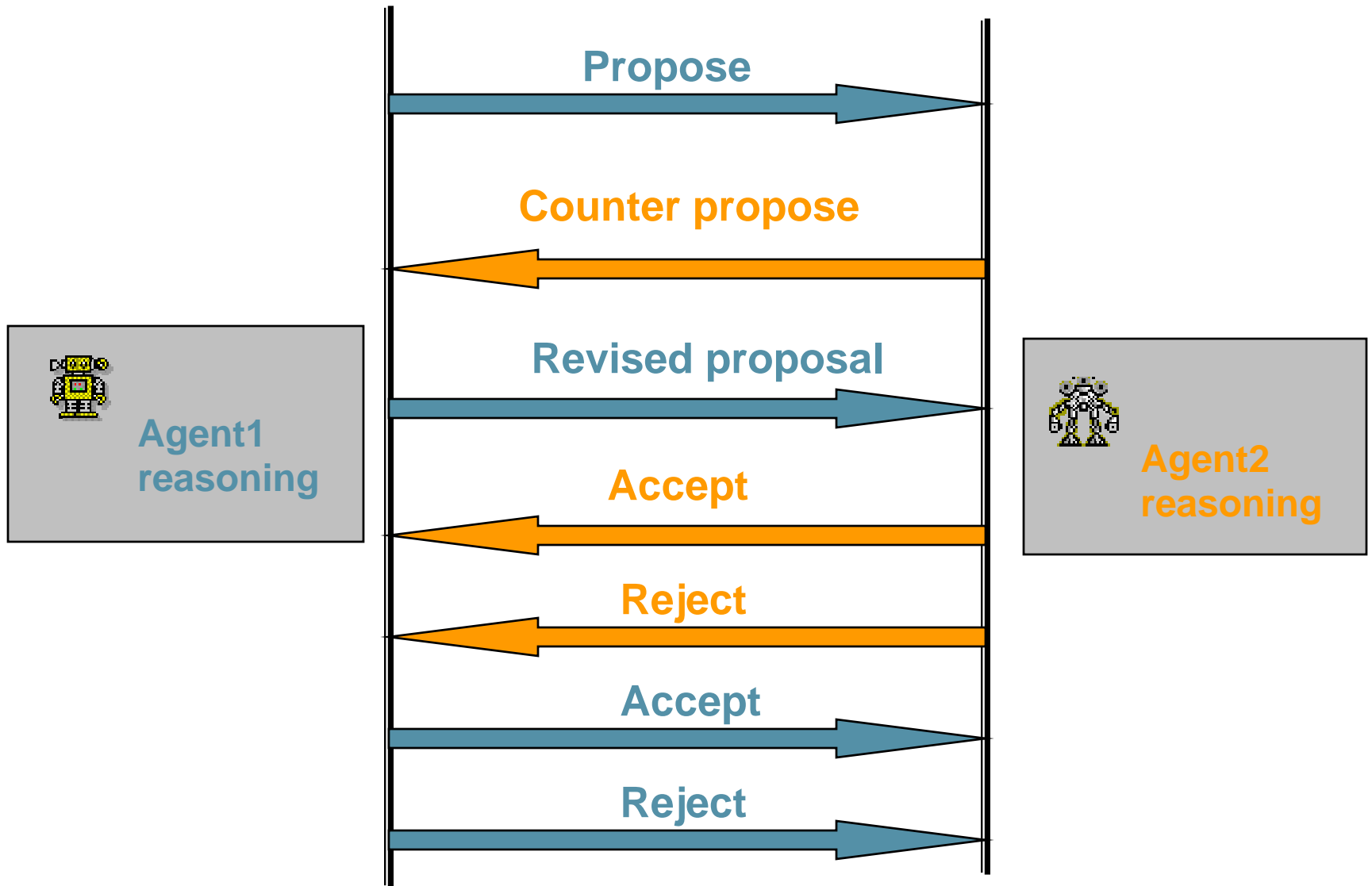
L'utilité d'une affaire représente combien l'agent doit gagner de l'affaire et peut être définie comme

- **utilité_i(α) = $c_i(E1) - c_i(D_i)$** , pour $i = 1, 2$ correspondant aux agents a_1 et a_2
- Une affaire α_1 est dite de **dominer** une autre affaire α_2 si et seulement si:
 - l'affaire α_1 est au moins aussi convenable que α_2 pour chaque agent
- ∀ $i \in \{1,2\}$ utilité_i(α_1) ≥ utilité_i(α_2)
- l'affaire α_1 est meilleure que α_2 pour au moins un agent
- ∃ $i \in \{1,2\}$ utilité_i(α_1) > utilité_i(α_2)



3 Négociation avec des heuristiques

- Produit une solution bonne mais pas optimale
- Négociation basée sur des heuristiques:
 - approximations des techniques de la théorie des jeux
 - modèles de négociation informels
- Sans médiateur central
- Les actes de paroles sont privés entre les agents négociants
- Le protocole ne donne pas un cours optimal d'action
- Problème principal: la prise des décisions heuristique pendant la négociation



- Un **objet de négociation (NO)** est l'intervalle des issues sur lesquelles un accord doit être trouvé
- L'objet de la négociation peut être une action que l'agent négociateur *A* demande à un autre agent *B* d'effectuer, un service que l'agent *A* demande à *B*, ou l'offre d'un service que *A* veut effectuer pour *B* si *B* accepte les conditions de *A*.

NO03: NO

- **Name:** Paint_House
- **Cost:** Value:100, Type: integer, Modif=Yes;
- **Deadline:** Value: May_12, Type: date, Modif=No;
- **Quality:** Value: high, Type: one of (low, average, high), Modif=Yes

- **(Request NO)** – demande d'un objet de négociation
- **(Accept name(NO))** – accepte la demande de NO
- **(Reject name(NO))** – refuse la demande de NO
- **(ModReq name(NO) value(NO,X,V1))** – modifie la demande en modifiant la valeur de l'attribut X du NO à une autre valeur V1



4 Négociation par argumentation

- **Arguments** utilisés pour convaincre l'autre d'accepter la proposition faite
- Différents types d'arguments
- Chaque type d'argument définit des pré-conditions pour son utilisation. Si elles sont remplies, alors l'agent peut utiliser l'argument.
- L'agent a besoin d'une stratégie pour décider quel argument utiliser
- On utilise un modèle BDI

- **Appels à une promesse passée** – le négociateur *A* rappelle *B* d'une promesse passée concernant le NO, i.e., l'agent *B* a promis dans une négociation passée à l'agent *A* d'offrir ou effectuer un NO.
 - **Pré-conditions:** *A* doit vérifier si une promesse d'un NO a été reçue dans passé dans une négociation conclue avec succès.
- **Promesse d'une récompense future** – le négociateur *A* promet de faire NO pour un autre agent *B* à un moment dans le futur.
 - **Pré-conditions:** *A* doit trouver un désir de l'agent *B* pour un moment dans le futur, si possible un désir qui peut être satisfait par une action (service) que *A* peut effectuer mais *B* non.
- **Appels au propre intérêt** – l'agent *A* croit que arrivant à un accord sur NO est dans l'intérêt de *B* et essaye de convaincre *B* de ça.
 - **Pré-conditions:** *A* doit trouver (ou inférer) un des désirs de *B* qui sera satisfaite si *B* a NO ou *A* doit trouver un autre objet de négociation NO' qui a été offert auparavant dans le marché et il croit que NO est mieux que NO'.
 - **Menace** – le négociateur menace de refuser faire/offrir quelque chose à *B* ou il menace qu'il fera quelque chose qui contredit les désirs de *B*.
 - **Pré-conditions:** *A* doit trouver un des désirs de *B* directement satisfaite par un NO que *A* peut offrir ou *A* doit trouver une action qui est contradictoire avec ce qu'il croit être un des désir de *B*.

Références

- T.W. Sandholm. Distributed rational decision making. In Multiagent Systems - A Modern Approach to Distributed Artificial Intelligence, G. Weiss (Ed.), The MIT Press, 2001, p.201-258.
- J.S. Rosenschein, G. Zlotkin. Designing conventions for automated négociation. In Readings in Agents, M. Huhns & M. Singh (Eds.), Morgan Kaufmann, 1998, p.253-370.
- M.P. Wellman. A market-oriented programming environment and its applications to distributed multicommodity flow problems. Journal of Artificial Intelligence Research, 1, 1993, p.1-23.
- N.R. Jennings, e.a., Automated négociation: prospects, methods, and challenges, Journal of Group Decision and négociation, 2000.
- S. Kraus, K. Sycara, A. Evenchik, Reaching agreements through arumentation: a logical model and implementation, Artificial Intelligence, Elsevier Science, 104, 1998, p. 1-69.
- A. Florea, B. Panghe. Achieving Cooperation of Self-interested Agents Based on Cost”, In Proceedings of the 15th European Meeting on Cybernetics and System Research, Session: From Agent Theories to Agent Implementation, Vienna, 2000, p.591-596.