

Étude de dynamiques d'argumentation

Martin Alexis^{1 2}, encadré par Nicolas Maudet² et Élise Bonzon³

¹ MPRI, Université Paris 7 Diderot

² LIP6 – équipe SMA, Université Paris 6 (UPMC)

³ LIPADE – équipe IAD, Université Paris 5 Descartes

21 août 2016

Le contexte général

Depuis la croissance du Web 2.0 plusieurs sites de communication et d'interaction sociale ont vu le jour. On peut ainsi citer les sites Twitter et Facebook. D'autres plateformes telles que *debategraph.org* ou *debate.org* sont également apparues sur la toile. Ces plateformes proposent aux utilisateurs de participer à des débats. La recherche s'est donc intéressée à des méthodes d'analyse afin de quantifier l'issue d'un débat, connaître les arguments les plus importants ou encore détecter les trolls. C'est le cas de Leite et Martins [1, 2] qui ont trouvé un algorithme permettant de calculer un score sur chacun des arguments en fonction de la topologie du graphe (on construit une relation d'attaque entre arguments), des votes qui sont donnés à chacun des arguments ou encore des votes sur les attaques (non étudié ici).

Le problème étudié

Actuellement, lorsqu'un débat en ligne est clos, il est souvent laissé sans conclusion. On peut évidemment se fonder notre propre avis à l'aide de notre propre connaissance, en lisant les arguments qui ont été avancé au cours du débat. En revanche, il est difficile, voire impossible de déterminer l'avis général qui en ressort. Il peut être donc intéressant de donner une conclusion à ce débat. Une approche dynamique mettant en jeu plusieurs personnes est a priori une bonne solution pour tenter de répondre à cette question. Tout ces facteurs font de cette question un sujet intéressant et

important à étudier. L'objectif de ce stage est donc de construire un système dynamique et d'analyser les propriétés que ce système possède.

La contribution proposée

Étant donné un débat terminé, aucun argument supplémentaire ne sera ajouté et le graphe d'argumentation est connu, le modèle propose à un certain nombre d'agents (réels ou bots) de voter sur chacun des arguments. Les votes ainsi obtenu et le graphe lui même nous permettra d'établir un avis général sur l'issue du débat. Cet avis est donné sous la forme d'un score attribué à la question initialement posée. Ce score est calculé grâce à l'algorithme de Leite et Martins [1]. Ce jeu se déroule en deux étapes :

Étape 1 chaque agent se fait son propre avis à partir du graphe d'argumentation.

Étape 2 L'ensemble des agents se regroupent et vote, au tour par tour, sur chacun des arguments. Le score de la question est mise à jour à chaque tour.

Les arguments en faveur de sa validité

Le fait de faire un jeu post débat permet de faire intervenir des personnes qui n'auraient potentiellement pas participé au débat. Cela permet à ces joueurs d'avoir un certain recul. De plus ce jeu permet de faire intervenir un nombre important de joueurs ayant chacun un avis différent. Ces deux points permettent d'augmenter la probabilité de s'approcher de l'opinion des personnes participants au débat. Enfin la facilité d'implémentation de ce jeu sur une plateforme et la possibilité d'utiliser exclusivement des bots pour analyser le débat sont d'autres atouts à cette solution. Toutefois, afin que ce jeu soit réellement implémentable il faut s'assurer de sa terminaison.

Le bilan et les perspectives

Une des difficultés dans l'analyse des débats est la subjectivité inhérente à ces débats. Il est donc difficile d'évaluer notre méthode par rapport à des débats réels. L'étude de ce système permet d'avoir une approche nouvelle dans l'évaluation de graphes d'argumentation. C'est une direction qui n'a pas encore été traitée et qui peut se révéler utile et cohérente dans l'analyse de ces débats.

On pourrait étendre ce système en rajoutant la possibilité de voter sur les attaques de la même façon que Leite et Martins le font dans [2].

1 Introduction et présentation

On va commencer par définir formellement comment on représente un débat et un algorithme de Leite et Martins permettant de mettre un poids sur chaque argument.

1.1 Le contexte

Dung [3] propose d'interpréter un débat comme un graphe, les noeuds sont les arguments annoncés par les différentes parties et les arêtes correspondent à une relation d'attaque entre les arguments. On ajoute à chaque argument deux entiers qui correspondent aux votes positifs et aux votes négatifs. De plus on intègre la *question* dans le graphe sous la forme d'un noeud qui n'a pas de vote.

Définition 1.1.

Formellement un graphe d'argumentation est un triplet :

$F = \langle \mathcal{A}, \mathcal{R}, V \rangle$ avec :

- $\mathcal{A} : L$ l'ensemble des arguments et la question q .
- $\mathcal{R} \subseteq \mathcal{A} \times \mathcal{A} : Une relation binaire entre arguments tels que : $(a, b) \in \mathcal{R}$ si a attaque b .$
- $V : \mathcal{A} \setminus \{q\} \rightarrow \mathbb{N} \times \mathbb{N} : La fonction qui associe, à chaque argument, le nombre de "like" et de "dislike", $V(a) = (v^+, v^-)$ signifie que a a v^+ "like" et v^- "dislike".$

Dans la suite on notera $v^+(a)$ (resp. $v^-(a)$) pour désigner le nombre de "like" (resp. "dislike") de l'argument a . Pour plus de lisibilité, si il n'y a pas d'ambigüité sur l'argument on notera simplement v^+ et v^- .

Exemple 1.1.

Un retour en arrière nous porte à l'époque où les romains était maître de toute la Gaule, excepté dans un petit village de notre Bretagne actuelle, en Armorique [4]. Dans ce village, un débat d'une importance capitale éclate, "doit-on garder Assurancetourix¹ dans le village? ".

L'ensemble des villageois participent au débat et on recueille l'ensemble des arguments dans l'ordre d'apparition :

- a_1 . Il est inutile, ça fait une bouche de plus à nourrir.
- a_2 . Ce n'est pas vrai, il nous a aidé dans certain cas critiques, lors de la construction, par les romains, du domaine des dieux par exemple.

1. Assurancetourix est le barde du village

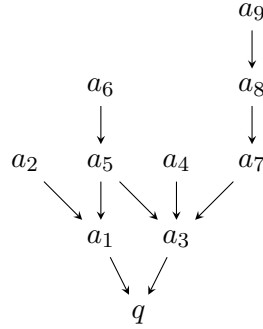


FIGURE 1 – Graphe représentant le débat sur l’expulsion d’Assurancetourix.

- a_3 . Il chante faux et nous casse les oreilles en permanence.
- a_4 . Rien nous empêche de continuer à l’en empêcher, en plus ça nous défoule.
- a_5 . Vous n’allez tout de même pas créer un apatride !
- a_6 . On est pas obligé de créer un apatride, je propose de l’envoyer à Lutèce, il sera bien là-bas.
- a_7 . Il est au contraire essentiel, il apporte un minimum de culture dans notre village qui en est dépourvu.
- a_8 . On a pas besoin de culture, on a besoin de nourriture et que le ciel ne nous tombe pas sur la tête.
- a_9 . En revanche il dispense des cours à nos jeunes du village.

Après ces quelques arguments énoncés, les gaulois se sont énervés et une bagarre a commencé. La figure 1 montre le graphe construit à l’aide de ces arguments.

Sur ce graphe, la question est le noeud q et les arguments $\{a_1, \dots, a_9\}$ sont les arguments cités ci-dessus. De plus on considère qu’il n’y a eu aucun vote pour le moment, qui ne sont donc pas représentés ici.

On remarque dans un premier temps que la question n’attaque aucun argument. Ce sera vrai pour tous les graphes d’argumentation traités dans ce travail. Dans un second temps on peut remarquer, par exemple, que l’argument a_4 attaque l’argument a_3 qui lui même attaque la question. On peut donc considérer a_4 comme un argument *défenseur* de la question.

Notation 1.1.

On note $Att(a)$ l’ensemble des arguments directs qui attaquent a : $Att(a) = \{b \mid (b, a) \in \mathcal{R}\}$

Sur l'exemple précédent $Att(a_3) = \{a_4, a_5, a_7\}$.

1.2 L'algorithme de Leite et Martins

Afin de juger de l'importance de chacun des arguments dans le débat, Leite et Martins [1] ont proposé un algorithme qui calcule une valeur comprise entre $[0, 1]$ pour chacun des arguments. Plus la valeur d'un argument sera proche de 1, plus son poids dans le débat sera important.

Cet algorithme procède en 2 étapes :

Prise en compte des votes Dans un premier temps on définit une fonction τ qui, pour chaque argument, calcule une valeur dans l'intervalle $[0, 1]$.

Définition 1.2 (Simple Vote Aggregation).

Etant donné un graphe d'argumentation $F = \langle \mathcal{A}, \mathcal{R}, V \rangle$, on définit la fonction $\tau_\varepsilon : \mathcal{A} \rightarrow [0, 1]$ par :

$$\tau_\varepsilon(a) = \begin{cases} 0 & \text{si } v^+ = v^- = 0 \\ \frac{v^+}{v^+ + v^- + \varepsilon} & \text{sinon} \end{cases}$$

Avec $\varepsilon \geq 0$.

Utilisation de la topologie du graphe On peut ensuite définir une sémantique d'argumentation qui permettra de calculer l'importance d'un argument dans le graphe.

Définition 1.3 (Simple Product Semantics).

On définit une sémantique de produit simple par un tuple $\mathcal{S}_\varepsilon = \langle [0, 1], \tau_\varepsilon, \wedge, \vee, \neg \rangle$ tel que :

1. $x_1 \wedge x_2 = x_1 \cdot x_2$
2. $x_1 \vee x_2 = x_1 + x_2 - x_1 \cdot x_2$
3. $\neg x_1 = 1 - x_1$
4. $\varepsilon \geq 0$

Enfin, il reste à utiliser cette sémantique afin de déterminer le poids de chaque argument.

Définition 1.4 (Social Model).

Soit $F = \langle \mathcal{A}, \mathcal{R}, V \rangle$ un graphe d'argumentation et $\mathcal{S} = \langle [0, 1], \tau, \wedge, \vee, \neg \rangle$. La fonction $LM : \mathcal{A} \rightarrow [0, 1]$ est un \mathcal{S} -Model pour F si :

$$LM(a) = \tau(a) \wedge \neg \vee \{LM(a_i) \mid a_i \in Att(a)\}$$

On est maintenant capable, en fonction des votes et de la topologie du graphe de qualifier l'importance des arguments dans le graphe.

La méthode de Leite et Martins à plusieurs avantages par rapport à d'autres algorithmes tel que l'algorithme décrit par Philippe Besnard et Anthony Hunter [5] ou celui de Leila Amgoud [6].

En effet l'algorithme de Leite et Martins possède trois propriétés importantes. Soit F un graphe d'argumentation et \mathcal{S} une simple product sémantique.

1. Si $\varepsilon > 0$ alors il existe un unique \mathcal{S} -Model. Cela a pour conséquence de n'avoir qu'une seule valeur possible pour $LM(a)$ pour tout a .
2. Il existe une suite qui converge vers ce modèle. Ainsi quelle que soit la topologie du graphe on pourra calculer la valeur de chaque argument.
3. La généralité de leur modèle permet de modifier, entre autre, la fonction τ_ε et de garder vrai les deux premières propriétés.

En revanche il est difficile de savoir si cette méthode permet réellement d'extraire les arguments les plus importants. En effet, à aucun moment on utilise le contenu des arguments et la préférence d'un argument par rapport à un autre est une question très subjective.

2 Le jeu

On va définir un jeu multi-joueurs dont l'objectif est d'avoir un aperçu de la conclusion que l'on pourrait donner au débat.

Dans notre exemple 1 elle permettrait de dire si le village est plutôt pour garder Assurancetourix ou non.

L'intérêt d'avoir plusieurs agents dans le jeu nous permet de créer une conclusion utilisant plusieurs avis différents et donc, en théorie, d'être plus objectif.

Afin de rendre plus clair le fonctionnement du jeu on le présentera de façon informelle sur l'exemple précédent avant de le présenter formellement.

Définition 2.1 (Jeu d'argumentation).

Soit $F = \langle \mathcal{A}, \mathcal{R}, V \rangle$, un jeu d'argumentation sur le graphe F est un tuple $G = \langle \mathcal{J}, \mathcal{F}, \mathcal{F}_{\mathcal{J}}, \mathcal{S}, \mathcal{U} \rangle$ où :

- $\mathcal{J} = \{1, \dots, n\}$ est un ensemble agents (= joueurs)
- \mathcal{F} est un ensemble d'états tel que $F_i \in \mathcal{F}$ si $F_i = \langle \mathcal{A}, \mathcal{R}, V_i \rangle$.
- $\mathcal{F}_{\mathcal{J}} = (F_1, \dots, F_n)$ un ensemble de graphe d'argumentation ou F_k est un graphe associé au joueur k .
- \mathcal{S} est la sémantique utilisée.
- $\mathcal{U} = (u_1, \dots, u_n)$ tel que u_k est la fonction d'utilité du joueur k .

2.1 Initialisation

Astérix, Panoramix et Abraracourcix ont été désignés par leur sagesse ou leur rang pour délibérer de l'issue du débat qui encours dans le village.

Chacun d'eux commence par analyser le débat et, de manière personnelle, vote sur chacun des arguments en fonction de leur préférences. Ainsi Abraracourcix, vote positivement sur tous les arguments en faveur de l'expulsion et contre tous les autres. Panoramix vote exactement l'inverse d'Abraracourcix. Quant à Astérix, il préfère simplement voter pour les arguments avec lesquels il est d'accord et contre ceux qu'il rejète sans tenir compte du résultat.

Définition 2.2.

$F_k = \langle \mathcal{A}, \mathcal{R}, V_k \rangle$ est le graphe d'argumentation associé au joueur k . La fonction V_k définit donc la préférence du joueur k sur chacun des arguments de \mathcal{A} .

En particulier, F_k est équivalent à au moins un état de \mathcal{F} .

Définition 2.3.

Soit LM_k un \mathcal{S} -Model pour le graphe F_k .

On appelle la valeur du jeu pour le joueur k (ou valeur du joueur k) la valeur de $LM_k(q)$. dans la suite, on notera cette valeur simplement LM_k .

A la suite de cette première étape, ils calculent chacun leur valeur personnelle grâce à une formule établie par Archéoptérix².

La figure 2 référence les votes et la valeur calculée pour les trois participants.

Astérix et Panoramix sont donc plutôt pour garder Assurancetourix (valeurs proches de 1) alors qu'Abraracourcix est plutôt contre (valeur proche de 0).

2.2 Déroulement du jeu

Les trois membres du village se regroupe ensuite pour conclure l'issue du débat. Au début, le débat est dépourvu de vote et la valeur du débat est calculée également grâce à la formule d'Archéoptérix. Cette formule étant connue par les 3 participants ils peuvent connaître l'impact d'un vote sur le débat. Chacun leur tour, ils vont voter sur un argument. Le tour d'un joueur ce déroule de cette façon :

2. Archéoptérix est le maître druide de Panoramix

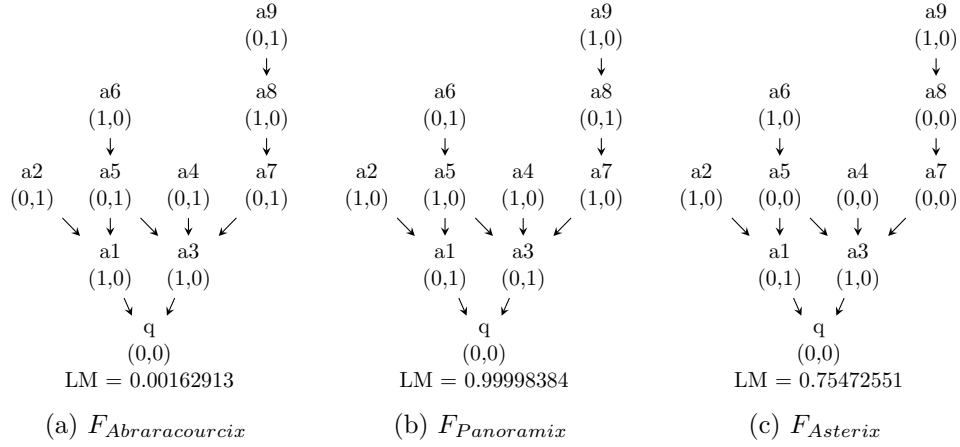


FIGURE 2 – Votes et valeur de chaque joueur

1. Le joueur choisi un argument.
2. Il vote pour ou contre cet argument.
3. La valeur du débat est recalculée.

Sachant qu'un joueur ne peut avoir, au maximum, qu'un seul vote par argument. En revanche, un joueur a le droit de changer d'avis. Il peut donc commencer par voter pour l'argument a_1 , puis plus tard changer son vote pour voter contre ou retirer son vote positif, etc.

Définition 2.4.

Soit F_i l'état courant de G et LM_{F_i} un \mathcal{S} -Model pour le graphe F_i . on appelle valeur du jeu la valeur de $LM_{F_i}(q)$. On la notera LM ou LM_{F_i} dans la suite.

On appelle $F_0 \in \mathcal{F}$ l'état initial. F_0 est le graphe d'argumentation $F_0 = \langle \mathcal{A}, \mathcal{R}, V_0 \rangle$ tel que $V_0(a) = (0,0)$ pour tout $a \in \mathcal{A}$.

Le coup d'un joueur est un tuple (F_i, k, a, v, F_j) tel que :

- F_i est l'état courant
- k est le joueur.
- $a \in \mathcal{A} \setminus \{q\}$ est l'argument sur lequel le joueur k a voté.
- $v \in \{-, 0, +\}$ indique le vote du joueur. $-$ signifie que le joueur a voté contre, 0 qu'il n'a pas voté ou retiré son vote et $+$ qu'il a voté pour l'argument a .

— F_j est l'état obtenu tel que $V_j(a') = V_i(a')$ pour tout $a' \in \mathcal{A} \setminus \{a\}$ et

$$v_j(a) = \begin{cases} (v_i^+(a) + 1, v_i^-(a)) & \text{si } v = + \text{ et } k \text{ n'a pas de vote sur } a \\ (v_i^+(a) + 1, v_i^-(a) - 1) & \text{si } v = + \text{ et } k \text{ a un vote négatif sur } a \\ (v_i^+(a) - 1, v_i^-(a)) & \text{si } v = 0 \text{ et } k \text{ a un vote positif sur } a \\ (v_i^+(a), v_i^-(a) - 1) & \text{si } v = 0 \text{ et } k \text{ a un vote négatif sur } a \\ (v_i^+(a), v_i^-(a) + 1) & \text{si } v = - \text{ et } k \text{ n'a pas de vote sur } a \\ (v_i^+(a) - 1, v_i^-(a) + 1) & \text{si } v = - \text{ et } k \text{ a un vote positif sur } a \end{cases}$$

Dans toutes les autres situations le coup est impossible et n'existe pas.

Une partie du jeu est un ensemble de coups ordonnés.

2.3 L'objectif

L'objectif pour chaque joueur est que la valeur du jeu soit la plus proche possible de sa valeur.

Définition 2.5 (Stratégie).

Une stratégie pour le joueur k est une fonction $\sigma_k : \mathcal{F} \longrightarrow \mathcal{F} \cup \emptyset$ tel que :

- Si $\sigma_k(F) = \emptyset$ alors le joueur k ne joue pas.
- Si $\sigma_k(F) = F'$ alors il existe un coup (F, k, a, v, F') valide.

On note Σ_k l'ensemble des stratégies du joueur k .

Définition 2.6.

u_k est la fonction d'utilité du joueur k , elle est défini par :

$$\begin{aligned} u_k : \mathcal{F} &\longrightarrow \mathbb{R} \\ F &\longmapsto 1 - |LM_F - LM_k| \end{aligned}$$

Définition 2.7.

Un coup (F, k, a, v, F') est dit améliorant pour le joueur k si $u_k(F') > u_k(F)$.

Autrement dit, l'objectif pour le joueur k est de maximiser u_k .

2.4 La dynamique du jeu

La dynamique d'un jeu comporte deux paramètres importants :

L'enchaînement entre les joueurs Dans ce travail on s'intéressera à deux enchaînements différents. Le plus connu, notamment utilisé dans la plupart des jeux de plateaux, est *Round Robin*. Les agents joueront toujours selon un ordre précis. Le deuxième qu'on utilisera est appelé *Random*. Il consiste à choisir le prochain joueur selon une loi de probabilité uniforme.

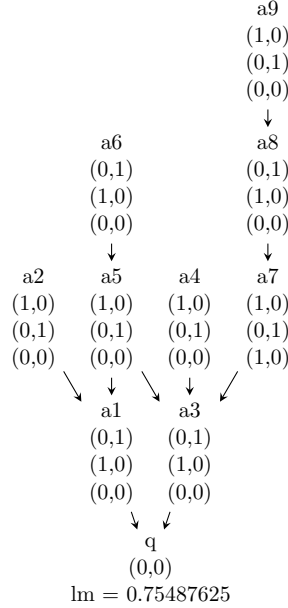


FIGURE 3 – Votes des trois gaulois

Le comportement de chacun des joueurs Le deuxième point de la dynamique permet de définir comment vont se comporter les joueurs. Cela va avoir un impact direct sur le nombre de stratégies possibles. Là encore on va considérer deux comportements différents.

best response Les joueurs vont toujours choisir le meilleur coup possible. Donc ils n'ont qu'une seule stratégie qui est

$$\sigma_k(F) = \operatorname{argmax}\{u_k(F') \mid (F, k, a, v, F') \text{ est un coup valide}\}.$$

better response Les joueurs vont choisir un coup améliorant. Les stratégies possibles sont donc de la forme

$$\sigma_k(F) \in \{F' \mid (F, k, a, v, F') \text{ est un coup valide et } u_k(F') > u_k(F)\}.$$

Durant le jeu entre les trois gaulois, ils ont décidé de choisir une dynamique *Round Robin* et *best response*. La valeur du débat à la fin du jeu est $LM = 0.86950214$. Le détail des votes finaux est donné sur la figure 3. La liste des coups joués ainsi qu'une courbe montrant l'évolution de la valeur du graphe sont données dans l'annexe B.

Sur ce graphe, les votes des trois joueurs sont écrits en dessous de chaque argument. Le premier couple est le vote de Panoramix, le second celui d'Abraracourcix et enfin le dernier est celui d'Astérix.

On remarque que Panoramix a voté de manière à faire augmenter la valeur au maximum et Abraracourcix de façon à la faire descendre au maximum. Seul Astérix a des votes mitigés, ce qui est cohérent avec leurs choix personnels. On remarque également que la valeur est plutôt en faveur du maintien d'Assurancetourix dans le village. Cela est dû à la présence de deux personnes plutôt pour le maintien ainsi qu'à la topologie du graphe.

Ce deuxième point est important, en effet, considérons le graphe n'ayant que des arguments attaquant la question et aucun argument défenseur. Un jeu sur ce graphe avec 2 agents qui sont plutôt pour et un agent contre sera remporté par l'agent qui est contre. On trouvera un exemple de jeu ayant cette caractéristique dans l'annexe C.

3 Les problèmes liés au modèle

Il y a plusieurs problèmes liés à la définition de l'agrégation des votes τ_ε .

- Dans notre cas, on considère que la question ne reçoit aucun vote. Mais si on prend cette définition pour la sémantique de Leite et Martins [1] alors $LM(q) = 0$ quelques soit les attaquants car $\tau_\varepsilon(q) = 0$.
- D'après la définition de Leite et Martins, un argument qui n'a pas de vote a un poids à 0 (car $\tau_\varepsilon(a) = 0$) et un vote qui n'a que des votes négatif a également un poids à 0.

Afin de résoudre ces problèmes on a proposé une variante de la fonction d'agrégation des votes.

Définition 3.1 (Laplace Vote Aggregation).

Soit $\tau : \mathcal{A} \rightarrow [0, 1]$ la fonction défini par :

$$\begin{cases} \tau(q) &= 1 \\ \tau(a) &= \frac{1}{2} \cdot \left(1 + \text{sgn}(v^+ - v^-) \cdot \left(1 - e^{\frac{-|v^+ - v^-|}{\varepsilon}} \right) \right) \end{cases} \quad (\forall a \in \mathcal{A} \setminus \{q\})$$

$$\text{où } \text{sgn}(x) = \begin{cases} -1 & \text{si } x < 0 \\ 1 & \text{sinon} \end{cases} \quad \text{et } \varepsilon = -\frac{\#\text{joueurs}}{\ln(0.04)}$$

En utilisant la fonction de répartition de la loi de Laplace, un vote qui a un nombre équivalent de votes positifs et de votes négatifs est égale à 0.5. Un nombre de votes positifs important fait croître la valeur de l'argument. À l'inverse, un nombre de votes négatifs important fait décroître la valeur de l'argument. on répond donc bien deux problèmes précédents.

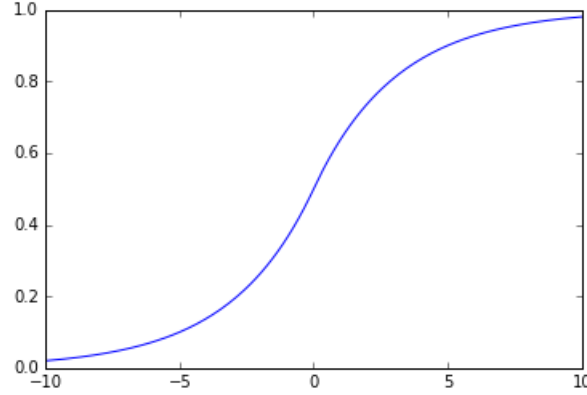


FIGURE 4 – Poids d'un argument en fonction du nombre de votes, avec 10 joueurs

La figure 4 représente l'évolution du poids d'un argument en fonction de la différence entre le nombre de votes positifs et le nombre de votes négatifs. On remarque bien qu'une différence égale à 0 implique une valeur $\tau(a) = 0.5$, et que plus le nombre de votes positifs augmente, plus la valeur du jeu augmente et inversement. On remarque également que la valeur max est égale à 0.98 et que la valeur min est égale à 0.02.

Afin de garder les propriétés de convergence de l'algorithme de Leite et Martins, il faut que la fonction τ soit croissante selon $v^+(a)$ et décroissante selon $v^-(a)$.

Proposition 3.1. *La fonction τ est croissante selon $v^+(a)$ et décroissante selon $v^-(a)$.*

Démonstration. La preuve est donnée dans l'annexe D

□

3.1 Difficultés combinatoires

Un jeu avec k joueurs sur un graphe à n arguments (sans compter la question) possède 3^{n*k} états. Ainsi un jeu à 2 joueurs sur un graphe possédant 3 arguments (sans compter la question) possédera 729 états différents.

Rechercher l'état qui satisfasse au mieux l'ensemble des joueurs devient donc très vite impossible. Un tel état pourrait être, par exemple, celui qui minimise la somme des valeurs absolues des distances entre la valeur des joueurs et la valeur du jeu.

L'intérêt de ce jeu serait donc de trouver "rapidement" un état qui est satisfaisant, mais pas forcément le meilleur.

4 Les résultats

Dans cette section on va analyser notre système pour déterminer ses propriétés. Une des propriété recherchée est la convergence.

Définition 4.1.

Un état d'équilibre dans le jeu G est un état dans lequel aucun joueur n'a de coup améliorant.

Définition 4.2.

Un jeu est dit convergent si, depuis tout état atteignable depuis la situation initiale, il est possible d'atteindre un équilibre.

On pourra s'intéresser également à la valeur finale obtenue, et la comparer :

- À la moyenne des valeurs des joueurs.
- Aux valeurs du joueur min et du joueur max. Le joueur min (resp. max) est celui qui à la plus petite (resp. grande) valeur parmi tous les joueurs.
- À la valeur obtenue par aggrégation sincère des votes.

4.1 Le cas des arbres

Afin de voir si un cycle pouvait arriver sur les arbres j'ai réalisé un grand nombre de tests. Le résultat de ces tests montre, a priori, que l'on converge toujours et ce quelle que soit la dynamique.

On va essayer de caractériser ces jeux et voir quelles sont leurs propriétés.

Soit F un arbre d'argumentation et G un jeu d'argumentation avec n joueurs sur F . On rappelle que $\Sigma = \Sigma_1 \times \Sigma_2 \times \cdots \times \Sigma_n$ est l'ensemble des stratégies de tous les joueurs. Pour le joueur i , on note Σ_{-i} l'ensemble des stratégies des autres joueurs. $\Sigma_{-i} = \Sigma_1 \times \cdots \times \Sigma_{i-1} \times \Sigma_{i+1} \times \cdots \times \Sigma_n$.

4.1.1 Quelques propriétés

Définition 4.3.

Un joueur max-extrême (resp. min-extrême) est un joueur dont la valeur est maximale (resp. minimale) dans le jeu.

Un joueur max-extrême (resp. min-extrême) a décidé de voter pour (resp. contre) tous les arguments défendant la question et contre (resp. pour) tous les autres.

On note j_{min} (resp. j_{max}) le joueur ayant la plus petite (resp. grande) valeur entre tous les joueurs.

Propriétés 4.1.

1. *A un joueur, la valeur finale du jeu peut être différente de la valeur du joueur.*
2. *Un joueur extrême ne peut jamais changer un de ses votes.*
3. *Il existe des états d'équilibres qui ne sont pas entre j_{min} et j_{max} .*
4. *Il n'existe pas forcément d'équilibre entre j_{min} et j_{max} .*
5. *A deux joueurs, si la dynamique est Round_Robin avec la stratégie best response et si la valeur initiale du jeu est entre j_{min} et j_{max} alors on converge vers cette valeur.*

Démonstration.

1. un exemple est donnée dans l'annexe E
2. Soit j un joueur max-extrême. Son objectif va être de monter la valeur au maximum, il va donc toujours vouloir voter contre les argument attaquant la question et pour les arguments défendant. Il ne changera donc jamais d'avis.
3. un exemple est donnée dans l'annexe F
4. un exemple est donnée dans l'annexe G

□

Preuve du point 5. Soit G un jeu à deux joueurs sur un arbre d'argumentation F .

Notons que la situation où le joueur j_{min} à voter de manière à faire descendre la valeur du jeu au maximum et j_{max} à jouer pour faire monter la valeur au maximum à une valeur égale à la situation initiale. Cela est dû au fait que les joueurs auront opposé tous leurs votes et donc on aura autant de votes positifs que de votes négatifs sur chaque argument. Notons de plus que cette situation est une situation d'équilibre. En effet, aucun joueur n'a intérêt à changer d'avis. Montrons que si la dynamique est *Round_Robin* et *best response*, alors on atteint ce point d'équilibre.

Supposons, quitte à réordonner les joueurs, que le joueur j_{min} commence à jouer. Notons LM_0 la valeur initiale du jeu et LM_s la valeur du jeu dans l'état s .

Remarquons en premier lieu que la dynamique *best response* nous assure qu'à chaque coup du joueur j , $|LM_s - LM_j| \geq |LM_0 - LM_j|$.

Soit s un état dans la partie et soit $\alpha_s = \{\alpha_{min}^1, \dots, \alpha_{min}^k, \alpha_{max}^1, \dots, \alpha_{max}^{k'}\}$ la liste des arguments n'ayant qu'un seul vote avec :

- $\{\alpha_{min}^1, \dots, \alpha_{min}^k\}$ les arguments où seul j_{min} a voté.
- $\{\alpha_{max}^1, \dots, \alpha_{max}^k\}$ les arguments où seul j_{max} a voté.

Si $\alpha_s = \emptyset$ alors $LM_s = LM_0$.

Comme la dynamique est *Round Robin* on a $k = k'$ ou $k = k' + 1$.

Lorsque le joueur j joue dans l'état s il a deux possibilités,

- Voter sur un argument de α_s . En votant contre le dernier vote de son adversaire il s'assure $|LM_s - LM_j| \geq |LM_0 - LM_j|$. (La valeur obtenue sera une valeur déjà obtenu précédemment).
- Il existe un argument $a \notin \alpha_s$ tel que voter sur cet argument est meilleur que tous les contres des arguments de α_s .

Dans le premier cas, $|\alpha_{s+1}| < |\alpha_s|$. Dans le second cas on diminue le nombre d'arguments qui n'ont pas encore de vote.

Tant que α_s ne sera pas vide on aura pas une situation d'équilibre. \square

Définition 4.4 (Chemin d'amélioration).

Un chemin dans Σ est une suite $s = (s^0, s^1, \dots)$ d'éléments de Σ tel que $\forall k > 0, \exists i \in \mathcal{J}$ et $s^k = (s_{-i}^{k-1}, x)$ avec $x \in \Sigma_i$ et $x \neq s_i^{k-1}$. s^0 est appelé le point initial.

s est un chemin d'amélioration si pour tout $k > 0$ on a $u_i(s^k) > u_i(s^{k-1})$

Exemple 4.1.

Soit G le jeu sur l'arbre à trois noeuds et 2 joueurs présenté si dessous :

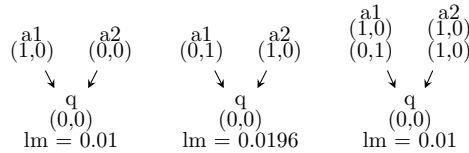


FIGURE 5 – de gauche à droite, graphe du joueur 1, graphe du joueur 2, graphe de l'état final.

Sur ce jeu, la partie :

$[0.25](1, a_2, +)[0.05] \rightarrow (2, a_2, +)[0.01] \rightarrow (2, a_1, -)[0.018] \rightarrow (1, a_1, +)[0.01]$ est un chemin améliorant. On a indiqué la valeur du jeu au départ et après chaque coup. Ce qui est important de noter dans ce chemin c'est que le

troisième coup, coup du joueur 2, fait éloigner le joueur 1 de son objectif. En effet u_1 passe de 1 à 0.992. Mais étant donné que c'est le joueur 2 qui joue, seule l'évolution de u_2 nous intéresse.

Soit le graphe orienté G_Σ construit à l'aide de tous les chemins d'amélioration. Notons que les sommets puits du graphe G_Σ correspondent exactement aux états d'équilibres dans notre jeu.

Définition 4.5 (Weak Finite Improvement Property).

On dit que le jeu possède la Weak FIP si, dans le graphe G_Σ , depuis tout sommet s il existe un chemin qui mène à un sommet puits.

Définition 4.6 (Finite Improvement Property).

On dit que le jeu possède la FIP si le graphe G_Σ est acyclique.

Il est clair qu'un jeu qui possède la FIP ou la Weak FIP converge.

Remarque 4.1.

Un jeu qui ne possède que la Weak FIP peut continuer indéfiniment dans un cycle, en revanche, sous des conditions raisonnables [7, 8, 9], on peut dire que cela est improbable.

Dans notre cas, G est un jeu fini, quelque soit la dynamique. En effet, le nombre d'état possible est fini, donc le nombre de stratégie l'est aussi. En revanche, G ne possède pas forcément la FIP. Si on considère la dynamique *Random en better response*, le jeu présenté dans l'annexe H en est un contre exemple.

Conjecture 4.1. *Soit F un arbre et G un jeu d'argumentation sur F . Si on munit G de l'une des trois dynamiques :*

1. Round.Robin et best response.
2. Round.Robin et better response.
3. Random et best response.

Alors G possède la FIP.

Conjecture 4.2. *Soit F un arbre et G un jeu d'argumentation sur F . Si on munit G de la dynamique Random et better response. Alors G possède la Weak FIP.*

Définition 4.7 (fonction de potentiel).

Soit F un graphe d'argumentation et $\mathcal{J} = \{1, \dots, n\}$ n joueurs. Soit Σ_i l'ensemble des stratégies du joueur i et $\Sigma = \Sigma_1 \times \Sigma_2 \times \dots \times \Sigma_n$. De plus, on note $u_i : \Sigma \rightarrow \mathcal{R}$ la fonction de paiement du joueur i .

$P : \Sigma \longrightarrow \mathcal{R}$ est un potentiel pour le jeu si pour tout $i \in \mathcal{I}$ et pour tout $s_{-i} \in \Sigma_{-i}$ on a :

$$u_i(s_{-i}, x) - u_i(s_{-i}, z) = P(s_{-i}, x) - P(s_{-i}, z)$$

Pour tout $x, z \in \Sigma_i$

les jeux possédant la FIP exhibent des fonctions potentielles, notre conjecture est donc que de telles fonctions existent pour notre jeu. Toutefois, nous n'avons pas réussi à trouver de telles fonctions.

4.2 Les graphes

Jusqu'à présent on a regardé uniquement des arbres. Étant donné que la plus part des débats sont des graphes, il est intéressant de se demander comment notre modèle se comporte sur des graphes.

4.2.1 Les graphes cycliques

Est-ce que la convergence est assuré sur les graphes ? C'est la première question que je me suis posé. Afin d'y répondre j'ai réalisé des tests et malheureusement, ce qui semble vrai sur les arbres ne l'est plus sur les graphes. En effet, on trouve un cycle dans la dynamique *Round_Robin* et *best response*. Le jeu de la figure 6 en est un contre exemple.

On observe bien sur ce jeu que très rapidement (au bout du neuvième tour) les joueurs rentrent dans un cycle sur l'argument a_4 . Ce jeu ne possède donc pas la FIP et ce, quelque soit la dynamique, car dans tous les cas ce cycle existera. Il ne possède pas la Weak FIP non plus car la stratégie *best response* nous empêche de sortir du cycle.

En revanche il est possible qu'un jeu avec une autre dynamique possède la Weak FIP.

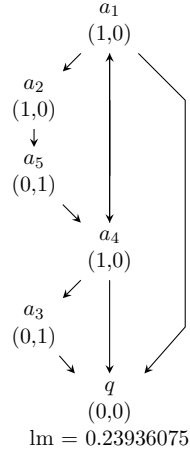
Définition 4.8 (Argument ambigu).

Un argument ambigu est un argument qui est attaquant et défenseur de la question.

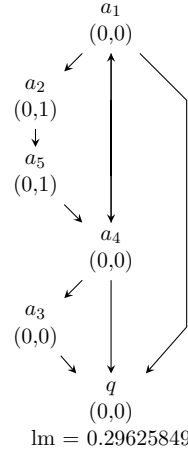
Remarque : Dans tous les jeux tester jusqu'à présent et dans celui-ci également les cycles font toujours intervenir au moins un argument ambigu.

On peut donc conclure que les arguments ambigus augmentent fortement les cas de cycle.

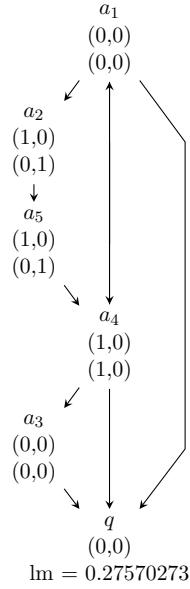
On peut se demander donc quel est la plus petite classe de graphes sur laquelle on semble converger.



(a) Graphe du joueur 1



(b) Graphe du joueur 2



(c) Situation finale

	joueur 1	joueur 2	général
tour 1	-	-	0.28723462
tour 2	a4 dislike	-	0.25156434
tour 3	-	a4 like	0.28723462
tour 4	a5 like	-	0.2675169
tour 5	-	a5 dislike	0.28723462
tour 6	a4 like	-	0.27570273
tour 7	-	a2 dislike	0.2962805
tour 8	a2 like	-	0.27570273
tour 9	-	a4 dislike	0.28723462
tour 10	a4 dislike	-	0.2500625
tour 11	-	a4 like	0.28723462
tour 12	a4 like	-	0.27570273

FIGURE 6 – Exemple de jeu qui contient un cycle

4.2.2 Les graphes acycliques

De la même manière que les graphes cycliques, il existe des graphes acycliques sur lesquels notre modèle ne converge pas. Vous trouverez un exemple de graphe dans l'annexe I. Ce jeu utilise également la dynamique *Round_Robin* avec la stratégie *best response*.

On remarque également qu'une fois encore le cycle met en cause un argument ambigu. La même question que sur les graphes cycliques peut alors se poser, à savoir, est-ce que les jeux avec une autre dynamique peuvent posséder la Weak FIP ?

Étant donné que tous les cycles rencontrés jusqu'à présent mettent en cause un argument ambigu on va donc vérifier si notre modèle converge sur les graphes sans argument ambigu.

4.2.3 Les graphes non ambigu

Pour plus de lisibilité, on appelle graphe non ambigu un graphe qui ne contient pas d'argument ambigu.

Proposition 4.1. *Un graphe F est non ambigu si et seulement si F est un graphe biparti*

Démonstration.

\Rightarrow Soit F un graphe non ambigu. F ne possède pas de cycle impair, car un cycle impair rendrait ambigu tous les arguments qui le compose. Donc F est biparti.

\Leftarrow Soit F un graphe biparti, alors il existe $X, Y \subset \mathcal{A}$ tel que X et Y forment une partition de \mathcal{A} et $X \cap Y = \emptyset$. De plus il n'existe pas d'arêtes entre les sommets de X (resp. Y). Donc tous les arguments de X (resp. Y) se défendent entre eux et attaquent ceux de Y (resp. X). Donc il n'existe pas d'argument ambigu. \square

Tous les tests réalisés sur des graphes non ambigus convergent. De plus le point 5 de la propriété 4.1 est également vrai pour les graphes non ambigu. En effet, dans la démonstration du point 5, on utilise seulement implicitement que les arbres ne possèdent pas d'argument ambigu, mais pas la structure particulière des arbres.

Il est également possible que, si la convergence est assurée sur les arbres, alors la même démonstration soit utilisable sur les graphes non ambigu.

5 Implémentation

Le code et les tests ont été réalisés en Lua 5.3. Tout le code est disponible sur GitHub à l'adresse <https://github.com/Alexis-Martin/DebateGames>.

6 Conclusion

Le modèle présenté ici semble bien se comporter sur l'ensemble des graphes non ambigus. En revanche, dès qu'il y a un argument ambigu, il y a de forte chance que notre système ne soit plus convergeant. Notons qu'un argument ambigu apporte une incohérence au débat car c'est un argument qui défend et attaque la question.

De plus, l'aspect multi-joueur permet d'avoir un peu plus d'objectivité sur le résultat du débat.

Afin d'assurer la convergence sur les graphes ambigus on pourra allouer à chacun des joueurs un intervalle dans lequel ils sont satisfaits. Si la valeur du jeu est dans l'intervalle d'un joueur alors ce joueur ne joue pas. Ainsi, avec un intervalle suffisamment grand, on est sûr que le jeu converge. Est-il alors possible de caractériser la taille minimale de l'intervalle pour que le jeu converge.

On peut également étendre le jeu en ajoutant des poids sur les attaques. Cela permettrait de quantifier le degré d'attaque d'un argument sur un autre. Il serait intéressant de voir si l'ajout de cette possibilité pour les joueurs change réellement la valeur finale du jeu.

7 Remerciements

Je tiens tout d'abord à remercier mes tuteurs de stage, Nicolas Maudet et Élise Bonzon, pour m'avoir conseillé et guidé tout au long du stage. Je voudrais également remercier Fanny Pascual et Zahia Guessoum, deux professeurs du LIP6, pour m'avoir conseillé des articles et donné des pistes. Enfin je voudrais remercier Vasiliki Myrodia ainsi que tous mes co-stagiaires qui m'ont aidé et soutenu pendant toute la durée du stage.

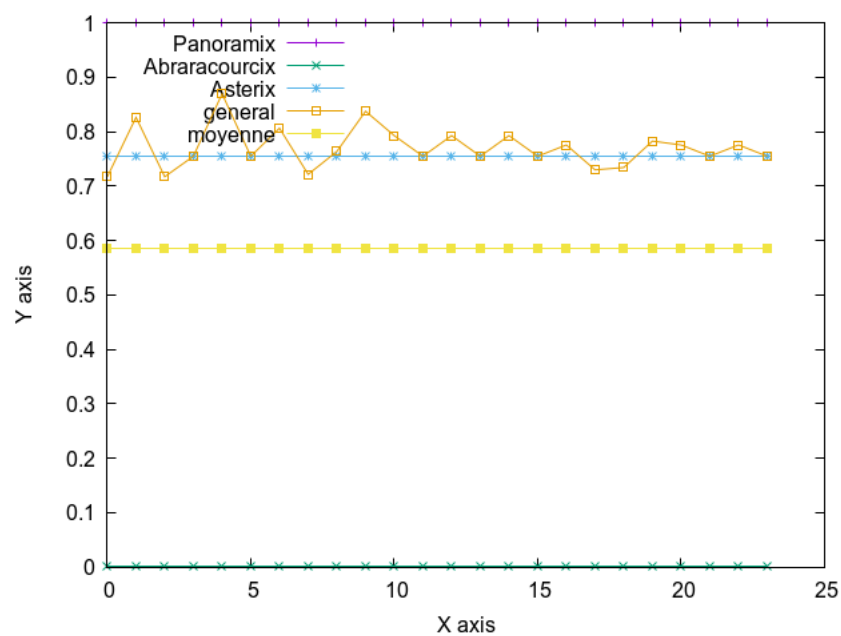
A Bibliographie

Références

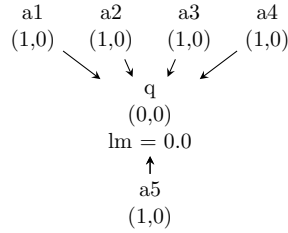
- [1] João LEITE et João MARTINS : Social abstract argumentation. *In Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume Three*, IJCAI'11, pages 2287–2292. AAAI Press, 2011.
- [2] Sinan EGILMEZ, João MARTINS et João LEITE : Extending social abstract argumentation with votes on attacks. *In Theory and Applications of Formal Argumentation - Second International Workshop, TAFA 2013, Beijing, China, August 3-5, 2013, Revised Selected papers*, pages 16–31, 2013.
- [3] Phan Minh DUNG : An argumentation semantics for logic programming with explicit negation. *In Proceedings of the Tenth International Conference on Logic Programming on Logic Programming*, ICLP'93, pages 616–630, Cambridge, MA, USA, 1993. MIT Press.
- [4] René GOSCINNY et Albert UDERZO : *Astérix le Gaulois*. 1961.
- [5] Philippe BESNARD et Anthony HUNTER : A logic-based theory of deductive arguments, 2001.
- [6] Leila AMGOU, Jonathan BEN-NAIM, Dragan DODER et Srdjan VESIC : Ranking arguments with compensation-based semantics. *In Principles of Knowledge Representation and Reasoning : Proceedings of the Fifteenth International Conference, KR 2016, Cape Town, South Africa, April 25-29, 2016.*, pages 12–21, 2016.
- [7] James W. FRIEDMAN et Claudio MEZZETTI : Learning in games by random sampling. *Journal of Economic Theory*, 98(1) :55 – 84, 2001.
- [8] Ehud KALAI et David SCHMEIDLER : An admissible set occurring in various bargaining situations. *Journal of Economic Theory*, 14(2) :402 – 411, 1977.
- [9] Igal MILCHTAICH : Congestion games with player-specific payoff functions. *Games and Economic Behavior*, 13(1) :111 – 124, 1996.

B Détail des coups joués et graphique pour le jeu présenté Figure 3

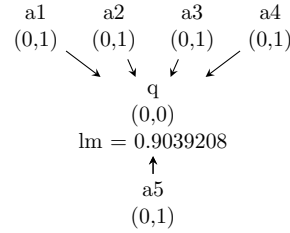
	Panoramix	Abraracourcix	Astérix	général
tour 1	-	-	-	0.71728516
tour 2	a2 like	-	-	0.82620294
tour 3	-	a2 dislike	-	0.71728516
tour 4	-	-	a7 like	0.75487625
tour 5	a1 dislike	-	-	0.86950214
tour 6	-	a1 like	-	0.75487625
tour 7	a5 like	-	-	0.80636351
tour 8	-	a6 like	-	0.72169531
tour 9	-	-	a4 like	0.7636392
tour 10	a6 dislike	-	-	0.837462
tour 11	-	a5 dislike	-	0.79279296
tour 12	-	-	a4 annule	0.75487625
tour 13	a4 like	-	-	0.79279296
tour 14	-	a4 dislike	-	0.75487625
tour 15	a3 dislike	-	-	0.79279296
tour 16	-	a3 like	-	0.75487625
tour 17	a9 like	-	-	0.77565166
tour 18	-	a7 dislike	-	0.72981552
tour 19	-	-	a9 like	0.73410085
tour 20	a7 like	-	-	0.78275675
tour 21	-	a9 dislike	-	0.77565166
tour 22	-	-	a9 annule	0.75487625
tour 23	a8 dislike	-	-	0.77565166
tour 24	-	a8 like	-	0.75487625



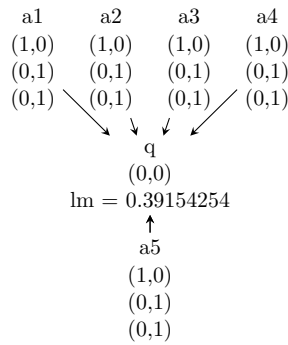
C Exemple de jeu qui va contre la majorité



(a) Graphe du joueur 1



(b) Graphe du joueur 2 et 3



(c) Situation finale

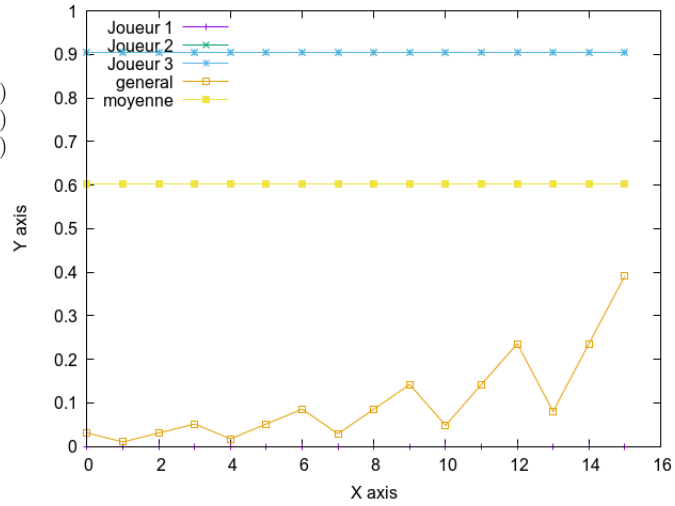


FIGURE 7 – Deux joueurs sont pour et c'est le joueur contre qui gagne.

La valeur finale du jeu est 0.3915. Cette valeur étant plus proche du joueur 1 on peut dire que c'est lui qui remporte le jeu.

D Preuve de la proposition 3.1

Proposition D.1. *La fonction τ est croissante selon $v^+(a)$ et décroissante selon $v^-(a)$.*

Démonstration. On va montrer la croissance (resp. décroissance) selon $v^+(a)$ (resp. $v^-(a)$) pour la fonction τ .

Commençons par remarquer que :

$$\tau(a) = \begin{cases} \frac{1}{2} \cdot \exp(\frac{v^+ - v^-}{\varepsilon}) & \text{si } v^+ - v^- < 0 \\ 1 - \frac{1}{2} \cdot \exp(\frac{v^- - v^+}{\varepsilon}) & \text{sinon} \end{cases}$$

Soit $a_1, a_2 \in \mathcal{A} \setminus \{q\}$ tel que $v^-(a_1) = v^-(a_2)$ et $v^+(a_1) < v^+(a_2)$. Montrons que $\tau(a_1) < \tau(a_2)$

3 cas possibles :

— si $v^+(a_1) - v^- < 0$ et $v^+(a_2) - v^- < 0$

$$\tau(v^+(a_1), v^-) - \tau(v^+(a_2), v^-) = \frac{1}{2} \cdot \exp(\frac{-v^-}{\varepsilon}) (\exp(\frac{v^+(a_1)}{\varepsilon}) - \exp(\frac{v^+(a_2)}{\varepsilon}))$$

\exp est une fonction croissante donc $\tau(v^+(a_1), v^-) < \tau(v^+(a_2), v^-)$

— si $v^+(a_1) - v^- < 0$ et $v^+(a_2) - v^- \geq 0$

$$\tau(v^+(a_1), v^-) - \tau(v^+(a_2), v^-) = \frac{1}{2} \cdot \exp(\frac{v^+(a_1) - v^-}{\varepsilon}) - 1 + \frac{1}{2} \cdot \exp(\frac{v^- - v^+(a_2)}{\varepsilon})$$

En remarquant que $\exp(\frac{v^+(a_1) - v^-}{\varepsilon}) < 1$ et que $\exp(\frac{v^- - v^+(a_2)}{\varepsilon}) \leq 1$ on en conclut que $\tau(v^+(a_1), v^-) < \tau(v^+(a_2), v^-)$

— si $v^+(a_1) - v^- \geq 0$ et $v^+(a_2) - v^- \geq 0$

$$\tau(v^+(a_1), v^-) - \tau(v^+(a_2), v^-) = \frac{1}{2} \cdot \exp(\frac{v^-}{\varepsilon}) (\exp(\frac{-v^+(a_2)}{\varepsilon}) - \exp(\frac{-v^+(a_1)}{\varepsilon}))$$

$\exp(\frac{-v^+(a_2)}{\varepsilon}) < \exp(\frac{-v^+(a_1)}{\varepsilon})$ donc $\tau(v^+(a_1), v^-) < \tau(v^+(a_2), v^-)$

La preuve est similaire pour la décroissance selon $v^-(a)$. \square

E La valeur finale peut être différente de celle de l'unique joueur

Sur cet exemple, la valeur finale est inférieure à celle du joueur.

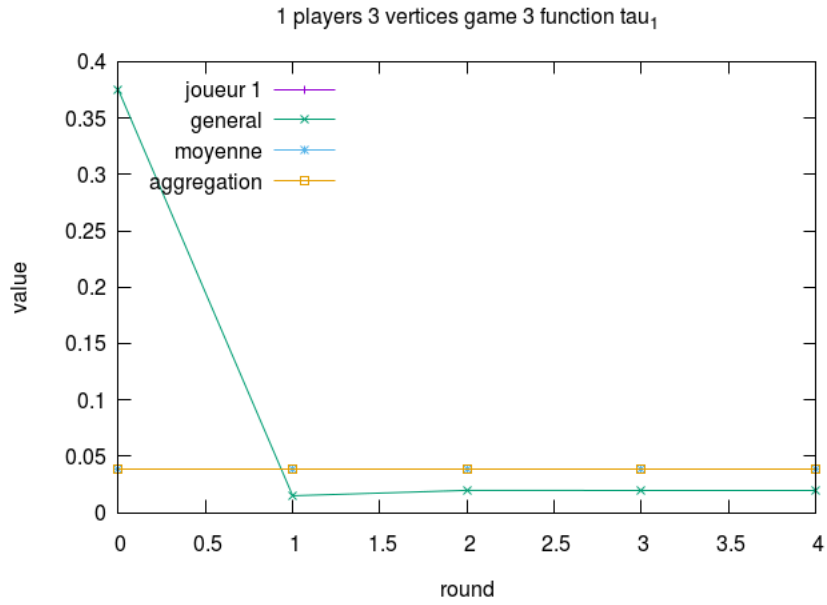
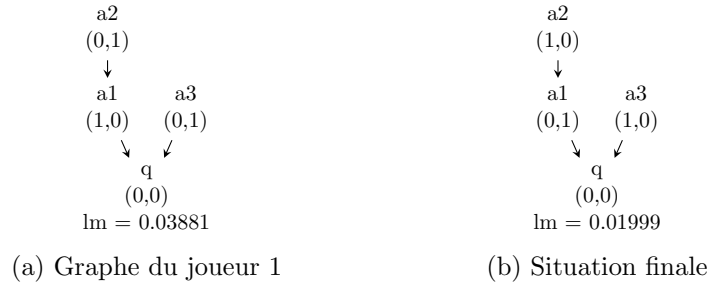
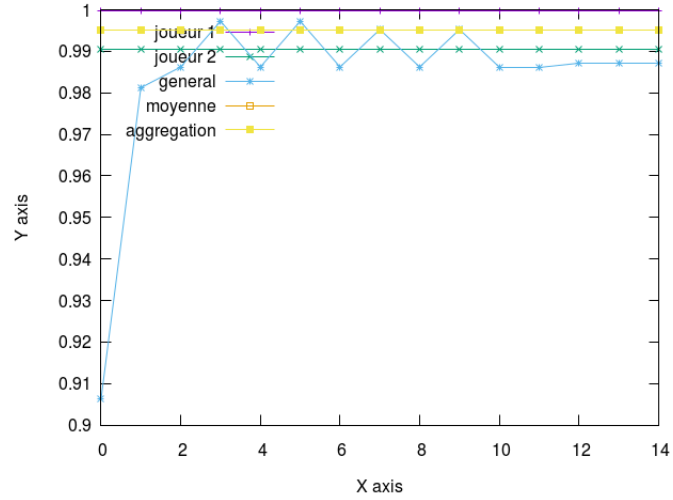
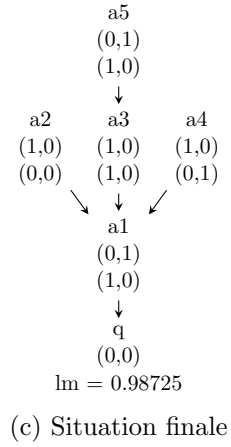
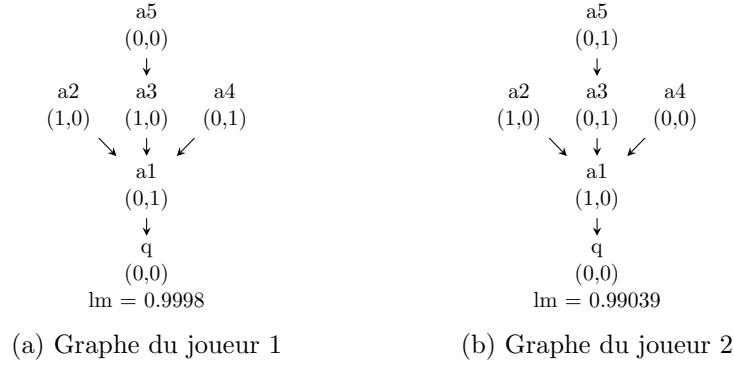


FIGURE 8 – La valeur du jeu est inférieure à celle du joueur

F Équilibre qui ne se trouve pas entre j_{min} et j_{max}

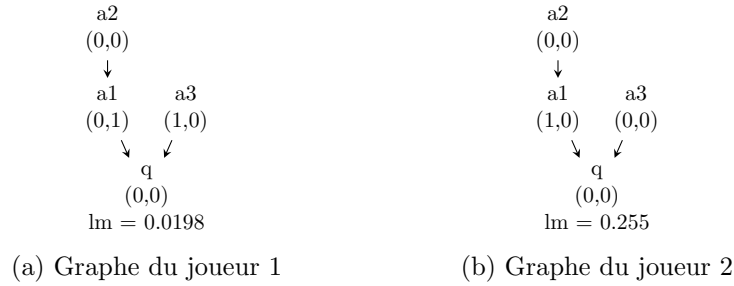
Sur le jeu à 2 joueurs présenté ci-dessous, on voit bien que la valeur finale du jeu est inférieure à j_{min} .

FIGURE 9 – L'état d'équilibre n'est pas entre j_{min} et j_{max}

La valeur finale du jeu est $LM = 0.9872$ alors que la valeur des deux joueurs est $LM_{j_1} = 0.9998$ et $LM_{j_2} = 0.99039$.

G Exemple qui ne possède pas d'équilibre entre j_{min} et j_{max}

La configuration du jeu ci-dessous ne possède pas d'équilibre entre j_{min} et j_{max} . Une énumération complète du jeu à permis d'arriver à cette conclusion.

FIGURE 10 – Aucun état d'équilibre n'existe entre j_{min} et j_{max}

H Jeu avec la dynamique *Random* et *best response* qui contient un cycle

Le jeu n'est pas trivial, c'est un jeu à 5 joueurs sur un graphe à 17 arguments sans compter la question.

Dans ce jeu, les tours 106 et 115 correspondent au même état dans le jeu. Seul deux joueurs sont impliqués dans ce cycle et il met seulement deux arguments en cause.

Voici la situation pour ces deux joueurs sur les deux arguments impliqués juste avant le début du cycle.

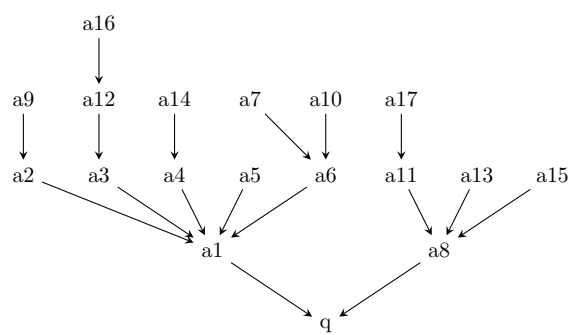
	a6	a7
Joueur 2	-1	0
Joueur 3	0	0

Et voici le cycle.

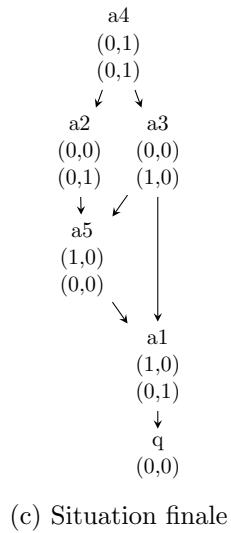
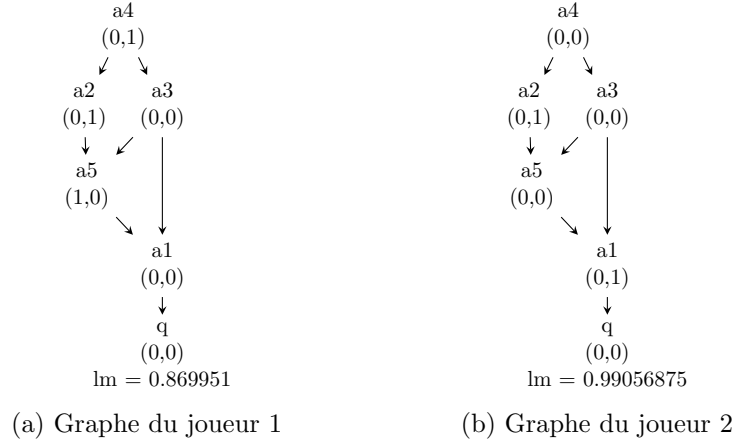
	joueur 1	joueur 2	joueur 3	joueur 4	joueur 5	general
tour 107	-	a7 dislike	-	-	-	0.75586167
tour 108	-	-	a7 dislike	-	-	0.75667651
tour 109	-	-	a6 like	-	-	0.7621542
tour 110	-	-	a7 like	-	-	0.7586651
tour 111	-	a7 annule	-	-	-	0.75431051
tour 112	-	a6 annule	-	-	-	0.75586167
tour 113	-	-	a7 annule	-	-	0.76095258
tour 114	-	-	a6 annule	-	-	0.7586651
tour 115	-	a6 dislike	-	-	-	0.75431051

On remarque bien que la situation à la fin de ce cycle est la même que la situation avant le cycle.

Le graphe du jeu est représenté ci-dessous.



I Jeu sur un graphe acyclique qui possède un cycle



	joueur 1	joueur 2	général
tour 1	-	-	0.73046875
tour 2	a5 like	-	0.83171875
tour 3	-	a1 dislike	0.99326875
tour 4	a1 like	-	0.83171875
tour 5	-	a4 like	0.98044651
tour 6	a4 dislike	-	0.83171875
tour 7	-	a2 dislike	0.89786875
tour 8	a3 like	-	0.87117451
tour 9	-	a4 dislike	0.98981544
tour 10	a3 annule	-	0.87004412
tour 11	-	a3 like	0.98981544
tour 12	a3 dislike	-	0.87004412
tour 13	-	a3 dislike	0.97984011
tour 14	a3 like	-	0.87004412
tour 15	-	a3 annule	0.98981544
tour 16	a3 annule	-	0.87004412
tour 17	-	a3 like	0.98981544

FIGURE 11 – L'état d'équilibre n'est pas entre j_{min} et j_{max}