

# RAPPORT DU PROJET MOSIMA - MARCHÉ DU TRAVAIL SIMPLIFIÉ -

M2 Informatique - Spécialité ANDROIDE

Sorbonne Université



Année universitaire 2019-2020

# 1 Modèle de Base

## 1.1 Etude du modèle proposé

### 1.1.1 Agents-Attributs-Comportements

Comme décrit dans la section 3.1 de l'article, nous pouvons définir 3 type d'agent **PERSON**, **COMPANY** et **MATCHING** possédant les attributs suivants :

Attributs de l'agent **PERSON** :

1. Employed/Unemployed "Status de l'agent"
2. Skills "Compétences de l'agent"
3. Location "Position de l'agent dans le monde"
4. salary "Le salaire cible de l'agent"

Attributs de l'agent **COMPANY** :

1. Job Filled/Vacant "statut de l'unique travail proposé par l'agent)"
2. Skills "Compétences requises pour le travail"
3. Location "Position de l'agent dans le monde"
4. Salary "Le salaire proposé par l'agent"

Attributs de l'agent **MATCHING** :

1. UnemploymentPeople "liste des demandeurs d'emplois"
2. vacancyJobs "liste des proposeurs d'emplois"

Nous rapportons ici les différents comportements tels qu'ils sont définis dans les parties 3.2 et 3.3 de l'article. L'évolution des agents va se baser sur la comparaison de chacun d'entre eux et sur de potentiels appariements en couple employeur/employé. Cependant, l'appariement n'aura lieu que lorsque les deux parties se jugeront respectivement suffisamment similaires. En effet les mesures de similarités entre un agent **PERSON** et un agent **COMPANY** et entre un agent **COMPANY** et un agent **PERSON** ne sont pas symétriques.

A chaque tour de simulation :

- les agents **PERSON** sans emplois se déclareront auprès de l'agent **MATCHING**.
- les agents **COMPANY** sans employés se déclareront auprès de l'agent **MATCHING**, autrement ils vérifieront la productivité de leur employé et le licensiera si besoin.
- L'agent **MATCHING** générera un certain nombre de couple chomeur/recruteur, vérifiera la qualité du couple et validera l'appariement s'il passe les critères.

### 1.1.2 Algorithmes et Boucle ticks

---

#### Algorithme 1 Boucle Principale

---

- 1: **procedure** Go
  - 2:   ask all the agents **PERSON** to execute GO **PERSON**()
  - 3:   ask all the agents **COMPANY** to execute GO **COMPANY**()
  - 4:   ask all the agents **MATCHING** to execute GO **MATCHING**()
- 

---

#### Algorithme 2 Comportement des agents **PERSON**

---

- 1: **procedure** GO **PERSON**
  - 2:   **if** not employed **then**
  - 3:     add the agent to the unemployed list of the agent **MATCHING**
-

---

**Algorithmme 3** Comportement des agents COMPANY

---

```
1: procedure GO COMPANY
2:   if no employee then
3:     add the agent to the recruiter list of the agent MATCHING
4:   else
5:     bad productivity = productivity(employee's skill, job's skill) < firing quality treshold
6:     bad luck = draw random number < unexpected firing
7:     if bad productivity or bad luck then
8:       fire the employee
```

---

---

**Algorithmme 4** Comportement de l'agent MATCHING

---

```
1: procedure GO MATHING
2:   unemployedT = draw randomly nMatch unemployed
3:   recruitersT = draw randomly nMatch recruiters
4:   for all i in [1 ; nMatch] do
5:     i person = unemployedT(i)
6:     i xompany = recruitersT(i)
7:     simiPerson = similiraty person to company(i person, i company)
8:     simiCompany = similiraty company to person(i company, i person)
9:     cEnough = |simiPerson - simiCompany| <= exceptional matching)
10:    gEnough = (simiPerson + simiCompany)/2 >= matching quality treshold)
11:    if cEnough or gEnough then
12:      associate the job of i company to i person
```

---

---

**Algorithmme 5** Productivite d'un agent PERSON pour un travail

---

```
1: procedure PRODUCTIVITY(skill person, skills job)
2:   basic productivity = skillSimilarity(skill person, skill job)
3:   luck = draw randomly in [-(max productivity fluctuation) ; (max. productivity fluctua-
   tion)]
4:   return basic productivity + luck
   =0
```

---

---

**Algorithmme 6** Productivite d'un agent PERSON pour un travail

---

```
1: procedure SKILLSIMILARITY(skill1, skill2)
2:   compteur = 0
3:   n skills = lenght(skill1)
4:   for all i in [1 ; n skills] do
5:     if skills1(i) == skills2(i) then
6:       accumulateur = accumulateur + 1
7:   return compteur / n skills
```

---

---

**Algorithmme 7** Similarité d'un agent COMPANY pour un agent PERSON

---

```
1: procedure SIMILIRATY PERSON TO COMPAN(a person, a company)
2:   compteur = 0
3:   compteur = compteur + skillSimilarity(a person.skill, a company.skills)
4:   compteur = compteur + localisationSimilarity(a person.localisation, a company.location)
5:   compteur = compteur + salarySimilarity(a person.salary, a company.salary)
6:   stochastic = draw randomly in [0 ; (unexpected worker motivation)]
7:   return (compteur + stochastic) / (3 + unexpected worker motivation)
```

---

---

**Algorithme 8** Similarité d'un agent PERSON pour un agent COMPANY

---

```
1: procedure SIMILIRATY PERSON TO COMPAN(a person,a company)
2:   compteur = 0
3:   compteur = compteur + skillSimilarity(a person.skill,a company.skills)
4:   compteur = compteur + localisationSimilarity(a person.localisation,a company.location)
5:   compteur = compteur + salarySimilarity(a company.salary, a person.salary)
6:   stochastic = draw randomly in [0; (unexpected company motivation)]
7:   return (compteur +stochastic) / (3+ unexpected company motivation)
```

---

---

**Algorithme 9** Similarité entre deux localisations

---

```
1: procedure LOCALISATIONSIMILARITY(l1,l2)
2:   (x1,y1) = l1
3:   (x2,y2) = l2
4:   dist =  $\sqrt{(x2 - x1)^2 - (y2 - y1)^2}$ 
5:   return 1 - (dist /distMax)
```

---

---

**Algorithme 10** Similarité entre deux salaire

---

```
1: procedure SALARYSIMILARITY(s1,s2)
2:   diff = s2 - s1
3:   return (1 + (s /sMax))/2
```

---

## 1.2 Programmation du modèle

### 1.2.1 paramètres de la simulation

Certains paramètres sont présentés explicitement dans l'article :

- matching quality treshold : variable servant à savoir quel est le seuil que doit atteindre la moyenne des similarités d'une PERSON et d'une COMPANY.
- firing quality treshold : le seuil de productivité minimal en dela duquel un employé est licencié.
- unexpected firing : pourcentage de se faire licencié "par hasard".
- max. productivity fluctuation : le taux de fluctuation maximum de la productivité.
- unexpected company motivation : variation aléatoire maximum d'une similarité d'agent COMPANY.
- unexpected worker motivation : variation aléatoire maximum d'une similarité d'agent PERSON.
- exceptional matching : difference maximale entre les similarité pour qu'elles soit considérées fiables..
- Person Number : le nombre d'agents PERSON dans la simulation.
- Companies Number : le nombre d'agents COMPANY dans la simulation.
- n match : le nombre de couples testé aléatoirement par l'agent MATCHING.
- salary max fluctuation : la fluctuation maximal de salaire lors de la répartition initiale de l'attribut "salary" chez les agents PERSON et COMPANY.

### 1.2.2 Visualisation des agents et leurs interactions

La répartition de nos agents se fait aléatoirement dans le monde sauf l'agent MATCHING. Les agents PERSON , COMPANY et MATCHING sont représentés (respectivement) par un

personnage, une maison et une cible orange dans l'interface. Si un agent PERSON est lié par un contrat avec l'agent cOMPANY alors un lien est mis en place dans l'interface. On change également la couleur des agents lorsqu'ils sont en recherche (blanc) ou non (bleu). Reciproquement, les COMPANIES qui emploient sont affichées différemment si elle ont un employé (vert) ou non (gris).

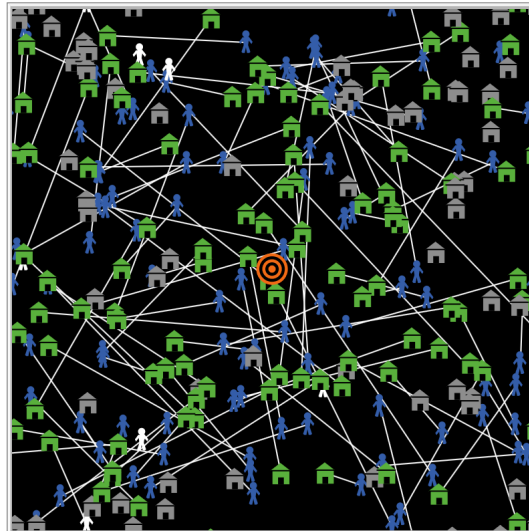


FIGURE 1 – L'interface graphique du monde de la simulation.

### 1.2.3 Reproduction de la courbe de Beveridge

Nous pouvons reproduire la courbe de Beveridge de l'article directement via l'interface en faisant tourner successivement plusieurs simulations de marché du travail. En faisant varier les paramètres de ces simulations (typiquement le nombre d'agents PERSON et COMPANY), on obtient ainsi plusieurs couple de valeurs "vacancy rate" et "unemployment rate". Afficher ces points dans un graphique nous donne alors la courbe de Beveridge. Pour arrêter une simulation nous utilisons un critère de convergence consistant à ne pas trop s'écarter de la valeur d'une moyenne mobile. Comme explicité dans les algorithmes ci dessous, une simulation sera arrêtée dès lors que ses valeurs de "vacancy rate" et "unemployment rate" ne s'écarteront plus (à un epsilon donné) de la valeur de leur moyenne dans une fenetre de temps restreinte.

---

**Algorithme 11** calcul pour la courbe de Beveridge

---

```

1: procedure BEVERIDGE((numberTicksMax, step, Ncompanies, Nperson, epsilon,time
   windows,simulation params))
2:   tabVacancyRate = [ ]
3:   tabUnemploymentRate = [ ]
4:   for all i in NCompanies do
5:     for all j in Nperson do
6:       initialize the simulation with simulation params
7:       tickCount = 0
8:       hasConverged = False
9:       while tickCount < numberTicksMax do
10:        run one tick of the simulation
11:        hasConverged = isConverge(epsilon,time windows)
12:        tickCount = tick count + 1
13:      add VacancyRate of the simulation to tabVacancyRate
14:      add UnemploymentRate of the simulation to tabUnemploymentRate
15:    plotxy (tabVacancyRate,tabUnemploymentRate)

```

---

---

**Algorithme 12** calcul de convergence d'une simulation

---

```
1: procedure BEVERIDGE((numberTicksMax, step, Ncompanies, Nperson, epsilon,time
   windows,simulation params))
2:   if current time > time windows then
3:     Vacancy rate mean = compute the mean Vacancy rate over the time windows [current
   time - time windows ; current time]
4:     Unemployment rate mean = compute the mean Unemployment rate over the time
   windows [current time - time windows ; current time]
5:     isConverge = True
6:     index = 0
7:     while isConverge and index <= time windows do
8:       itime to check = current time - index
9:       convVacancy = |vancancy rate at time(time to check) -vacancy rate mean| <
   epsilon
10:      convUnemployment = |Unemployment rate at time(time to check) - unemploye-
   ment rate mean| < epsilon
11:      isConverge = convVacancy and convUnemployment
12:      index = index + 1
13:      return isConverge
14:     else
15:       return False
```

---

### 1.3 Résultat de la courbe de BEVERIDGE

fichier beveridge.nlogo

Pour la reproduction de la courbe de Beveridge tel que montré dans la figure 2 de l'article , nous avons utilisé les memes paramètres que celle de l'article. à savoir :

- matching quality treshold = 0.5
- firing quality treshold = 0.5
- max productivity fluctuation = 0.3
- unexpected firing = 0.1
- unexpected worker motivation = 0.1
- le nombre d'agents PERSON allant de 100 à 400 avec un pas de 100.
- le nombre d'agents COMPANY allant de 100 à 400 avec un pas de 100.
- comme "exceptionnal matching" n'était pas défini , nous l'avons mis à 1 pour qu'il n'ait pas d'incidence sur la simulation.

Voici les points obtenues avec notre simulation (Figure 2).

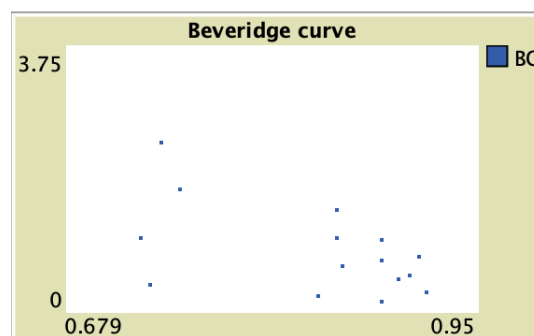


FIGURE 2 – Courbe de BEVERIDGE

## 1.4 efficacité de ce marché du travail

fichier basicmodel.nlogo

Evolution dans le temps du taux de chômage (unemployment rate), et du taux de vacance (vacancy rate) :

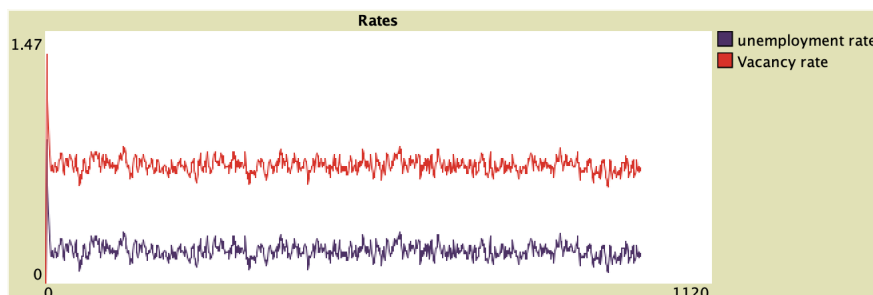


FIGURE 3 – plot du U rate et V rate

Evolutions des taux d'embauches (Hiring Rate) et de licenciements (Firing Rate) :

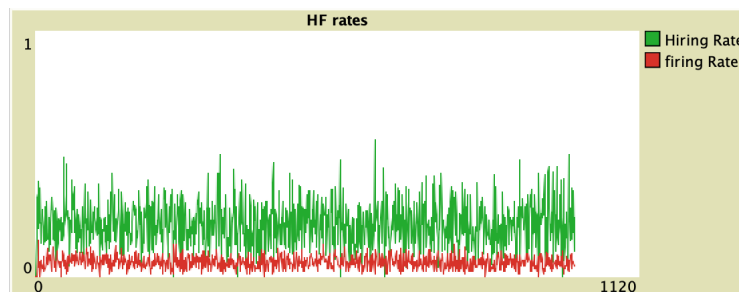


FIGURE 4 – plot du H rate et F rate

Dans nos simulations le taux de chômage est calculé en divisant le nombre de personne sans emploi par le nombre de personnes en activité (c'est à dire recherchant ou ayant un emploi) et le taux de vacancy est calculé en divisant le nombre de postes vacants sur le nombre de personne en activité.

Nous présentons en Figure 3, un exemple d'évolution des courbes du chômage et des postes vacants dans nos simulations. On peut constater que les deux courbes se stabilisent systématiquement après un certain temps, oscillant autour d'une moyenne fixe. C'est ce qui nous a permis dans la question précédente d'utiliser un critère de convergence aussi simple pour arrêter nos simulations.

Nous présentons également en figure 4, un exemple d'évolution de la courbe des taux d'embauche et de licenciement, même remarque de convergence que la figure 3.

Néanmoins, on peut voir que certains paramètres de la simulation influent tout particulièrement sur les valeurs où ces courbes convergent. En particulier, le firing quality threshold et unexpeted firing, on obtient une corrélation positive avec le taux de chômage, de vacancy et de licenciement : plus il est bas et plus le taux de chômage, de vacancy et de licenciement est bas et vice versa. Le hiring rate est très particulièrement influencé par le matching quality threshold, on obtient une corrélation négative car les entreprises deviennent plus exigeantes.

## 1.5 Sensibilité aux paramètres

fichier sensibilityanalysis.nlogo

### 1.5.1 variation du paramètre firing quality treshold

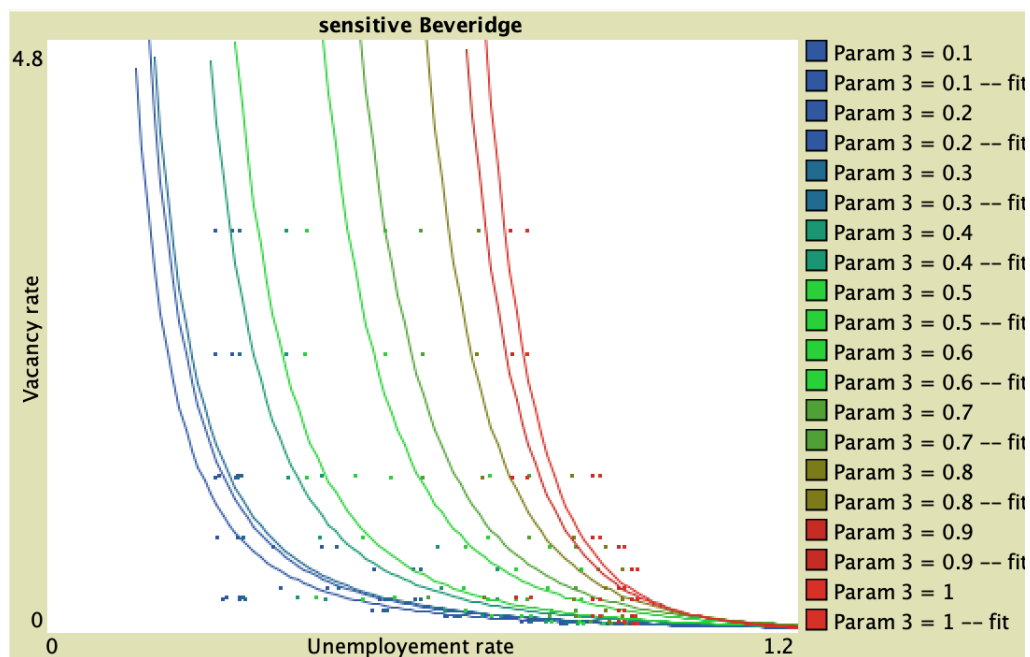


FIGURE 5 – Les différentes courbes de Beveridge en fonction du "firing quality threshold".



FIGURE 6 – evolution du firing et hiring rate en fonction du "firing quality threshold".

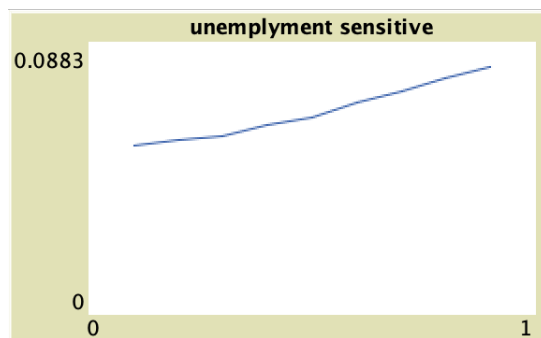


FIGURE 7 – evolution du unemployment rate en fonction du "firing quality threshold".



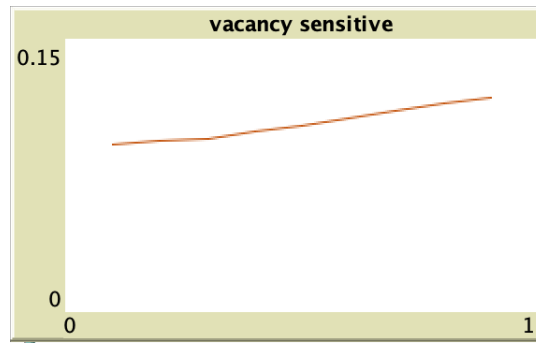


FIGURE 8 – evolution du vacancy rate en fonction du "firing quality threshold".

Dans les Figures 5, 6, 7 et 8 on peut observer que, comme on peut s'y attendre, licencier des gens au hasard n'est pas une bonne idée. Cela augmente le taux de chômage, vacancy et de licenciement, on obtient d'ailleurs une corrélation positive. C'est un paramètre qui ne fait que nuire au marché.

### 1.5.2 variation du paramètre matching quality threshold

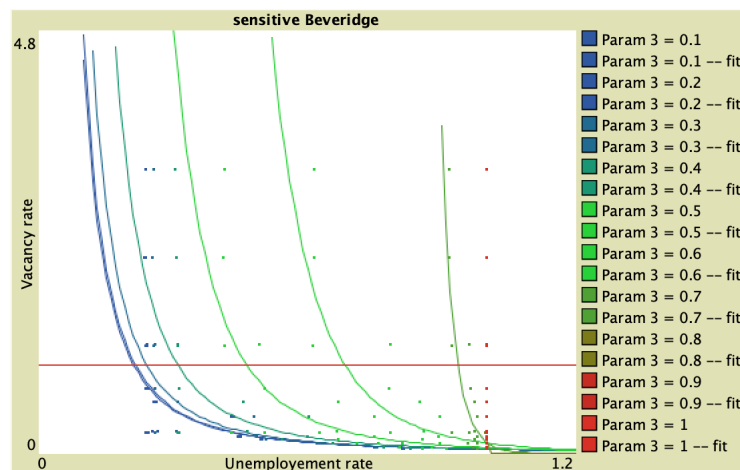


FIGURE 9 – Les différentes courbes de Beveridge en fonction du "matching quality threshold".



FIGURE 10 – evolution du hiring et firing en fonction du "matching quality threshold".

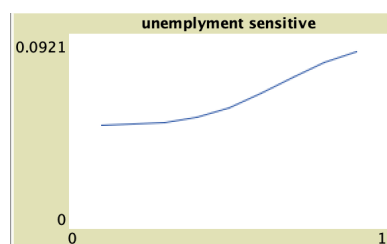


FIGURE 11 – evolution du unemployment rate en fonction du "matching quality threshold".

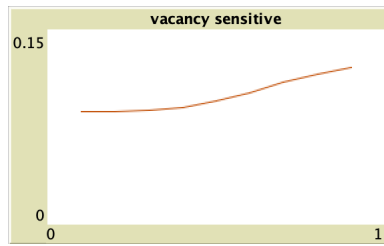


FIGURE 12 – evolution du vacancy rate en fonction du "matching quality threshold".

Dans les Figures 9 ,10, 11 et 12 on peut observer que plus le "matching quality threshold" est élevé, plus la courbe de beveridge est "distante" des axes X et Y . Cela veut donc dire que le taux entre les personnes au chômage et la quantité de postes vacant est élevé. Cela n'améliore pas le marché. On peut voir aussi que le chômage augmente, ce qui rejoint l'analyse des courbes de Beveridge. On peut également constater que le nombre d'embauche chute de plus en plus que le matching quality threshold augmente car les entreprises deviennent de plus en plus exigeantes (similarité ente l'offre et la compétence).

### 1.5.3 variation du paramètre unexpected firing

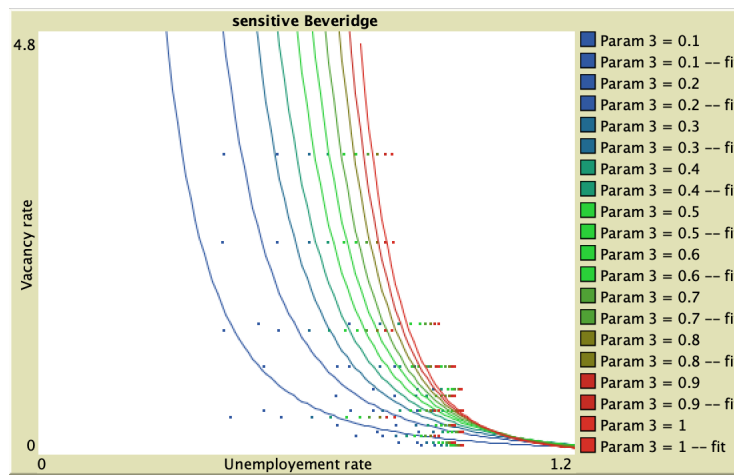


FIGURE 13 – Les différentes courbes de Beveridge en fonction du "unexpected firing".

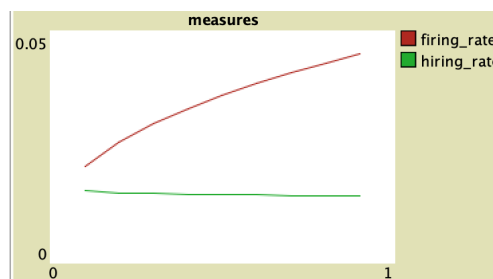


FIGURE 14 – evolution du hiring et firing rate en fonction du "unexpected firing".

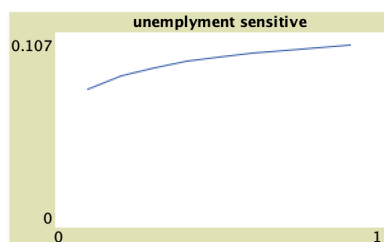


FIGURE 15 – evolution du unemployment rate en fonction du "unexpected firing".

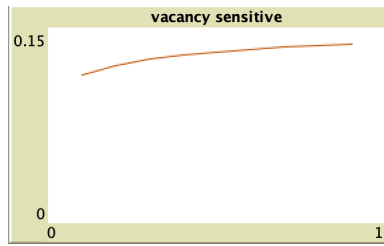


FIGURE 16 – evolution du vacancy rate en fonction du "unexpected firing".

Dans les Figures 17 ,18, 19 et 20 , on peut observer que, comme on peut s'y attendre, augmenter le pourcentage de licenciement des gens au hasard n'est pas une bonne idée. Cela augmente le taux de chômage et augmente de façon exponentiel le firing rate .le taux d'embauche quand à lui est presque constant .

#### 1.5.4 variation du paramètre max productivity fluctuation

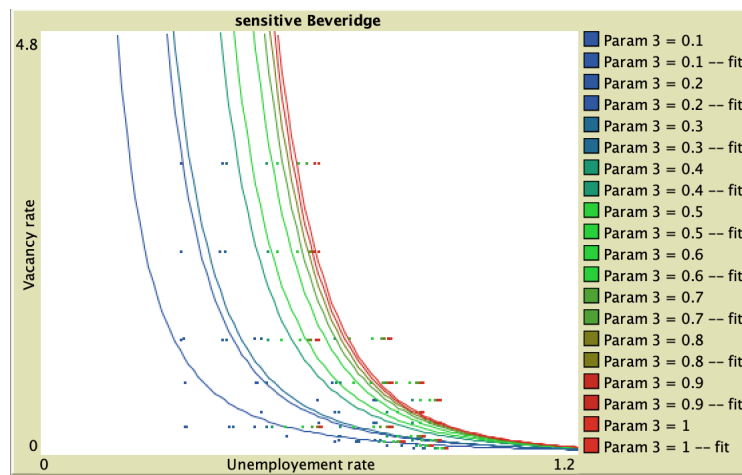


FIGURE 17 – Les différentes courbes de Beveridge en fonction du "max productivity fluctuation".

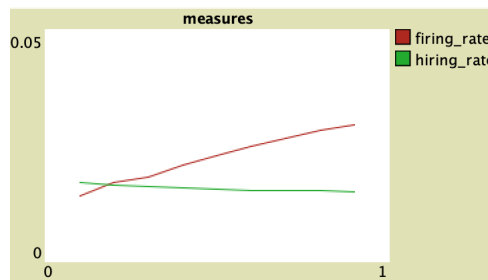


FIGURE 18 – evolution du firing et hiring rate en fonction du "max productivity fluctuation".

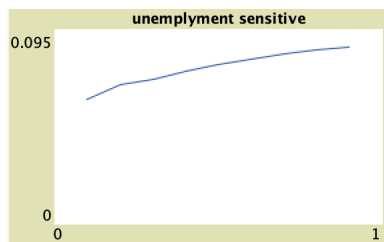


FIGURE 19 – evolution du unemployment rate en fonction du "max productivity fluctuation".

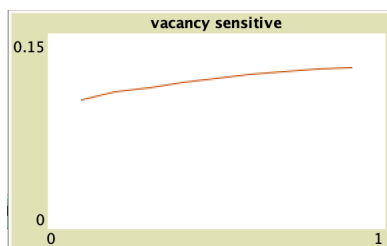


FIGURE 20 – evolution du vacancy rate en fonction du "max productivity fluctuation".

Dans les Figures 13, 14, 15 et 16 on peut observer qu'en faisant varier "max productivity fluctuation", la productivité est plus fluctuante et il y a forcément plus de personnes se faisant licencié pour manque de productivité. On peut alors tirer les mêmes conclusion que pour "l'unexpected firing quality".

### 1.5.5 variation du paramètre unexpected worker motivation

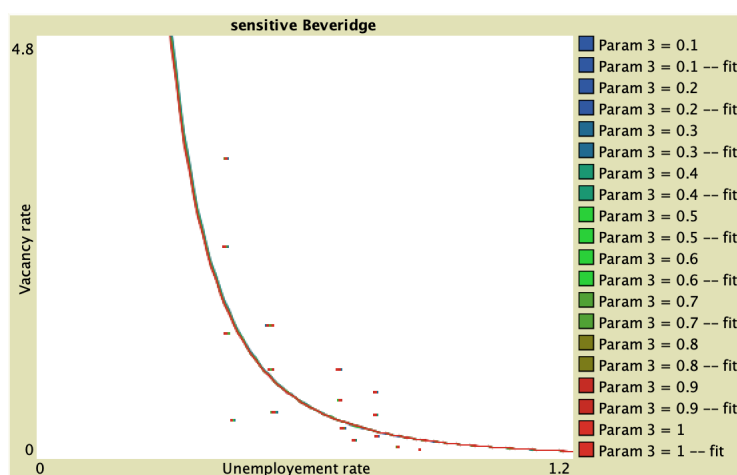


FIGURE 21 – Les différentes courbes de Beveridge en fonction du "unexpected worker motivation".

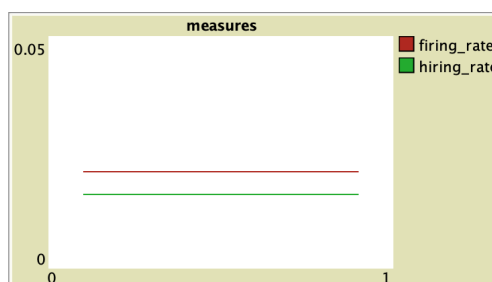


FIGURE 22 – evolution du firing et hiring rate en fonction du "unexpected worker motivation".

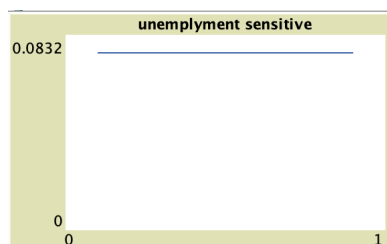


FIGURE 23 – evolution du unemployment rate en fonction du "unexpected worker motivation".

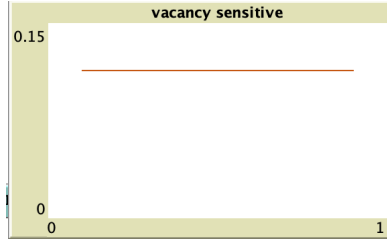


FIGURE 24 – evolution du vacancy rate en fonction du "unexpected worker motivation".

Dans les Figures 22, 23 et 24 on peut observer les évaluations de nos mesures en fonction du paramètre "unexpected worker motivation". Une première hypothèse serait que plus ce paramètre est grand, plus les agents PERSON se font engager et moins il y a de chômage. Néanmoins, on peut observer que ce paramètre a sensiblement pas d'impact sur le modèle. Cela peut s'expliquer par le fait que les agents PERSON extrêmement motivés s'avèrent incompetents et sont donc automatiquement licenciés. Le modèle rejete donc la bonne motivation au profit de l'efficacité.

## 2 Extention du modèle

fichier extendmodel.nlogo

### 2.1 Processus de démission des employés

On va maintenant affiner le système avec l'ajout de démissions des agents PERSON. Nous proposons de se baser sur le "bonheur" d'un agent PERSON pour moduler ses envies de démission. Pour cela, le paramètre UnexpetedResignation est la limite maximale de bonheur qui poussera un agent PERSON a démissionner. Le bonheur sera ici calculé comme pour l'indicateur éponyme, c'est à dire à partir de la valeur de matching d'un agent PERSON couplé avec sa productivité. Sa productivité fluctuant dans le temps, le bonheur d'un agent PERSON l'est tout également.

---

#### Algorithme 13 Optimized Resignation

---

```

1: procedure RESIGN
2:   for all PERSONS do
3:     resign = SIMILARITY PERSON TO COMPANY (himself, his company ) + PRO-
       DUCTIVITY(skill,company.skill)
4:     if resign <= hapinessTreshold then
5:       resign

```

---

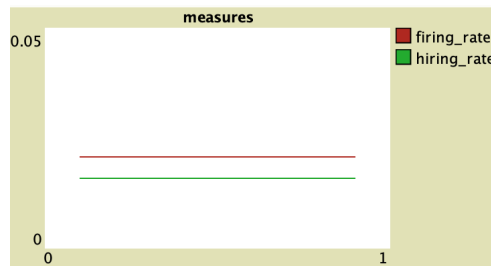


FIGURE 25 – evolution du firing et hiring rate en fonction du "UnexpetedResignation".

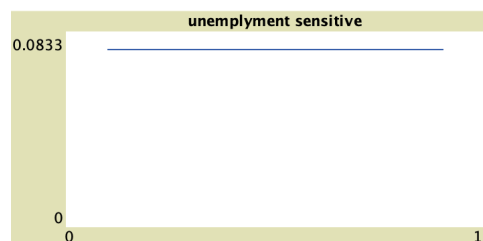


FIGURE 26 – evolution du unemployment rate en fonction du "UnexpetedResignation".

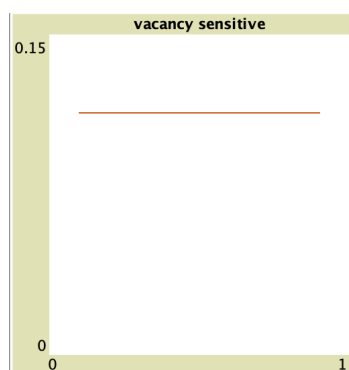


FIGURE 27 – evolution du vacancy rate en fonction du "UnexpetedResignation".

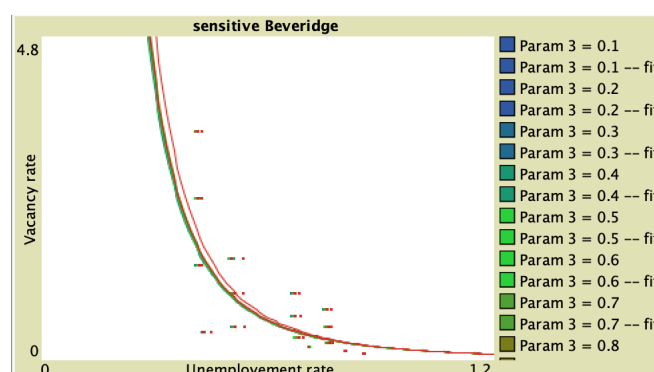


FIGURE 28 – evolution des courbes de beveridge en fonction du "UnexpetedResignation".

On peut remarquer que le paramètre unexpected resignation n'est pas trop pertinents et n'influe pas sur nos mesures . On peut également remarquer que les points utilisées par la courbe de Beveridge sont marginalement plus dispersées. Cela peut s'expliquer par le fait que lorsque les agents PERSON se voient affiliés un travail les "contentant" ils ne démissionnent pas. Les paires convergent ainsi rapidement vers des paires solides qui ne sont brisées que sous l'effet du paramètre "unexpeted firing". 17

## 2.2 Limites du modèle

On peut dégager les limites suivantes dans cette simulation :

- Les comportements des différents agents sont très pauvres et ne représentent pas le comportement de vrai personnes sur le marché du travail. Certains chercheront à changer de travail pour en avoir un qui le satisfera plus, d'autre mettrons des priorités sur leurs critères ex : la distance parcouru devenant plus importante que le salaire si l'agent à des enfants . Pour contrer cela, il faudrait avoir des agents ayant une intelligence plus grande et prendre plus de paramètres à mettre en place.
- III ne représente pas le marché du travail correctement car les entreprises ont pour la grosse majorité plusieurs employés, qu'il existe plusieurs forme de contracts.
- a voir

On peut suggérer pour améliorer la simulation plusieurs propositions :

- **Système de vieillissement dans le temps** : Il serait intéressant de mettre en place le fait que certaines personnes vont à la retraite et que d'autres entrent sur le marché du travail.
- **evolution des compétences dans le temps** : A force de s'exercer, nous dévelopons nos compétences. De même, si nous ne pratiquons pas dans un domaine ,nous perdons peu à

peu nos capacités dans ce domaine. Il serait alors intéressant d'appliquer le même système à nos agents PERSON. Ils deviendraient donc plus fort dans les compétences nécessaire à leur emploi et plus faible dans les compétences non utilisés.

- **La reconversion :** Au cours du temps, Certaines personnes veulent changer de voie professionnel et dans ce but veulent apprendre de nouvelles compétences. De même du coté des entreprises, certaines compétences peuvent devenir obsolètes tandis que d'autre deviennent cruciales. Ces évolutions pourraient facilement être intégrées dans la simulation.

## 2.3 Améliorations suggérées

### 2.3.1 reconversion d'un agent PERSON

Nous avons choisi d'implémenter la reconversion évoquée précédement. Ci dessous est présenté l'algorithme utilisé dans la reconversion des agents PERSON (pouvant aussi être appliqué aux agents COMPANY)

---

#### Algorithme 14 PersonReconverion (reconversion rate)

---

```

1: procedure RESIGN
2:   for all PERSONS do
3:     reconversion? = randomfloat(0,1)
4:     if reconversion ≤ reconversion rate then
5:       ask employer (fire the employee)
6:       ask person (update skills)
```

---

Nous avons tester ce modèle avec la reconversion pour les agents PERSON. En faisant varier "reconversion rate" de 0.1 à 0.9 par pas de 0.1, nous obtenons la Figures 29 et 30. Nous pouvons contraster que la reconversion déséquilibre le marché. En effet, les agents PERSONS ne conviennent plus au poste où ils sont et les postes où ils correspondent ne leur sont pas nécaissairement proposés. Il y a donc une augmentation temporaire du chômage.



FIGURE 29 – evolution du chômage .

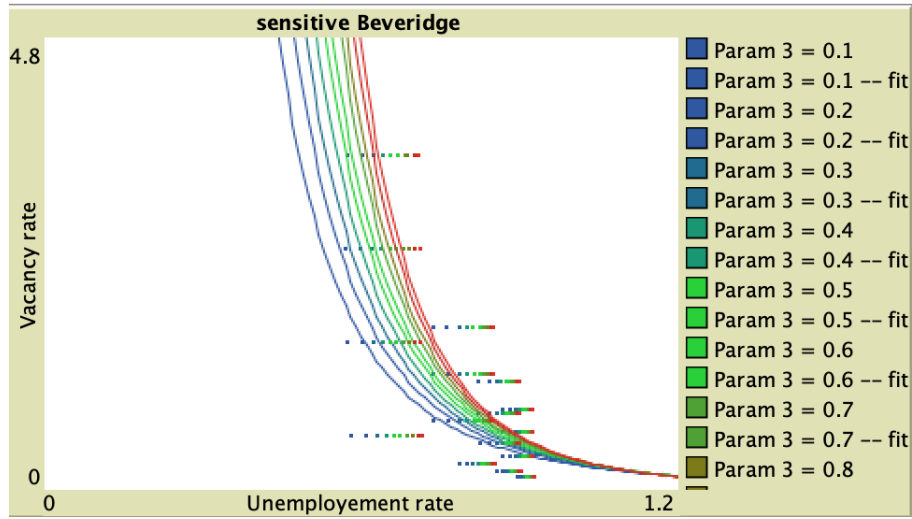


FIGURE 30 – evolution des courbes de beveridge .

### 2.3.2 Préférence des critères d'un agent PERSON

Jusqu'à maintenant nous calculions pour chaque critère une valeur comprise entre 0 et 1 et nous normalisions cette valeur équitablement entre les 3 paramètres (compétences, salaire, distance). Néanmoins dans la réalité, les individus ont des préférences non équivalentes. Par exemple certains préfèrent demander plus, même si leurs compétences ne conviennent pas au travail proposer. Nous proposons donc de pondérer les différents critères afin de donner plus ou moins d'importance à un critère plutôt qu'un autre.

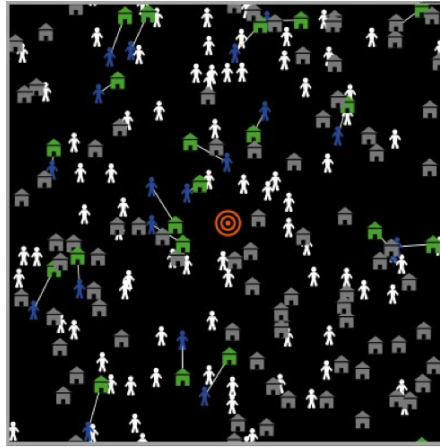


FIGURE 31 – résultat avec le matching optimisé.

Résultat : Visuellement, nous pouvons voir dans la figure 25 que si on privilégie la distance comme critère au dépend des autres, alors les agents préféreront travailler exclusivement avec leurs voisins. Ainsi seul de courts liens sont présents dans la visualisation du monde, donnant lieu à une simulation que nous jugeons plus plausible.