

MODELE PRINCIPAL-AGENT

Le modèle Principal-Agent étudie les relations de délégation où un principal confie une tâche à un agent, en présence d'asymétrie d'information. Le principal ne peut pas observer directement l'effort fourni par l'agent, ce qui crée un problème d'aléa moral (moral hazard). Cette situation est omniprésente dans les relations économiques : employeur-salarié, actionnaire-manager, assuré-assureur, etc.

L'objectif de ce notebook est de résoudre numériquement le modèle principal-agent.

Import et configuration

```
In [226... import numpy as np
from scipy.optimize import minimize
import matplotlib.pyplot as plt
from typing import Dict, Tuple, List

# Configuration pour les graphiques
plt.style.use('seaborn-v0_8-darkgrid')
plt.rcParams['figure.figsize'] = (12, 6)
plt.rcParams['font.size'] = 10
```

Classe principale du modèle

Classe `PrincipalAgentModel`

Rôle général de la classe

La classe `PrincipalAgentModel` implémente un modèle standard de **principal-agent avec aléa moral**. Elle regroupe l'ensemble des paramètres économiques du modèle (profits, probabilités, désutilités, utilité de réserve et aversion au risque) ainsi que les fonctions fondamentales permettant de calculer l'utilité de l'agent et le profit du principal. Cette classe constitue le **cœur du modèle**, sur lequel reposent toutes les résolutions (first-best, second-best et vérification de l'équilibre).

Arguments

- `q_bar` : profit du principal en cas de succès.
- `q` : profit du principal en cas d'échec.
- `pi_H` : probabilité de succès lorsque l'agent fournit l'effort élevé.
- `pi_B` : probabilité de succès lorsque l'agent fournit l'effort faible.
- `d_H` : désutilité associée à l'effort élevé.
- `d_B` : désutilité associée à l'effort faible.
- `u_r` : utilité de réserve de l'agent.
- `gamma` : coefficient d'aversion au risque de l'agent.

Sortie

- Aucune valeur retournée.
- Les paramètres sont stockés comme attributs de l'objet.

Interprétation économique Ces paramètres définissent l'environnement informationnel et technologique du contrat. Les assertions garantissent que le modèle respecte les hypothèses standards de la théorie du principal-agent.

Fonction `utility_agent`

Description Cette fonction définit la **fonction d'utilité instantanée** de l'agent en fonction du salaire perçu.

Arguments

- `w` : salaire reçu par l'agent.

Sortie

- Valeur de l'utilité instantanée ($u(w)$).

Forme fonctionnelle

$$u(w) = w^\gamma$$

Interprétation économique La concavité de la fonction ($\gamma < 1$) traduit l'aversion au risque de l'agent. Cette fonction est utilisée pour calculer l'utilité espérée associée à un contrat donné.

Fonction `expected_utility_agent`

Description Cette fonction calcule l'**utilité espérée nette de l'agent**, en tenant compte de l'incertitude sur le résultat et de la désutilité de l'effort.

Arguments

- `w_bar` : salaire en cas de succès.
- `w` : salaire en cas d'échec.
- `effort` : niveau d'effort choisi par l'agent (`'H'` ou `'B'`).

Sortie

- Utilité espérée nette de l'agent.

Formule implémentée

$$U_A = \pi_e u(\bar{w}) + (1 - \pi_e) u(w) - d(e)$$

Interprétation économique Cette fonction est centrale pour vérifier :

- la **contrainte de participation** (acceptation du contrat),

- la **contrainte d'incitation** (choix optimal de l'effort).

Fonction `expected_profit_principal`

Description Cette fonction calcule le **profit espéré du principal** en fonction du contrat proposé et de l'effort fourni.

Arguments

- `w_bar` : salaire versé en cas de succès.
- `w` : salaire versé en cas d'échec.
- `effort` : effort fourni par l'agent (`'H'` ou `'B'`).

Sortie

- Profit espéré du principal.

Formule implémentée $\Pi = \pi_e (\bar{q} - \bar{w}) + (1-\pi_e)(q - w)$

Interprétation économique Cette fonction permet d'évaluer la rentabilité du contrat pour le principal sous chaque régime informationnel.

In [227...

```
class PrincipalAgentModel:
    """
    Modèle Principal-Agent avec aléa moral

    Notations conformes au document:
    - q_bar: profit en cas de succès
    - q: profit en cas d'échec
    - w_bar: salaire en cas de succès
    - w: salaire en cas d'échec
    - pi_H: probabilité de succès avec effort élevé H
    - pi_B: probabilité de succès avec effort faible B
    - d_H: désutilité de l'effort élevé
    - d_B: désutilité de l'effort faible
    - u_r: utilité de réserve de l'agent
    """

    def __init__(self, q_bar=100, q=20, pi_H=0.8, pi_B=0.4,
                  d_H=15, d_B=5, u_r=10, gamma=0.5):
        """
        Initialisation des paramètres du modèle

        Args:
            q_bar: profit en cas de succès
            q: profit en cas d'échec
            pi_H: probabilité de succès avec effort élevé
            pi_B: probabilité de succès avec effort faible
            d_H: désutilité effort élevé
            d_B: désutilité effort faible
            u_r: utilité de réserve
            gamma: coefficient d'aversion au risque (0 < gamma ≤ 1)
        """

        # Vérifications des paramètres
        assert q_bar > q, "Le profit en cas de succès doit être > profit en cas
        assert 0 < pi_B < pi_H < 1, "Condition: 0 < pi_B < pi_H < 1"
```

```

assert d_H > d_B >= 0, "Le coût de l'effort élevé doit être > effort fai
assert 0 < gamma <= 1, "Coefficient d'aversion au risque: 0 < gamma ≤ 1"

self.q_bar = q_bar
self.q = q
self.pi_H = pi_H
self.pi_B = pi_B
self.d_H = d_H
self.d_B = d_B
self.u_r = u_r
self.gamma = gamma

def utility_agent(self, w: float) -> float:
    """
    Fonction d'utilité de l'agent:  $u(w) = w^\gamma$ 

    Args:
        w: salaire
    Returns:
        Utilité de l'agent
    """
    return w**self.gamma if w >= 0 else -np.inf

def expected_utility_agent(self, w_bar: float, w: float, effort: str) -> flo
    """
    Utilité espérée de l'agent:  $E[u(w(y))]$  - d_e

    Args:
        w_bar: salaire en cas de succès
        w: salaire en cas d'échec
        effort: 'H' (élevé) ou 'B' (faible)
    Returns:
        Utilité espérée de l'agent
    """
    pi = self.pi_H if effort == 'H' else self.pi_B
    d = self.d_H if effort == 'H' else self.d_B

    return pi * self.utility_agent(w_bar) + (1 - pi) * self.utility_agent(w)

def expected_profit_principal(self, w_bar: float, w: float, effort: str) ->
    """
    Profit espéré du principal:  $E[q(y) - w(y)]$ 

    Args:
        w_bar: salaire en cas de succès
        w: salaire en cas d'échec
        effort: 'H' ou 'B'
    Returns:
        Profit espéré du principal
    """
    pi = self.pi_H if effort == 'H' else self.pi_B

    return pi * (self.q_bar - w_bar) + (1 - pi) * (self.q - w)

```

Résolution du first-best



Fonction solve_first_best

Description générale

La fonction `solve_first_best` résout le problème du principal dans le cas **first-best**, c'est-à-dire lorsque l'effort de l'agent est parfaitement observable. Dans ce contexte, le principal peut imposer directement le niveau d'effort et offrir une **assurance complète** à l'agent.

Arguments

- `model` : instance de la classe `PrincipalAgentModel`.
- `target_effort` : effort à imposer ('H' ou 'B').

Sorties

La fonction retourne un dictionnaire contenant :

- `effort` : effort imposé,
- `w_bar` : salaire en cas de succès,
- `w` : salaire en cas d'échec,
- `profit_principal` : profit espéré du principal,
- `utility_agent` : utilité nette de l'agent,
- `type` : type de régime (`First-Best`).

Logique économique

Dans le first-best :

- le principal impose l'effort optimal,
- la contrainte de participation est **saturée**,
- le salaire est identique dans tous les états ($\bar{w} = w$),
- l'agent reçoit exactement son utilité de réserve.

L'utilité de l'agent ne varie donc pas et reste égale à (u_r), tandis que le salaire s'ajuste pour compenser la désutilité de l'effort.

In [228...

```
def solve_first_best(model: PrincipalAgentModel, target_effort: str = 'H', verbose: bool = False):  
    """  
    Résolution du cas first-best (effort observable)  
  
    Le principal peut imposer directement l'effort et assurer complètement  
    l'agent ( $w_{\text{bar}} = w$ ) car il est averse au risque.  
  
    Args:  
        model: instance du modèle  
        target_effort: 'H' ou 'B' - effort à imposer  
    Returns:  
        dict avec contrat optimal et profits  
    """  
    print("\n" + "-"*50)  
    print("Résolution du first best (Information Symétrique)")  
    print("-"*50)
```

```

d = model.d_H if target_effort == 'H' else model.d_B
pi = model.pi_H if target_effort == 'H' else model.pi_B

# Salaire optimal: w_bar = w (assurance complète)
# Contrainte de participation saturée: u(w) - d = u_r
w_optimal = (model.u_r + d)**(1/model.gamma)
w_bar_optimal = w_optimal

# Profit espéré du principal
profit_principal = model.expected_profit_principal(
    w_bar_optimal, w_optimal, target_effort
)

# Utilité de l'agent
utility_agent = model.expected_utility_agent(
    w_bar_optimal, w_optimal, target_effort
)

print(f"\nEffort imposé: {target_effort}")
print(f"Probabilité de succès:  $\pi_{\{target\_effort\}} = \{pi:.2f\}")
print(f"\nContrat optimal (assurance complète):")
print(f"   $\bar{w}$  (succès) = w (échec) = {w_optimal:.2f}")
print(f"\nRésultats:")
print(f"  Profit espéré Principal: {profit_principal:.2f}")
print(f"  Utilité Agent: {utility_agent:.2f} (= u_r = {model.u_r:.2f})")

# Comparaison des efforts
if target_effort == 'H':
    w_B = (model.u_r + model.d_B)**(1/model.gamma)
    profit_B = model.expected_profit_principal(w_B, w_B, 'B')

    print(f"\nComparaison:")
    print(f"  Si effort H imposé: Profit = {profit_principal:.2f}")
    print(f"  Si effort B imposé: Profit = {profit_B:.2f}")

    if profit_principal > profit_B:
        print(f"  Le principal préfère imposer l'effort H")
    else:
        print(f"  Le principal préfère imposer l'effort B")

return {
    'effort': target_effort,
    'w_bar': w_bar_optimal,
    'w': w_optimal,
    'profit_principal': profit_principal,
    'utility_agent': utility_agent,
    'type': 'First-Best'
}$ 
```

Résolution du second-best

Fonction `solve_second_best`

Description générale

La fonction `solve_second_best` résout le problème du principal dans le cas **second-best**, où l'effort de l'agent n'est pas observable. Le principal doit alors concevoir un contrat incitatif respectant simultanément la contrainte de participation et la contrainte d'incitation.

Arguments

- `model` : instance de `PrincipalAgentModel`.

Sorties

La fonction retourne un dictionnaire contenant :

- `w_bar` : salaire en cas de succès,
- `w` : salaire en cas d'échec,
- `profit_principal` : profit espéré du principal,
- `utility_agent_H` : utilité de l'agent s'il choisit l'effort élevé,
- `utility_agent_B` : utilité de l'agent s'il choisit l'effort faible,
- `CP` : valeur de la contrainte de participation,
- `CI` : valeur de la contrainte d'incitation,
- `type` : type de régime (`Second-Best`).

Logique économique

Dans le second-best :

- le principal introduit une **variabilité des salaires** ($\bar{w} > w$),
- l'agent supporte un risque pour être incité à fournir l'effort élevé,
- les contraintes sont généralement **saturées**,
- un **coût d'agence** apparaît par rapport au first-best.

In [229...

```
def solve_second_best(model: PrincipalAgentModel, verbose=True) -> Dict:
    """
    Résolution du cas second-best (aléa moral - effort non observable)

    Le principal doit satisfaire:
    1. Contrainte de Participation (CP):  $EU_A(H) \geq u_r$ 
    2. Contrainte d'Incitation (CI):  $EU_A(H) \geq EU_A(B)$ 

    Args:
        model: instance du modèle
    Returns:
        dict avec contrat optimal et profits
    """
    print("\n" + "-"*40)
    print("Résolution du second best (Aléa Moral)")
    print("-"*40)

    def objective(x):
        """Fonction à minimiser: -Profit du Principal"""
        w_bar, w = x
        if w_bar < 0 or w < 0:
```

```

        return 1e10
    return -model.expected_profit_principal(w_bar, w, 'H')

def constraint_participation(x):
    """Contrainte de Participation (CP):  $EU_A(H) - u_r \geq 0$ """
    w_bar, w = x
    return model.expected_utility_agent(w_bar, w, 'H') - model.u_r

def constraint_incentive(x):
    """Contrainte d'Incitation (CI):  $EU_A(H) - EU_A(B) \geq 0$ """
    w_bar, w = x
    eu_H = model.expected_utility_agent(w_bar, w, 'H')
    eu_B = model.expected_utility_agent(w_bar, w, 'B')
    return eu_H - eu_B

# Contraintes
constraints = [
    {'type': 'ineq', 'fun': constraint_participation},
    {'type': 'ineq', 'fun': constraint_incentive}
]

# Point initial
w_init = (model.u_r + model.d_H)**(1/model.gamma)
x0 = [w_init * 1.5, w_init * 0.8]

# Optimisation
result = minimize(
    objective,
    x0,
    method='SLSQP',
    bounds=[(0, None), (0, None)],
    constraints=constraints,
    options={'ftol': 1e-9, 'maxiter': 1000}
)

if not result.success:
    print("Attention: L'optimisation n'a pas convergé!")
    print(f"Message: {result.message}")

w_bar_opt, w_opt = result.x

# Calcul des résultats
profit_principal = model.expected_profit_principal(w_bar_opt, w_opt, 'H')
utility_agent_H = model.expected_utility_agent(w_bar_opt, w_opt, 'H')
utility_agent_B = model.expected_utility_agent(w_bar_opt, w_opt, 'B')

# Vérification des contraintes
cp_slack = utility_agent_H - model.u_r
ci_slack = utility_agent_H - utility_agent_B

print(f"\nContrat optimal:")
print(f"   $\bar{w}$  (succès) = {w_bar_opt:.2f}")
print(f"  w (échec) = {w_opt:.2f}")
print(f"  Écart  $\bar{w} - w = {w_bar_opt - w_opt:.2f}$ ")

print(f"\nRésultats:")
print(f"  Profit espéré Principal: {profit_principal:.2f}")
print(f"  Utilité Agent (si e=H): {utility_agent_H:.2f}")
print(f"  Utilité Agent (si e=B): {utility_agent_B:.2f}")

```



```

print(f"\nVérification des contraintes:")
print(f"  CP ( $\geq 0$ ): {cp_slack:.4f} {' contrainte saturée' if abs(cp_slack) < 0.0001}")
print(f"  CI ( $\geq 0$ ): {ci_slack:.4f} {' contrainte saturée' if abs(ci_slack) < 0.0001}")

print(f"\nInterprétation:")
if w_bar_opt > w_opt:
    print(f"  L'agent est incité à fournir l'effort élevé H")
    print(f"  Mais il supporte du risque ( $\bar{w} \neq w$ )")

return {
    'effort': 'H',
    'w_bar': w_bar_opt,
    'w': w_opt,
    'profit_principal': profit_principal,
    'utility_agent': utility_agent_H,
    'type': 'Second-Best',
    'cp_slack': cp_slack,
    'ci_slack': ci_slack
}

```

Vérification de l'équilibre parfait bayésien

Fonction de vérification du PBE

Description générale

La fonction de vérification du PBE teste si le contrat issu du second-best constitue un **équilibre bayésien parfait**. Elle vérifie successivement la rationalité de l'agent, la cohérence des croyances du principal et l'absence de déviation profitable.

```

In [230... def verify_perfect_bayesian_equilibrium(model: PrincipalAgentModel,
                                         contract: Dict) -> Dict:
    """
    Vérifie que la solution constitue un Équilibre Bayésien Parfait (PBE)

    Un PBE requiert:
    1. Optimalité séquentielle: chaque joueur optimise à chaque nœud de décision
    2. Croyances bayésiennes: cohérentes avec les stratégies d'équilibre
    3. Pas de déviation profitable

    Args:
        model: instance du modèle
        contract: dictionnaire avec w_bar et w

    Returns:
        dict avec résultats de vérification
    """
    print("\n" + "-"*50)
    print("Vérification de l'équilibre bayésien parfait (PBE)")
    print("-"*50)

    w_bar = contract['w_bar']
    w = contract['w']

    # ==== 1. RATIONALITÉ SÉQUENTIELLE DE L'AGENT ====
    print("\n1. Vérification de la rationalité de l'Agent:")

```

```

# Utilités pour chaque effort
eu_H = model.expected_utility_agent(w_bar, w, 'H')
eu_B = model.expected_utility_agent(w_bar, w, 'B')
eu_reject = model.u_r # Utilité si refus

print(f"    EU_A(accepter, e=H) = {eu_H:.4f}")
print(f"    EU_A(accepter, e=B) = {eu_B:.4f}")
print(f"    EU_A(refuser)          = {eu_reject:.4f}")

# Décision de participation
accept_contract = max(eu_H, eu_B) >= eu_reject
print(f"\n    → L'agent {'Accepte' if accept_contract else 'Refuse'} le contrat")

# Choix d'effort optimal
if accept_contract:
    optimal_effort = 'H' if eu_H >= eu_B else 'B'
    print(f"    Si accepté, effort optimal: {optimal_effort}")

    ic_satisfied = eu_H >= eu_B
    print(f"    Contrainte d'incitation: {'SATISFAITE' if ic_satisfied else 'V"}

# ==== 2. CROYANCES DU PRINCIPAL ====
print("\n2. Croyances du Principal (après observation du résultat):")

# Probabilités a priori
print(f"    Croyances a priori (avant observation):")
print(f"    P(e=H | contrat proposé) = 1 (anticipé)")
print(f"    P(e=B | contrat proposé) = 0")

# Mise à jour bayésienne après observation du succès
if accept_contract and optimal_effort == 'H':
    # Probabilité de succès observé sachant e=H
    p_success_given_H = model.pi_H
    # Probabilité de succès observé sachant e=B
    p_success_given_B = model.pi_B

    # Règle de Bayes:  $P(H|S) = P(S|H)*P(H) / P(S)$ 
    # Si le principal anticipe correctement e=H
    prob_H_given_success = (p_success_given_H * 1.0) / p_success_given_H
    prob_H_given_failure = ((1-p_success_given_H) * 1.0) / (1-p_success_give

    print(f"\n    Mise à jour bayésienne:")
    print(f"    P(e=H | Succès observé) = {prob_H_given_success:.4f}")
    print(f"    P(e=H | Échec observé) = {prob_H_given_failure:.4f}")
    print(f"    Croyances cohérentes avec stratégie d'équilibre")

# ==== 3. OPTIMALITÉ DU PRINCIPAL ====
print("\n3. Vérification de l'optimalité du Principal:")

profit_equilibrium = model.expected_profit_principal(w_bar, w, optimal_effort)
print(f"    Profit à l'équilibre: {profit_equilibrium:.4f}")

# Test de déviations possibles
print(f"\n    Test de déviations:")

# Déviation 1: offrir un contrat qui induit e=B
w_B_deviation = (model.u_r + model.d_B)**(1/model.gamma)
profit_deviation_B = model.expected_profit_principal(w_B_deviation, w_B_deviation)
print(f"    Déviation vers e=B: Profit = {profit_deviation_B:.4f}")

```

```

# Déviation 2: ne pas proposer de contrat
profit_no_contract = 0
print(f"      Ne pas contracter: Profit = {profit_no_contract:.4f}")

no_profitable_deviation = (profit_equilibrium >= profit_deviation_B) and (pr
print(f"\n    Pas de déviation profitable: {'OUI' if no_profitable_deviation

# ==== 4. TEST DE ROBUSTESSE ====
print("\n4. Tests de robustesse:")

# Test 1: Si Le principal augmente w_bar
w_bar_test = w_bar * 1.1
eu_H_test = model.expected_utility_agent(w_bar_test, w, 'H')
eu_B_test = model.expected_utility_agent(w_bar_test, w, 'B')
profit_test = model.expected_profit_principal(w_bar_test, w, 'H')

print(f"    Si  $\bar{w}$  augmente de 10%:")
print(f"        CI reste satisfaite: {eu_H_test >= eu_B_test}")
print(f"        Profit Principal: {profit_test:.4f} {'<' if profit_test < profit

# Test 2: Si Le principal diminue w_bar
w_bar_test2 = w_bar * 0.9
eu_H_test2 = model.expected_utility_agent(w_bar_test2, w, 'H')
eu_B_test2 = model.expected_utility_agent(w_bar_test2, w, 'B')

print(f"\n    Si  $\bar{w}$  diminue de 10%:")
print(f"        CI satisfaite: {eu_H_test2 >= eu_B_test2}")
print(f"        CP satisfaite: {eu_H_test2 >= model.u_r}")

# ==== CONCLUSION ====
print("\n" + "-"*50)
print("Conclusion sur l'Équilibre Bayésien Parfait:")
print("-"*50)

is_pbe = accept_contract and ic_satisfied and no_profitable_deviation

if is_pbe:
    print("PBE vérifié")
    print("\nLe contrat proposé constitue un PBE car:")
    print("    1. L'agent accepte et choisit rationnellement e=H")
    print("    2. Les croyances du principal sont cohérentes")
    print("    3. Aucun joueur ne peut dévier de manière profitable")
else:
    print("Pas de PBE")

return {
    'is_pbe': is_pbe,
    'agent_accepts': accept_contract,
    'optimal_effort': optimal_effort if accept_contract else None,
    'ic_satisfied': ic_satisfied if accept_contract else False,
    'no_profitable_deviation': no_profitable_deviation,
    'equilibrium_profit': profit_equilibrium,
    'eu_H': eu_H,
    'eu_B': eu_B
}

```

Comparaison et analyse des deux situations

Analyse de sensibilité

Objectif général

L'analyse de sensibilité vise à étudier la robustesse des résultats du modèle principal-agent face aux variations de certains paramètres clés. Elle permet de comprendre comment les **incitations**, le **coût d'agence** et la **rentabilité du principal** réagissent lorsque l'environnement économique change. Deux paramètres centraux sont analysés :

- l'aversion au risque de l'agent (γ),
- la productivité de l'effort faible (π_B).

Sensibilité à l'aversion au risque de l'agent (γ)

Description

Le paramètre γ mesure le degré d'aversion au risque de l'agent. Plus γ est faible, plus l'agent est averse au risque ; à l'inverse, lorsque γ se rapproche de 1, l'agent devient neutre au risque. L'analyse consiste à faire varier γ sur un intervalle donné et à résoudre successivement les problèmes first-best et second-best pour chaque valeur.

Logique économique

Lorsque l'agent est très averse au risque (γ faible), le principal doit offrir une forte assurance pour satisfaire la contrainte de participation. Cela conduit à des salaires élevés et peut fortement réduire, voire annuler, le profit du principal. Dans ce cas, inciter l'agent à fournir un effort élevé devient coûteux.

À mesure que γ augmente :

- l'agent supporte mieux le risque,
- l'écart entre le salaire en cas de succès et d'échec diminue,
- le coût d'agence décroît progressivement.

Résultats observés

- Pour des valeurs faibles de γ , le profit du principal peut devenir négatif et le principal préfère parfois imposer l'effort faible.
- Lorsque γ augmente, le contrat incitatif devient moins coûteux.
- À la limite $\gamma = 1$ (agent neutre au risque), le second-best coïncide avec le first-best et le coût d'agence devient nul.

Interprétation économique

Cette analyse confirme un résultat fondamental de la théorie du principal-agent : **le coût d'agence est d'autant plus élevé que l'agent est averse au risque**. Lorsque l'agent est

neutre au risque, il est possible de l'inciter sans perte d'efficacité, ce qui élimine toute inefficience liée à l'asymétrie d'information.

Sensibilité à la probabilité de succès sous effort faible (π_B)

Description

La probabilité π_B mesure l'efficacité de l'effort faible. L'analyse de sensibilité consiste à augmenter progressivement π_B tout en maintenant π_H constant, afin d'évaluer comment la proximité entre les deux niveaux d'effort affecte la structure du contrat et le choix du principal.

Logique économique

Lorsque π_B est faible, l'effort élevé apporte un gain productif important par rapport à l'effort faible. Le principal est alors prêt à supporter un coût d'incitation pour encourager l'effort élevé. En revanche, lorsque π_B augmente, la différence entre les deux efforts se réduit, ce qui diminue l'intérêt économique d'inciter l'agent à fournir l'effort élevé.

Résultats observés

- À mesure que π_B augmente, l'écart salarial ($\bar{w} - w$) nécessaire pour inciter l'effort élevé augmente.
- Le profit du principal diminue, reflétant une hausse du coût d'agence.
- Au-delà d'un certain seuil, le principal préfère abandonner l'effort élevé et choisir directement l'effort faible.
- Dans certains cas extrêmes, la contrainte d'incitation devient plus contraignante que la contrainte de participation, ce qui conduit à une **rente informationnelle** pour l'agent.

Interprétation économique

Cette analyse montre que **l'incitation à l'effort élevé n'est rentable que si l'effort faible est suffisamment inefficace**. Lorsque les performances des deux efforts deviennent proches, l'asymétrie d'information engendre une inefficience trop importante, et le principal renonce à l'effort élevé.

Coût d'agence et enseignements généraux

Définition

Le coût d'agence est défini comme la différence de profit entre le régime first-best et le régime second-best :

$$\text{Coût d'agence} = \Pi^{FB} - \Pi^{SB}$$

Résultats globaux

- Le coût d'agence augmente lorsque :
 - l'agent est très averse au risque,
 - l'effort faible devient plus productif.
- Le coût d'agence tend vers zéro lorsque :
 - l'agent est neutre au risque,
 - l'effort élevé domine largement l'effort faible.

Interprétation synthétique

L'analyse de sensibilité met en évidence le compromis central du modèle principal-agent : **inciter un agent sous asymétrie d'information implique nécessairement un coût**, dont l'ampleur dépend à la fois des préférences de l'agent et de la technologie de production.

```
In [231...] def compare_regimes(model: PrincipalAgentModel) -> Tuple[Dict, Dict, float]:
    """
    Compare les résultats First-Best vs Second-Best

    Args:
        model: instance du modèle
    Returns:
        tuple (first_best, second_best, agency_cost)
    """
    print("\n" + "="*50)
    print("Comparaison des régimes First-Best vs Second-Best")
    print("="*50)

    fb = solve_first_best(model, 'H')
    sb = solve_second_best(model)

    # Coût d'agence
    agency_cost = fb['profit_principal'] - sb['profit_principal']

    print(f"\nCoût d'agence (perte due à l'asymétrie d'information):")
    print(f"  Profit FB - Profit SB = {agency_cost:.2f}")
    print(f"  Perte relative: {100 * agency_cost / fb['profit_principal']:.2f}%")

    print(f"\nÉcart de salaire:")
    print(f"  FB:  $\bar{w} - w = \{fb['w\_bar'] - fb['w']\} \cdot 100$  (assurance complète)")
    print(f"  SB:  $\bar{w} - w = \{sb['w\_bar'] - sb['w']\} \cdot 100$  (incitation)")

    return fb, sb, agency_cost
```

Analyse de sensibilité

```
In [232...] def sensitivity_analysis(model: PrincipalAgentModel):
    """
    Analyse de sensibilité: impact des paramètres sur le coût d'agence

    Args:
        model: instance du modèle
```

```

Returns:
    figure matplotlib
"""
print("\n" + "="*40)
print("Analyse de sensibilité du coût d'agence")
print("="*40)

# Sauvegarder paramètres originaux
original_gamma = model.gamma
original_pi_B = model.pi_B

# Analyse 1: Impact de l'aversion au risque (gamma)
gammas = np.linspace(0.3, 1.0, 8)
agency_costs_gamma = []

print("\n1. Impact de l'aversion au risque ( $\gamma$ ):")
for g in gammas:
    model.gamma = g
    fb = solve_first_best(model, 'H')
    sb = solve_second_best(model)
    ac = fb['profit_principal'] - sb['profit_principal']
    agency_costs_gamma.append(ac)
    print(f"     $\gamma = \{g:.2f\}$ : Coût d'agence =  $\{ac:.2f\}$ ")

# Analyse 2: Impact de  $\pi_B$ 
model.gamma = original_gamma
pi_Bs = np.linspace(0.2, 0.7, 8)
agency_costs_pi = []

print("\n2. Impact de  $\pi_B$  (effort faible):")
for pi_b in pi_Bs:
    if pi_b < model.pi_H:
        model.pi_B = pi_b
        fb = solve_first_best(model, 'H')
        sb = solve_second_best(model)
        ac = fb['profit_principal'] - sb['profit_principal']
        agency_costs_pi.append(ac)
        print(f"     $\pi_B = \{pi_b:.2f\}$ : Coût d'agence =  $\{ac:.2f\}$ ")

# Restaurer paramètres
model.gamma = original_gamma
model.pi_B = original_pi_B

# Graphiques
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(14, 5))

ax1.plot(gammas, agency_costs_gamma, 'b-o', linewidth=2, markersize=8)
ax1.set_xlabel('Coefficient d\'aversion au risque ( $\gamma$ )', fontsize=12)
ax1.set_ylabel('Coût d\'agence', fontsize=12)
ax1.set_title('Impact de l\'aversion au risque', fontsize=13, fontweight='bold')
ax1.grid(True, alpha=0.3)
ax1.axhline(y=0, color='r', linestyle='--', alpha=0.5)

ax2.plot(pi_Bs[:len(agency_costs_pi)], agency_costs_pi, 'g-o', linewidth=2, markersize=8)
ax2.set_xlabel('Probabilité de succès avec effort faible ( $\pi_B$ )', fontsize=12)
ax2.set_ylabel('Coût d\'agence', fontsize=12)
ax2.set_title('Impact de la difficulté de surveillance', fontsize=13, fontweight='bold')
ax2.grid(True, alpha=0.3)
ax2.axhline(y=0, color='r', linestyle='--', alpha=0.5)

```

```
plt.tight_layout()
```

```
return fig
```

Exemple d'utilisation

In [233...

```
# 1. Création du modèle
model = PrincipalAgentModel(
    q_bar=600,      # Profit si succès
    q=100,          # Profit si échec
    pi_H=0.7,       # pi_B=0.7 Proba succès si effort faible pi_H=0.7 donne Le PB
    pi_B=0.3,       # pi_B=0.3 Proba succès si effort faible pi_B=0.3 donne Le PB
    d_H=5,          # Coût effort élevé
    d_B=2,          # Coût effort faible
    u_r=10,         # Utilité de réserve
    gamma=0.5       # Aversion au risque
)

# 2. Résoudre First-Best
fb = solve_first_best(model, 'H')

# 3. Résoudre Second-Best
sb = solve_second_best(model)

# 4. Vérifier l'équilibre bayésien
pbe_results = verify_perfect_bayesian_equilibrium(model, sb)

# 5. Comparer Les régimes
fb, sb, agency_cost = compare_regimes(model)

# 6. Analyse de sensibilité
fig = sensitivity_analysis(model)
plt.show()
```

Résolution du first best (Information Symétrique)

Effort imposé: H

Probabilité de succès: $\pi_H = 0.70$

Contrat optimal (assurance complète):

\bar{w} (succès) = w (échec) = 225.00

Résultats:

Profit espéré Principal: 225.00

Utilité Agent: 10.00 (= $u_r = 10.00$)

Comparaison:

Si effort H imposé: Profit = 225.00

Si effort B imposé: Profit = 106.00

Le principal préfère imposer l'effort H

Résolution du second best (Aléa Moral)

Contrat optimal:

\bar{w} (succès) = 297.56

w (échec) = 95.06

Écart $\bar{w} - w = 202.50$

Résultats:

Profit espéré Principal: 213.19

Utilité Agent (si $e=H$): 10.00

Utilité Agent (si $e=B$): 10.00

Vérification des contraintes:

CP (≥ 0): -0.0000 contrainte saturée

CI (≥ 0): 0.0000 contrainte saturée

Interprétation:

L'agent est incité à fournir l'effort élevé H

Mais il supporte du risque ($\bar{w} \neq w$)

Vérification de l'équilibre bayésien parfait (PBE)

1. Vérification de la rationalité de l'Agent:

$EU_A(\text{accepter}, e=H) = 10.0000$

$EU_A(\text{accepter}, e=B) = 10.0000$

$EU_A(\text{refuser}) = 10.0000$

→ L'agent Refuse le contrat

2. Croyances du Principal (après observation du résultat):

Croyances a priori (avant observation):

$P(e=H \mid \text{contrat proposé}) = 1$ (anticipé)

$P(e=B \mid \text{contrat proposé}) = 0$

3. Vérification de l'optimalité du Principal:

Profit à l'équilibre: 94.1875

Test de déviations:
Déviation vers e=B: Profit = 106.0000
Ne pas contracter: Profit = 0.0000

Pas de déviation profitable: NON

4. Tests de robustesse:
Si \bar{w} augmente de 10%:
CI reste satisfaite: True
Profit Principal: $192.3581 \geq 94.1875$

Si \bar{w} diminue de 10%:
CI satisfaite: False
CP satisfaite: False

Conclusion sur l'Équilibre Bayésien Parfait:

Pas de PBE

=====
Comparaison des régimes First-Best vs Second-Best
=====

Résolution du first best (Information Symétrique)

Effort imposé: H
Probabilité de succès: $\pi_H = 0.70$

Contrat optimal (assurance complète):
 \bar{w} (succès) = w (échec) = 225.00

Résultats:
Profit espéré Principal: 225.00
Utilité Agent: 10.00 (= $u_r = 10.00$)

Comparaison:
Si effort H imposé: Profit = 225.00
Si effort B imposé: Profit = 106.00
Le principal préfère imposer l'effort H

Résolution du second best (Aléa Moral)

Contrat optimal:
 \bar{w} (succès) = 297.56
 w (échec) = 95.06
Écart $\bar{w} - w = 202.50$

Résultats:
Profit espéré Principal: 213.19
Utilité Agent (si e=H): 10.00
Utilité Agent (si e=B): 10.00

Vérification des contraintes:
CP (≥ 0): -0.0000 contrainte saturée
CI (≥ 0): 0.0000 contrainte saturée

Interprétation:

L'agent est incité à fournir l'effort élevé H
Mais il supporte du risque ($\bar{w} \neq w$)

Coût d'agence (perte due à l'asymétrie d'information):

Profit FB - Profit SB = 11.81
Perte relative: 5.25%

Écart de salaire:

FB: $\bar{w} - w = 0.00$ (assurance complète)
SB: $\bar{w} - w = 202.50$ (incitation)

=====
Analyse de sensibilité du coût d'agence
=====

1. Impact de l'aversion au risque (γ):

Résolution du first best (Information Symétrique)

Effort imposé: H
Probabilité de succès: $\pi_H = 0.70$

Contrat optimal (assurance complète):
 \bar{w} (succès) = w (échec) = 8323.47

Résultats:
Profit espéré Principal: -7873.47
Utilité Agent: 10.00 (= $u_r = 10.00$)

Comparaison:
Si effort H imposé: Profit = -7873.47
Si effort B imposé: Profit = -3706.13
Le principal préfère imposer l'effort B

Résolution du second best (Aléa Moral)

Contrat optimal:
 \bar{w} (succès) = 13262.65
 w (échec) = 1980.08
Écart $\bar{w} - w = 11282.57$

Résultats:
Profit espéré Principal: -9427.88
Utilité Agent (si $e=H$): 10.00
Utilité Agent (si $e=B$): 10.00

Vérification des contraintes:
CP (≥ 0): -0.0000 contrainte saturée
CI (≥ 0): -0.0000 contrainte saturée

Interprétation:

L'agent est incité à fournir l'effort élevé H
Mais il supporte du risque ($\bar{w} \neq w$)
 $\gamma = 0.30$: Coût d'agence = 1554.41

Résolution du first best (Information Symétrique)

Effort imposé: H

Probabilité de succès: $\pi_H = 0.70$

Contrat optimal (assurance complète):

\bar{w} (succès) = w (échec) = 871.42

Résultats:

Profit espéré Principal: -421.42

Utilité Agent: 10.00 (= $u_r = 10.00$)

Comparaison:

Si effort H imposé: Profit = -421.42

Si effort B imposé: Profit = -248.83

Le principal préfère imposer l'effort B

Résolution du second best (Aléa Moral)

Contrat optimal:

\bar{w} (succès) = 1235.87

w (échec) = 296.83

Écart $\bar{w} - w = 939.04$

Résultats:

Profit espéré Principal: -504.16

Utilité Agent (si $e=H$): 10.00

Utilité Agent (si $e=B$): 10.00

Vérification des contraintes:

CP (≥ 0): 0.0000 contrainte saturée

CI (≥ 0): 0.0000 contrainte saturée

Interprétation:

L'agent est incité à fournir l'effort élevé H

Mais il supporte du risque ($\bar{w} \neq w$)

$\gamma = 0.40$: Coût d'agence = 82.74

Résolution du first best (Information Symétrique)

Effort imposé: H

Probabilité de succès: $\pi_H = 0.70$

Contrat optimal (assurance complète):

\bar{w} (succès) = w (échec) = 225.00

Résultats:

Profit espéré Principal: 225.00

Utilité Agent: 10.00 (= $u_r = 10.00$)

Comparaison:

Si effort H imposé: Profit = 225.00

Si effort B imposé: Profit = 106.00

Le principal préfère imposer l'effort H

Résolution du second best (Aléa Moral)

Contrat optimal:

\bar{w} (succès) = 297.56
 w (échec) = 95.06
Écart $\bar{w} - w = 202.50$

Résultats:

Profit espéré Principal: 213.19
Utilité Agent (si $e=H$): 10.00
Utilité Agent (si $e=B$): 10.00

Vérification des contraintes:

CP (≥ 0): -0.0000 contrainte saturée
CI (≥ 0): 0.0000 contrainte saturée

Interprétation:

L'agent est incité à fournir l'effort élevé H
Mais il supporte du risque ($\bar{w} \neq w$)
 $\gamma = 0.50$: Coût d'agence = 11.81

Résolution du first best (Information Symétrique)

Effort imposé: H

Probabilité de succès: $\pi_H = 0.70$

Contrat optimal (assurance complète):

\bar{w} (succès) = w (échec) = 91.23

Résultats:

Profit espéré Principal: 358.77
Utilité Agent: 10.00 (= $u_r = 10.00$)

Comparaison:

Si effort H imposé: Profit = 358.77
Si effort B imposé: Profit = 187.10
Le principal préfère imposer l'effort H

Résolution du second best (Aléa Moral)

Contrat optimal:

\bar{w} (succès) = 115.16
 w (échec) = 44.50
Écart $\bar{w} - w = 70.67$

Résultats:

Profit espéré Principal: 356.04
Utilité Agent (si $e=H$): 10.00
Utilité Agent (si $e=B$): 10.00

Vérification des contraintes:

CP (≥ 0): -0.0000 contrainte saturée

CI (≥ 0): 0.0000 contrainte saturée

Interprétation:

L'agent est incité à fournir l'effort élevé H

Mais il supporte du risque ($\bar{w} \neq w$)

$\gamma = 0.60$: Coût d'agence = 2.73

Résolution du first best (Information Symétrique)

Effort imposé: H

Probabilité de succès: $\pi_H = 0.70$

Contrat optimal (assurance complète):

\bar{w} (succès) = w (échec) = 47.88

Résultats:

Profit espéré Principal: 402.12

Utilité Agent: 10.00 (= $u_r = 10.00$)

Comparaison:

Si effort H imposé: Profit = 402.12

Si effort B imposé: Profit = 215.19

Le principal préfère imposer l'effort H

Résolution du second best (Aléa Moral)

Contrat optimal:

\bar{w} (succès) = 58.46

w (échec) = 25.87

Écart $\bar{w} - w = 32.58$

Résultats:

Profit espéré Principal: 401.32

Utilité Agent (si $e=H$): 10.00

Utilité Agent (si $e=B$): 10.00

Vérification des contraintes:

CP (≥ 0): 0.0000 contrainte saturée

CI (≥ 0): 0.0000 contrainte saturée

Interprétation:

L'agent est incité à fournir l'effort élevé H

Mais il supporte du risque ($\bar{w} \neq w$)

$\gamma = 0.70$: Coût d'agence = 0.81

Résolution du first best (Information Symétrique)

Effort imposé: H

Probabilité de succès: $\pi_H = 0.70$

Contrat optimal (assurance complète):

\bar{w} (succès) = w (échec) = 29.52

Résultats:

Profit espéré Principal: 420.48
Utilité Agent: 10.00 (= u_r = 10.00)

Comparaison:

Si effort H imposé: Profit = 420.48
Si effort B imposé: Profit = 227.67
Le principal préfère imposer l'effort H

Résolution du second best (Aléa Moral)

Contrat optimal:

\bar{w} (succès) = 35.15
 w (échec) = 17.23
Écart $\bar{w} - w$ = 17.93

Résultats:

Profit espéré Principal: 420.22
Utilité Agent (si $e=H$): 10.00
Utilité Agent (si $e=B$): 10.00

Vérification des contraintes:

CP (≥ 0): 0.0000 contrainte saturée
CI (≥ 0): -0.0000 contrainte saturée

Interprétation:

L'agent est incité à fournir l'effort élevé H
Mais il supporte du risque ($\bar{w} \neq w$)
 $\gamma = 0.80$: Coût d'agence = 0.26

Résolution du first best (Information Symétrique)

Effort imposé: H
Probabilité de succès: $\pi_H = 0.70$

Contrat optimal (assurance complète):

\bar{w} (succès) = w (échec) = 20.27

Résultats:

Profit espéré Principal: 429.73
Utilité Agent: 10.00 (= u_r = 10.00)

Comparaison:

Si effort H imposé: Profit = 429.73
Si effort B imposé: Profit = 234.18
Le principal préfère imposer l'effort H

Résolution du second best (Aléa Moral)

Contrat optimal:

\bar{w} (succès) = 23.67
 w (échec) = 12.56
Écart $\bar{w} - w$ = 11.11

Résultats:

Profit espéré Principal: 429.66
Utilité Agent (si e=H): 10.00
Utilité Agent (si e=B): 10.00

Vérification des contraintes:

CP (≥ 0): 0.0000 contrainte saturée
CI (≥ 0): 0.0000 contrainte saturée

Interprétation:

L'agent est incité à fournir l'effort élevé H
Mais il supporte du risque ($\bar{w} \neq w$)
 $\gamma = 0.90$: Coût d'agence = 0.07

Résolution du first best (Information Symétrique)

Effort imposé: H

Probabilité de succès: $\pi_H = 0.70$

Contrat optimal (assurance complète):

\bar{w} (succès) = w (échec) = 15.00

Résultats:

Profit espéré Principal: 435.00
Utilité Agent: 10.00 (= $u_r = 10.00$)

Comparaison:

Si effort H imposé: Profit = 435.00
Si effort B imposé: Profit = 238.00
Le principal préfère imposer l'effort H

Résolution du second best (Aléa Moral)

Contrat optimal:

\bar{w} (succès) = 17.25
 w (échec) = 9.75
Écart $\bar{w} - w = 7.50$

Résultats:

Profit espéré Principal: 435.00
Utilité Agent (si e=H): 10.00
Utilité Agent (si e=B): 10.00

Vérification des contraintes:

CP (≥ 0): -0.0000 contrainte saturée
CI (≥ 0): -0.0000 contrainte saturée

Interprétation:

L'agent est incité à fournir l'effort élevé H
Mais il supporte du risque ($\bar{w} \neq w$)
 $\gamma = 1.00$: Coût d'agence = -0.00

2. Impact de π_B (effort faible):

Résolution du first best (Information Symétrique)

Effort imposé: H
Probabilité de succès: $\pi_H = 0.70$

Contrat optimal (assurance complète):
 \bar{w} (succès) = w (échec) = 225.00

Résultats:
Profit espéré Principal: 225.00
Utilité Agent: 10.00 (= $u_r = 10.00$)

Comparaison:
Si effort H imposé: Profit = 225.00
Si effort B imposé: Profit = 56.00
Le principal préfère imposer l'effort H

Résolution du second best (Aléa Moral)

Contrat optimal:
 \bar{w} (succès) = 282.24
 w (échec) = 116.64
Écart $\bar{w} - w = 165.60$

Résultats:
Profit espéré Principal: 217.44
Utilité Agent (si $e=H$): 10.00
Utilité Agent (si $e=B$): 10.00

Vérification des contraintes:
CP (≥ 0): -0.0000 contrainte saturée
CI (≥ 0): 0.0000 contrainte saturée

Interprétation:
L'agent est incité à fournir l'effort élevé H
Mais il supporte du risque ($\bar{w} \neq w$)
 $\pi_B = 0.20$: Coût d'agence = 7.56

Résolution du first best (Information Symétrique)

Effort imposé: H
Probabilité de succès: $\pi_H = 0.70$

Contrat optimal (assurance complète):
 \bar{w} (succès) = w (échec) = 225.00

Résultats:
Profit espéré Principal: 225.00
Utilité Agent: 10.00 (= $u_r = 10.00$)

Comparaison:
Si effort H imposé: Profit = 225.00
Si effort B imposé: Profit = 91.71
Le principal préfère imposer l'effort H

Résolution du second best (Aléa Moral)

Contrat optimal:

\bar{w} (succès) = 292.41
 w (échec) = 102.01
Écart $\bar{w} - w = 190.40$

Résultats:

Profit espéré Principal: 214.71
Utilité Agent (si $e=H$): 10.00
Utilité Agent (si $e=B$): 10.00

Vérification des contraintes:

CP (≥ 0): 0.0000 contrainte saturée
CI (≥ 0): -0.0000 contrainte saturée

Interprétation:

L'agent est incité à fournir l'effort élevé H
Mais il supporte du risque ($\bar{w} \neq w$)
 $\pi_B = 0.27$: Coût d'agence = 10.29

Résolution du first best (Information Symétrique)

Effort imposé: H

Probabilité de succès: $\pi_H = 0.70$

Contrat optimal (assurance complète):

\bar{w} (succès) = w (échec) = 225.00

Résultats:

Profit espéré Principal: 225.00
Utilité Agent: 10.00 (= $u_r = 10.00$)

Comparaison:

Si effort H imposé: Profit = 225.00
Si effort B imposé: Profit = 127.43
Le principal préfère imposer l'effort H

Résolution du second best (Aléa Moral)

Contrat optimal:

\bar{w} (succès) = 306.95
 w (échec) = 83.17
Écart $\bar{w} - w = 223.78$

Résultats:

Profit espéré Principal: 210.18
Utilité Agent (si $e=H$): 10.00
Utilité Agent (si $e=B$): 10.00

Vérification des contraintes:

CP (≥ 0): 0.0000 contrainte saturée
CI (≥ 0): -0.0000 contrainte saturée

Interprétation:

L'agent est incité à fournir l'effort élevé H

Mais il supporte du risque ($\bar{w} \neq w$)
 $\pi_B = 0.34$: Coût d'agence = 14.82

Résolution du first best (Information Symétrique)

Effort imposé: H
Probabilité de succès: $\pi_H = 0.70$

Contrat optimal (assurance complète):
 \bar{w} (succès) = w (échec) = 225.00

Résultats:
Profit espéré Principal: 225.00
Utilité Agent: 10.00 (= $u_r = 10.00$)

Comparaison:
Si effort H imposé: Profit = 225.00
Si effort B imposé: Profit = 163.14
Le principal préfère imposer l'effort H

Résolution du second best (Aléa Moral)

Contrat optimal:
 \bar{w} (succès) = 329.42
 w (échec) = 58.52
Écart $\bar{w} - w = 270.90$

Résultats:
Profit espéré Principal: 201.85
Utilité Agent (si $e=H$): 10.00
Utilité Agent (si $e=B$): 10.00

Vérification des contraintes:
CP (≥ 0): -0.0000 contrainte saturée
CI (≥ 0): 0.0000 contrainte saturée

Interprétation:
L'agent est incité à fournir l'effort élevé H
Mais il supporte du risque ($\bar{w} \neq w$)
 $\pi_B = 0.41$: Coût d'agence = 23.15

Résolution du first best (Information Symétrique)

Effort imposé: H
Probabilité de succès: $\pi_H = 0.70$

Contrat optimal (assurance complète):
 \bar{w} (succès) = w (échec) = 225.00

Résultats:
Profit espéré Principal: 225.00
Utilité Agent: 10.00 (= $u_r = 10.00$)

Comparaison:

Si effort H imposé: Profit = 225.00
Si effort B imposé: Profit = 198.86
Le principal préfère imposer l'effort H

Résolution du second best (Aléa Moral)

Contrat optimal:

\bar{w} (succès) = 368.64
 w (échec) = 27.04
Écart $\bar{w} - w$ = 341.60

Résultats:

Profit espéré Principal: 183.84
Utilité Agent (si $e=H$): 10.00
Utilité Agent (si $e=B$): 10.00

Vérification des contraintes:

CP (≥ 0): 0.0000 contrainte saturée
CI (≥ 0): 0.0000 contrainte saturée

Interprétation:

L'agent est incité à fournir l'effort élevé H
Mais il supporte du risque ($\bar{w} \neq w$)
 $\pi_B = 0.49$: Coût d'agence = 41.16

Résolution du first best (Information Symétrique)

Effort imposé: H

Probabilité de succès: $\pi_H = 0.70$

Contrat optimal (assurance complète):

\bar{w} (succès) = w (échec) = 225.00

Résultats:

Profit espéré Principal: 225.00
Utilité Agent: 10.00 (= u_r = 10.00)

Comparaison:

Si effort H imposé: Profit = 225.00
Si effort B imposé: Profit = 234.57
Le principal préfère imposer l'effort B

Résolution du second best (Aléa Moral)

Contrat optimal:

\bar{w} (succès) = 453.69
 w (échec) = 0.09
Écart $\bar{w} - w$ = 453.60

Résultats:

Profit espéré Principal: 132.39
Utilité Agent (si $e=H$): 10.00
Utilité Agent (si $e=B$): 10.00

Vérification des contraintes:

CP (≥ 0): 0.0000 contrainte saturée

CI (≥ 0): 0.0000 contrainte saturée

Interprétation:

L'agent est incité à fournir l'effort élevé H

Mais il supporte du risque ($\bar{w} \neq w$)

$\pi_B = 0.56$: Coût d'agence = 92.61

Résolution du first best (Information Symétrique)

Effort imposé: H

Probabilité de succès: $\pi_H = 0.70$

Contrat optimal (assurance complète):

\bar{w} (succès) = w (échec) = 225.00

Résultats:

Profit espéré Principal: 225.00

Utilité Agent: 10.00 (= $u_r = 10.00$)

Comparaison:

Si effort H imposé: Profit = 225.00

Si effort B imposé: Profit = 270.29

Le principal préfère imposer l'effort B

Résolution du second best (Aléa Moral)

Contrat optimal:

\bar{w} (succès) = 1764.00

w (échec) = 0.00

Écart $\bar{w} - w = 1764.00$

Résultats:

Profit espéré Principal: -784.80

Utilité Agent (si $e=H$): 24.40

Utilité Agent (si $e=B$): 24.40

Vérification des contraintes:

CP (≥ 0): 14.4000 ✓

CI (≥ 0): -0.0000 contrainte saturée

Interprétation:

L'agent est incité à fournir l'effort élevé H

Mais il supporte du risque ($\bar{w} \neq w$)

$\pi_B = 0.63$: Coût d'agence = 1009.80

