Sok-2013 Arbeidskrav 1 – Høst 2024

Publisert i Canvas 25.09.2024 kl 12:30 / Innlevering i Canvas innen 1.10.2024 kl 12:00

Oppgave 2 og 3 innebærer bruk av simulering. Kode for simulering ved bruk av R er lagt ved på bunnen av oppgavesettet.

Oppgave 1: Modellen inneholder en rekke med parametere. Definer disse parameterne under:

- Lambda λ
- Epsilon ϵ
- Rho ρ
- Delta δ
- Phi ϕ
- Gamma γ
- Beta β
- T

Oppgave 2: Ved hjelp av koden på bunnen av oppgaven, simuler modellen ved bruk av benchmark-verdiene:

Benchmark-verdier: $\delta = 0.4$ $\epsilon = 5$ $T = 1.7$ $\phi_1 = \phi_2 = 0.5$

$\beta = 0.8$ $\rho = 1 - \frac{1}{\epsilon}$ $\gamma = 0.4$ $L = 1$ $\alpha = \frac{\gamma * L}{\epsilon}$

a) Print w1/w2 raten for alle lambda (0,1). Forklar kort hva disse verdiene forteller oss, og lag en graf som viser dette forholdet.
b) Forklar hvorfor $\frac{w_1}{w_2} = 1$ når $\lambda = 0.5$

Oppgave 3: I deloppgavene må du variere parameterne. Hver underoppgave løses i et vakuum, så for eksempel etter oppgave a) er fullført så settes T tilbake til 1.7 før neste oppgave løses. Vis gjerne grafisk.

a) T endres fra 1.7 til 1.2, hvordan endrer w1/w2 raten seg? For hvilke verdier av λ vil bedriftene agglomerere i region 1?
b) Andel bønder i region 2 øker fra 0.5 til 0.7? Hva skjer med w1/w2 raten?
c) Dersom benchmark verdiene holdes, for hvilke lambda vil den langsiktige likevekten være fullstendig agglomerering i en region?
d) Dersom $\epsilon = 2$, for hvilke verdier av lambda vil den langsiktige likevekten være fullstendig agglomerering i en region?

Kode for modell simulering

#Funksjon for å kalkulere w1/w2 raten

```
calculate_w_ratio <- function(lambda) {
 lmbda1 <- lambda
 lmbda2 <- 1 - lmbda1

 calc_Y <- function(lmbda, W) {
  return(lmbda * gamma * L * W + phi1 * (1 - gamma) * L)
 }

 calc_I <- function(lmbdar, Wr, lmbdan, Wn) {
  const <- (beta / rho) * ((gamma * L) / (alpha * epsilon))^(1 / (1 - epsilon))
  return(const * (lmbdar * Wr^(1 - epsilon) + lmbdan * T^(1 - epsilon) * Wn^(1 - epsilon))^(1 / (1 -
epsilon)))
 }

 calc_W <- function(Yr, Ir, Yn, In) {
  const <- (rho * beta^(-rho)) * (delta / ((epsilon - 1) * alpha))^(1 / epsilon)
  return(const * (Yr * Ir^(epsilon - 1) + Yn * T^(1 - epsilon) * In^(epsilon - 1))^(1 / epsilon))
 }

 W1 <- 1
 W2 <- 1
 iterations <- 0

 while (TRUE) {
  Y1 <- calc_Y(lmbda1, W1)
  Y2 <- calc_Y(lmbda2, W2)
```

```
    I1 <- calc_I(lmbda1, W1, lmbda2, W2)
    I2 <- calc_I(lmbda2, W2, lmbda1, W1)
    new_W1 <- calc_W(Y1, I1, Y2, I2)
    new_W2 <- calc_W(Y2, I2, Y1, I1)
    w1 <- new_W1 * I1^(-delta)
    w2 <- new_W2 * I2^(-delta)

    criterion1 <- abs((new_W1 - W1) / W1)
    criterion2 <- abs((new_W2 - W2) / W2)

    if (criterion1 < LIMIT && criterion2 < LIMIT) {
      break
    }

    W1 <- new_W1
    W2 <- new_W2
    iterations <- iterations + 1
  }

  return(w1 / w2)
}

#Funksjon for å simulere modellen
simulate_model <- function(lambda_values) {
  w_ratios <- numeric(length(lambda_values))

  for (i in seq_along(lambda_values)) {
    lambda <- lambda_values[i]
    w_ratios[i] <- calculate_w_ratio(lambda)
  }
```

```
  return(w_ratios)
}

#Simulering for benchmark verdier
lambda_values <- seq(0, 1, by = 0.1)
benchmark_ratios <- simulate_model(lambda_values)
```