Модель:

Модель 1. Модель у вигляді диференційних рівнянь.
Припустимо, що ті, що видужали, набувають стійкого імунітету.

Тоді введені величини пов'язані наступними рівняннями:

$S(t)+I(t)+R(t)+D(t)=N$,

$S(t+dt)=S(t)-S(t)p(I(t))dt-u(t)Sdt$,

$I(t+dt)=I(t)+S(t)p(I(t))dt-I(t)\alpha dt-I(t)\beta dt$,

$R(t+dt)=R(t)+I(t)\alpha dt+S(t)u(t)dt$,

$D(t+dt)=D(t)+I(t)\beta dt$

Звідси

$$\frac{dS}{dt}=-Sp(I)-Su(t),$$

$$\frac{dI}{dt}=Sp(I)-\alpha I-\beta I,$$

$$\frac{dR}{dt}=\alpha I+u(t)S \quad (\text{наслідок}),$$

$$\frac{dD}{dt}=\beta I \quad (\text{наслідок}), \qquad t\in[0,T].$$

В матричній формі

$$\frac{d}{dt}\begin{pmatrix} S \\ I \end{pmatrix}=\begin{pmatrix} -p(I) & 0 \\ p(I) & -(\alpha+\beta) \end{pmatrix}\begin{pmatrix} S \\ I \end{pmatrix}+\begin{pmatrix} -S \\ 0 \end{pmatrix}u$$ (нелінійна система),

$$p(I)=-\frac{rI\log(1-c)}{N}.$$

Початкові умови для цієї системи: $I(0)>0$, $S(0)=N-I(0)$.

Позначимо

$X=\begin{pmatrix} X_1=S \\ X_2=I \end{pmatrix}$ - вектор стану системи (або вектор фазових змінних),

$u(t)$ - управління (керована змінна), $u(t)\in U=\{u:0\le u\le u_{\max}<1\}$,

$U=\{u:0\le u\le u_{\max}<1\}$ - множина допустимих керувань.

Оптимальне керування

Позначимо наступні параметри:

a - вартість вакцинації одної людини,

b - вартість лікування однієї людини.

Функціонали, що підлягають оптимізації (мінімізації):

$J(u)=a\int_0^T S(t)u(t)dt+b\int_0^T \alpha I(t)dt\to\min_{u(t)\in U}$ (вартість імунізації та лікування),

$I(T,u)\to\min_{u(t)\in U}$ (число хворих на момент часу T);

$R(T,u)=\int_0^T \beta I(t)dt$ (загальне число померлих за час T.)

Хід виконання

Параметри моделі:

- N (загальне населення): невелике місто на 100 000 осіб.
- r (швидкість контакту): 5 контактів на день.
- c (ймовірність передачі за контакт): 0,1.
- α (швидкість одужання): інфекційний період приблизно 10 днів, $\alpha = 1/10$.
- β (швидкість смертності): відображаючи вищу смертність, 0,01.
- u_max (максимальний відсоток щеплених/ізольованих осіб на день): 0, вважаючи що спочатку не було щеплення/ізоляції

Деякі інші значення:

- I(0): 100 осіб на початку інфікованих.
- S(0): N - I(0), решта населення є вразливою.
- R(0) та D(0): 0, оскільки на початку ніхто не одужав або не помер.

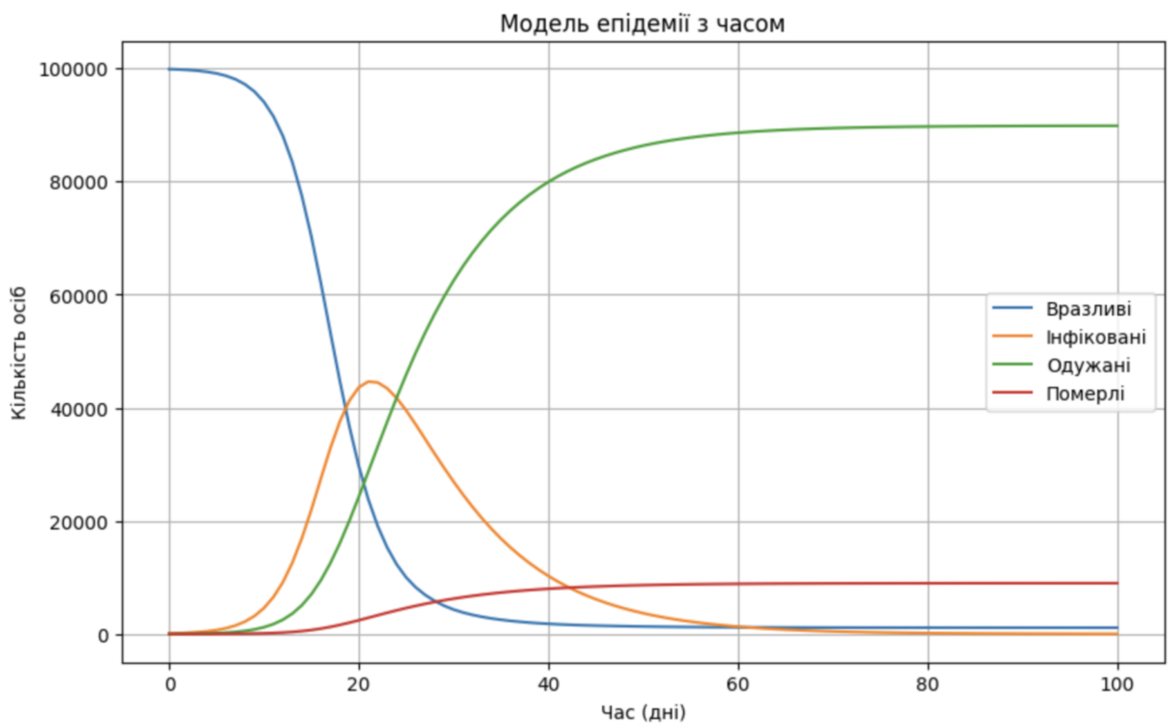
Ми будемо моделювати протягом T = 100 днів, що дозволить побачити прогресію та потенційну стабілізацію епідемії.

Спочатку, підготуємо нашу модель та диференціальні рівняння, розв'яжемо їх та візуалізуємо результати:

```
from scipy.integrate import solve_ivp
import numpy as np
import matplotlib.pyplot as plt

# Параметри моделі
N = 100000  # загальне населення
```

```python
r = 5       # швидкість контакту на особу на день
c = 0.1     # ймовірність зараження за контакт
alpha = 1/10  # швидкість одужання на день
beta = 0.01   # швидкість смертності на день
u_max = 0     # немає щеплення/ізоляції

# Початкові умови
I0 = 100
S0 = N - I0
R0 = 0
D0 = 0
initial_conditions = [S0, I0, R0, D0]

# Інтервал часу
T = 100
t = np.linspace(0, T, T+1)  # 101 точка від 0 до 100

# Диференціальні рівняння
def model(t, y, N, r, c, alpha, beta, u_max):
    S, I, R, D = y
    dSdt = -r*c*I*S/N - u_max*S
    dIdt = r*c*I*S/N - alpha*I - beta*I
    dRdt = alpha*I
    dDdt = beta*I
    return [dSdt, dIdt, dRdt, dDdt]

# Розв'язання моделі
solution = solve_ivp(model, [0, T], initial_conditions, args=(N, r, c,
alpha, beta, u_max), t_eval=t)

# Побудова графіка
plt.figure(figsize=(10, 6))
plt.plot(solution.t, solution.y[0], label='Вразливі')
plt.plot(solution.t, solution.y[1], label='Інфіковані')
```

```
plt.plot(solution.t, solution.y[2], label='Одужані')
plt.plot(solution.t, solution.y[3], label='Померлі')
plt.title('Модель епідемії з часом')
plt.xlabel('Час (дні)')
plt.ylabel('Кількість осіб')
plt.legend()
plt.grid(True)
plt.show()
```



- **Вразливі**: зменшується з часом, оскільки вірус поширюється серед населення. Це зменшення спочатку є поступовим, але прискорюється зі збільшенням кількості інфікованих осіб, максимізуючи швидкість передачі.
- **Інфіковані**: спочатку різко зростає, що означає швидке поширення інфекції. Після досягнення піку вона починає зменшуватися, оскільки

більше осіб одужує або помирає, а пул вразливих осіб зменшується, зменшуючи здатність віруса до поширення.

- **Одужані**: зростає з часом, відстаючи від кривої інфікування. Це збільшення відображає загальну кількість осіб, які набули імунітет до вірусу, чи то через одужання від інфекції, чи в інших сценаріях - через вакцинацію.
- **Померлі**: зростає з часом, слідуючи тенденції інфікування, але з меншою швидкістю.

```
# Максимальна кількість інфікованих та час, коли це відбувається
I_max = np.max(solution.y[1])
t_max = solution.t[np.argmax(solution.y[1])]

I_max, t_max
```

```
(44630.99235498599, 21.0)
```

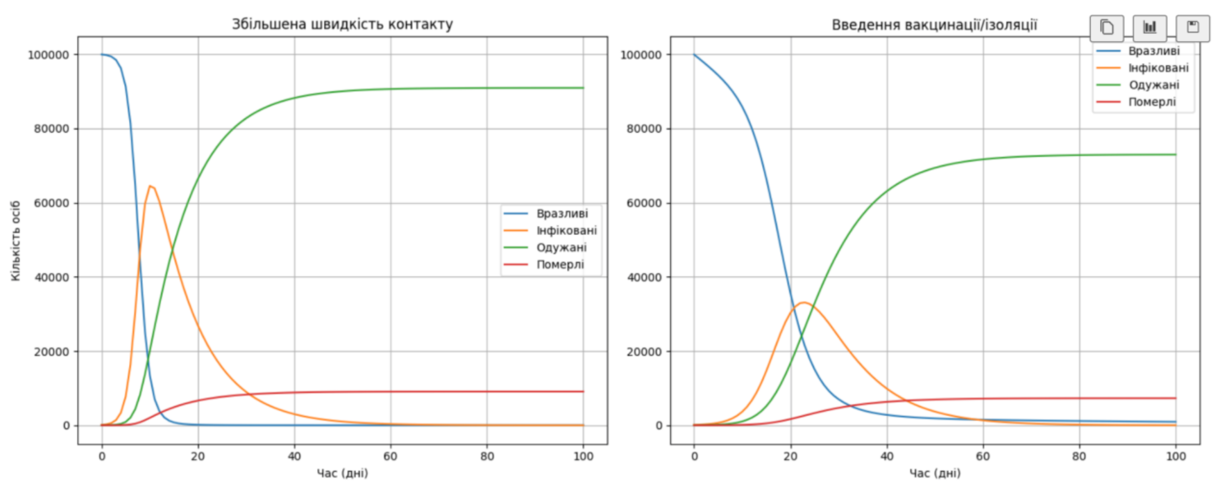
Пік епідемії з максимальною кількістю інфікованих осіб припадає на 44 631 особу на 21-й день.

Тепер змоделюємо ефект щеплення, встановивши u(t)=u_max на позитивне значення, що означає певний відсоток населення, яке щеплюється (або ізолюється) щодня.

Також збільшимо швидкість контакту (r) з 5 до 10 контактів на особу на день, залишаючи інші параметри незмінними, і візуалізуємо, як ця зміна впливає на прогресію епідемії. Потім змоделюємо введення щеплення з u_max = 0.01, що означає щоденне щеплення 1% вразливого населення.

```
# Сценарій 1: збільшення швидкості контакту
r_increased = 10  # Збільшена швидкість контакту

# Розв'язання моделі зі збільшеною швидкістю контакту
solution_increased_r = solve_ivp(model, [0, T], initial_conditions,
args=(N, r_increased, c, alpha, beta, u_max), t_eval=t)

# Сценарій 2: введення заходів (вакцинація/ізоляція)
u_max_vaccination = 0.01  # 1% вразливого населення щеплюється щодня

# Розв'язання моделі з вакцинацією
solution_vaccination = solve_ivp(model, [0, T], initial_conditions,
args=(N, r, c, alpha, beta, u_max_vaccination), t_eval=t)

# Побудова обох сценаріїв
plt.figure(figsize=(15, 6))

# Збільшена швидкість контакту
plt.subplot(1, 2, 1)
plt.plot(solution_increased_r.t, solution_increased_r.y[0],
label='Вразливі')
plt.plot(solution_increased_r.t, solution_increased_r.y[1],
label='Інфіковані')
plt.plot(solution_increased_r.t, solution_increased_r.y[2],
label='Одужані')
plt.plot(solution_increased_r.t, solution_increased_r.y[3],
label='Померлі')
plt.title('Збільшена швидкість контакту')
plt.xlabel('Час (дні)')
plt.ylabel('Кількість осіб')
plt.legend()
plt.grid(True)

# Вакцинація
```

```
plt.subplot(1, 2, 2)
plt.plot(solution_vaccination.t, solution_vaccination.y[0],
label='Вразливі')
plt.plot(solution_vaccination.t, solution_vaccination.y[1],
label='Інфіковані')
plt.plot(solution_vaccination.t, solution_vaccination.y[2],
label='Одужані')
plt.plot(solution_vaccination.t, solution_vaccination.y[3],
label='Померлі')
plt.title('Введення вакцинації/ізоляції')
plt.xlabel('Час (дні)')
plt.legend()
plt.grid(True)

plt.tight_layout()
plt.show()
```



Збільшена швидкість контакту:

- **Вразливі**: зменшення кількості вразливих осіб відбувається швидше через збільшення контактів, що призводить до швидшого поширення вірусу.

- **Інфіковані**: пік інфекцій вищий і відбувається раніше порівнюючи з базовим сценарієм. Тобто збільшення соціальної взаємодії без заходів пом'якшення значно погіршує ситуацію.
- Одужані і померлі: піднімаються стрімкіше, що відображає більшу кількість інфекцій.

Введення вакцинації/ізоляції

- **Вразливі**: поступове зменшення.
- **Інфіковані**: пік помітно нижчий, що демонструє ефективність вакцинації/ізоляції.
- **Одужані**: зростає стабільніше через поєднання природного одужання та ефекту зменшення вразливих осіб за допомогою вакцинації/ізоляції.
- **Померлі**: кількість смертей зростає повільніше, через зменшення поширення інфекцій.

```
# Оцінка кінцевих результатів на основі останніх значень

S_infinity = solution.y[0, -1]
I_infinity = solution.y[1, -1]
R_infinity = solution.y[2, -1]
D_infinity = solution.y[3, -1]

S_infinity, I_infinity, R_infinity, D_infinity
```

```
(1113.6710969928185, 20.171752425956157, 89878.32468234655, 8987.832468234654)
```

Почнемо з кроку 8, оцінивши кінцеві результати на основі поточної симуляції. Потім ми продовжимо подальше дослідження впливу змінних параметрів та впровадження різних заходів контролю.

```
# Оцінка кінцевих результатів на основі останніх значень з нашої симуляції
S_infinity = solution.y[0, -1]
I_infinity = solution.y[1, -1]
R_infinity = solution.y[2, -1]
D_infinity = solution.y[3, -1]

S_infinity, I_infinity, R_infinity, D_infinity
```

```
(1113.6710969928185, 20.171752425956157, 89878.32468234655, 8987.832468234654)
```

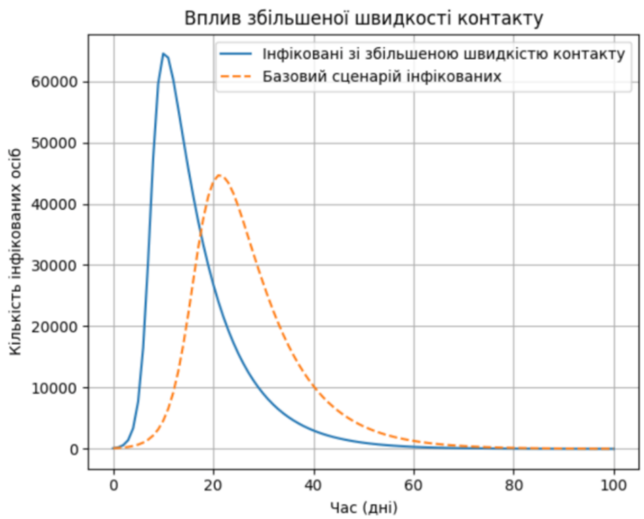
Оцінені кінцеві результати епідемії, на основі нашої симуляції з вибраними параметрами, такі:

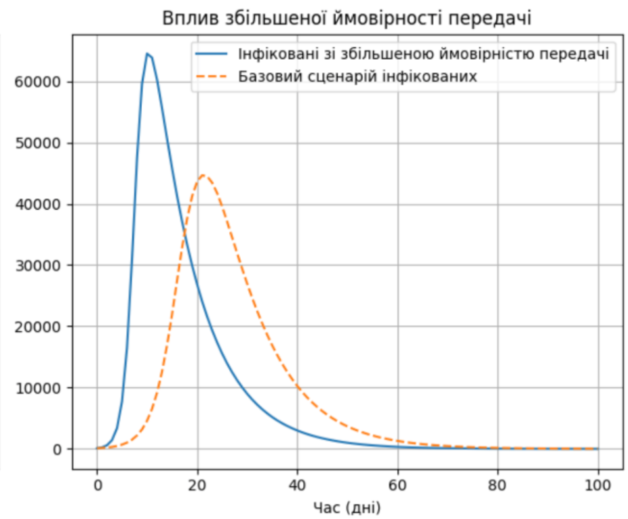
Вразливі: 1,114 осіб залишаються вразливими в кінці епідемії. Тобто значна частина населення була або інфікована, або отримала вакцинацію/ізоляцію.
Інфіковані: 20 осіб ще заражені. Тобтот епідемія наближається до кінця.
Одужані: 89,878 осіб одужали – це ті, хто отримав імунітет через інфікування та щеплення.
Померлі: 8,988 смертей.

Порівняємо, як зміни у швидкості контакту (r = 10, замість 5) та ймовірності передачі за контакт (c = 0.2, замість 0.1) впливають на динаміку епідемії.

```
# Параметри для сценаріїв зі збільшеною швидкістю контакту та ймовірністю передачі
r_higher = 10   # Збільшена швидкість контакту
c_higher = 0.2  # Збільшена ймовірність передачі

# збільшеа швидкість контакту
solution_higher_r = solve_ivp(model, [0, T], initial_conditions, args=(N, r_higher, c, alpha, beta, u_max), t_eval=t)
```

```
# збільшена ймовірність передачі
solution_higher_c = solve_ivp(model, [0, T], initial_conditions,
args=(N, r, c_higher, alpha, beta, u_max), t_eval=t)

# Побудова сценаріїв
plt.figure(figsize=(12, 5))

# Збільшена швидкість контакту
plt.subplot(1, 2, 1)
plt.plot(solution_higher_r.t, solution_higher_r.y[1], label='Інфіковані
зі збільшеною швидкістю контакту')
plt.plot(solution.t, solution.y[1], '--', label='Базовий сценарій
інфікованих')
plt.title('Вплив збільшеної швидкості контакту')
plt.xlabel('Час (дні)')
plt.ylabel('Кількість інфікованих осіб')
plt.legend()
plt.grid(True)

# Збільшена ймовірність передачі
plt.subplot(1, 2, 2)
plt.plot(solution_higher_c.t, solution_higher_c.y[1], label='Інфіковані
зі збільшеною ймовірністю передачі')
plt.plot(solution.t, solution.y[1], '--', label='Базовий сценарій
інфікованих')
plt.title('Вплив збільшеної ймовірності передачі')
plt.xlabel('Час (дні)')
plt.legend()
plt.grid(True)

plt.tight_layout()
plt.show()
```





Пік інфекцій виявляється набагато вищим і відбувається раніше порівняно з базовим сценарієм. Тобто бачимо роль дистанціювання та обмеження контактів у сповільненні поширення епідемії.

Збільшення ймовірності передачі призводить до вищого піку інфекцій. Епідемія поширюється швидше та більш агресивно, що знову підкреслює важливість зменшення передачі за допомогою таких заходів, як гігієна, носіння масок.

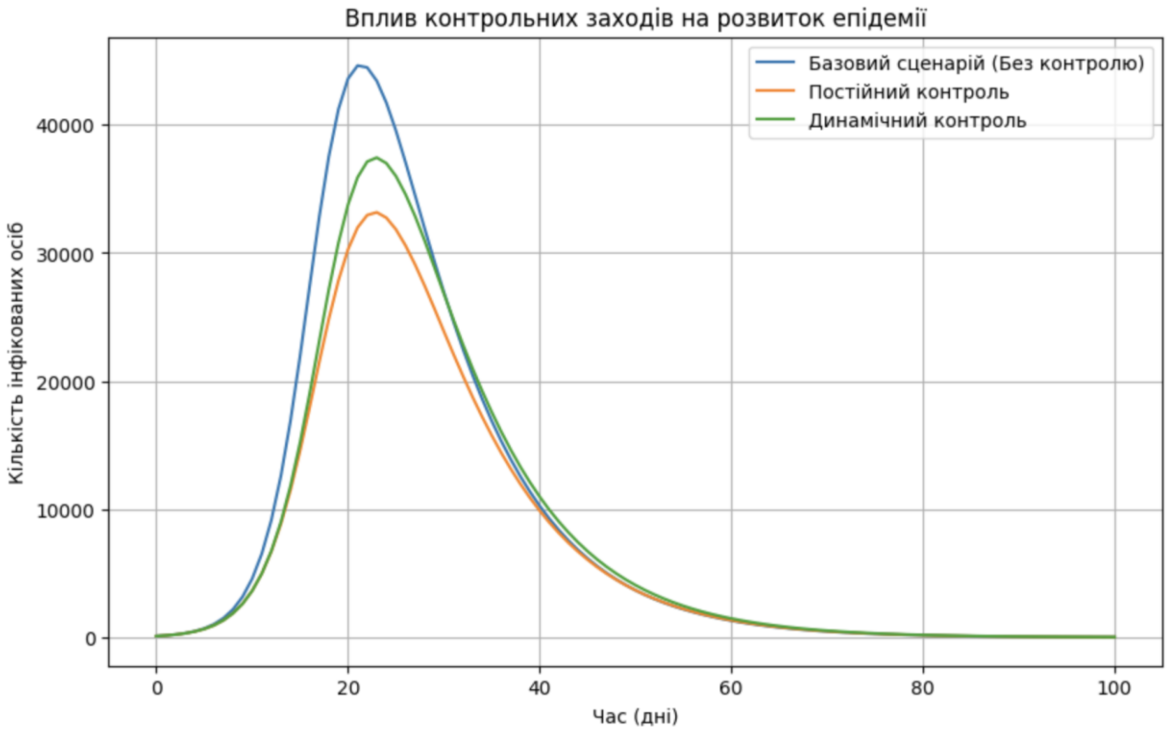
Розглянемо зміни у розвитку епідемії для різних контрольних заходів, включаючи динамічний контроль, такий як u(t), який представляє частку вакцинованих або ізольованих осіб з часом. Ми будемо симулювати вплив різних значень u(t) та вводимо сценарій, де u(t) динамічно налаштовується на основі стану епідемії, наприклад, збільшуючи швидкість ізоляції або вакцинації у відповідь на зростання числа інфікованих.

Зосередимось на впливі цих стратегій контролю на зменшення піку та розподіл інфекцій з часом.

```
def model_with_control(t, y, N, r, c, alpha, beta, u_static,
dynamic_control=False, I_threshold=0.05):
```

```
    S, I, R, D = y
    if dynamic_control and I/N > I_threshold:
        u = min(u_max, u_static * (I/N) / I_threshold)  # Динамічно
збільшуємо u в залежності від інфекції
    else:
        u = u_static
    dSdt = -r*c*I*S/N - u*S
    dIdt = r*c*I*S/N - alpha*I - beta*I
    dRdt = alpha*I + u*S
    dDdt = beta*I
    return [dSdt, dIdt, dRdt, dDdt]

u_static = 0.01  # Постійний рівень вакцинації/ізоляції

solution_base = solve_ivp(model_with_control, [0, T],
initial_conditions, args=(N, r, c, alpha, beta, 0, False), t_eval=t)
solution_static_control = solve_ivp(model_with_control, [0, T],
initial_conditions, args=(N, r, c, alpha, beta, u_static, False),
t_eval=t)
solution_dynamic_control = solve_ivp(model_with_control, [0, T],
initial_conditions, args=(N, r, c, alpha, beta, u_static, True),
t_eval=t)

# Візуалізація порівняння
plt.figure(figsize=(10, 6))
plt.plot(solution_base.t, solution_base.y[1], label='Базовий сценарій
(Без контролю)')
plt.plot(solution_static_control.t, solution_static_control.y[1],
label='Постійний контроль')
plt.plot(solution_dynamic_control.t, solution_dynamic_control.y[1],
label='Динамічний контроль')
plt.title('Вплив контрольних заходів на розвиток епідемії')
plt.xlabel('Час (дні)')
plt.ylabel('Кількість інфікованих осіб')
```

```
plt.legend()
plt.grid(True)
plt.show()
```



Явно видно, що введення контрольних заходів, як постійних, так і динамічних, знижує пік інфекцій та розтягує епідемію в часі, що може бути критично важливим для недопущення перевантаження системи охорони здоров'я. Найбільш помітний вплив має динамічний контроль, що показує важливість гнучкості під час епідемії.