# Математичні формули алгоритмів оптимізації QS рейтингу

## 1. Математична постановка задачі

### 1.1 Показники QS рейтингу

Система працює з 9 основними показниками QS World University Rankings:

* **AR** (Academic Reputation) - Академічна репутація
* **ER** (Employer Reputation) - Репутація серед роботодавців
* **FSR** (Faculty Student Ratio) - Співвідношення викладачів до студентів
* **CPF** (Citations per Faculty) - Цитування на викладача
* **IFR** (International Faculty Ratio) - Частка іноземних викладачів
* **ISR** (International Student Ratio) - Частка іноземних студентів
* **IRN** (International Research Network) - Міжнародна дослідницька мережа
* **EO** (Employment Outcomes) - Результати працевлаштування
* **SUS** (Sustainability) - Сталість

### 1.2 Базові параметри

**Поточні значення показників (2025):**

QS\_INPUT = {k: x₀ₖ | k ∈ {AR, ER, FSR, CPF, IFR, ISR, IRN, EO, SUS}}

**Ваги показників:**

QS\_WEIGHTS = {k: wₖ | k ∈ {AR, ER, FSR, CPF, IFR, ISR, IRN, EO, SUS}}

**Максимальні значення:**

QS\_MAX = {k: x\_maxₖ | k ∈ {AR, ER, FSR, CPF, IFR, ISR, IRN, EO, SUS}}

**Можливі прирости:**

QS\_DELTA = {k: Δₖ | k ∈ {AR, ER, FSR, CPF, IFR, ISR, IRN, EO, SUS}}

**Вартість покращення (RU за одиницю):**

QS\_COST = {k: cₖ | k ∈ {AR, ER, FSR, CPF, IFR, ISR, IRN, EO, SUS}}

**Бюджет ресурсів:**

MAX\_RU = B = 100

### 1.3 Основна математична задача

**Мета:** Максимізувати QS Score університету в межах бюджету ресурсів

**Цільова функція:**

maximize: QS\_Score(x) = Σ(k∈K) wₖ × xₖ

**Обмеження:**

subject to:  
 x₀ₖ ≤ xₖ ≤ min(x₀ₖ + Δₖ, x\_maxₖ), ∀k ∈ K  
 Σ(k∈K) cₖ × max(0, xₖ - x₀ₖ) ≤ B  
 xₖ = x₀ₖ, якщо Δₖ = 0 або cₖ = ∞

де: - x = {xₖ} - вектор значень показників - wₖ - вага показника k - K = {AR, ER, FSR, CPF, IFR, ISR, IRN, EO, SUS}

### 1.4 Пояснення обмежень

**1. Обмеження на значення показників:**

x₀ₖ ≤ xₖ ≤ min(x₀ₖ + Δₖ, x\_maxₖ), ∀k ∈ K

* x₀ₖ - поточне значення показника (2025 рік)
* xₖ - нове значення показника (2026 рік)
* Δₖ - максимальний можливий приріст
* x\_maxₖ - абсолютний максимум для показника

**2. Бюджетне обмеження:**

Σ(k∈K) cₖ × max(0, xₖ - x₀ₖ) ≤ B

* cₖ - вартість покращення показника k на 1 одиницю
* xₖ - x₀ₖ - приріст показника
* max(0, xₖ - x₀ₖ) - тільки позитивні прирости (не можемо “погіршити”)
* B = 100 - наш бюджет в RU

**3. Обмеження на заморожені показники:**

xₖ = x₀ₖ, якщо Δₖ = 0 або cₖ = ∞

* Якщо Δₖ = 0 - показник не можна покращити
* Якщо cₖ = ∞ - показник занадто дорогий
* В обох випадках: xₖ = x₀ₖ (залишається незмінним)

## 2. Модифікації основної задачі

### 2.1 Оптимізація всіх показників

**Модифікація:** Всі показники можна змінювати

Δ'\_k = Δₖ, ∀k ∈ K

### 2.2 Оптимізація обраних показників

**Модифікація:** Заморожування необраних показників

Δ'\_k = {  
 Δₖ, якщо k ∈ selected\_indicators  
 0, інакше  
}

### 2.3 Топ-N стратегії

**Модифікація:** Заморожування всіх показників крім поточної комбінації

Δ'\_k = {  
 Δₖ, якщо k ∈ combo  
 0, інакше  
}

**Множина придатних показників:**

E = {k ∈ K | Δₖ > 0 ∧ cₖ < ∞}

**Всі можливі комбінації з N показників:**

C\_N = {C ⊆ E | |C| = N}

## 3. Способи розв’язання

**Примітка:** Обидва алгоритми розв’язують одну і ту ж математичну задачу, але використовують різні способи формалізації.

### 3.1 Генетичний алгоритм (GA)

**Формалізація:** Використовує неперервні змінні xₖ (безпосередньо значення показників)

#### 3.1.1 Фітнес-функція

f(x) = {  
 QS\_Score(x), якщо всі обмеження виконані  
 -10000, якщо порушено обмеження на заморожені показники  
 -1000 × (RU\_used - B), якщо перевищено бюджет  
}

#### 3.1.2 Простір генів

gene\_space[k] = {  
 [x₀ₖ], якщо Δₖ = 0 або cₖ = ∞  
 {low: x₀ₖ, high: min(x₀ₖ + Δₖ, x\_maxₖ), step: 0.1}, інакше  
}

#### 3.1.3 Параметри алгоритму

* num\_generations - кількість поколінь (за замовчуванням: 400)
* sol\_per\_pop - розмір популяції (за замовчуванням: 60)
* num\_parents\_mating - кількість батьків для схрещування (за замовчуванням: 24)
* mutation\_percent\_genes - відсоток мутацій (за замовчуванням: 20%)
* stop\_criteria - критерій зупинки (за замовчуванням: “saturate\_15”)

#### 3.1.4 Автоматичний пошук параметрів (Optuna)

**Цільова функція для Optuna:**

objective(trial) = (1/n\_trials\_per\_eval) × Σ(i=1 to n\_trials\_per\_eval) QS\_Score(GA\_result(trial.params, seed\_i))

**Параметри для оптимізації:** - num\_generations ∈ [100, 500] - sol\_per\_pop ∈ [20, 100] - num\_parents\_mating ∈ [5, sol\_per\_pop/2] - mutation\_percent\_genes ∈ [5, 40] - random\_seed ∈ [1, 1000]

### 3.2 Лінійне програмування (LP)

**Формалізація:** Використовує дискретні змінні kₖ (кількість кроків 0.1)

#### 3.2.1 Змінні рішення

**Дискретні змінні:**

kₖ ∈ ℤ₊ - кількість кроків 0.1 для показника k

**Нові значення показників:**

xₖ = x₀ₖ + 0.1 × kₖ

#### 3.2.2 Аналітичне формулювання задачі LP

max Σ(k∈K) wₖ × (x₀ₖ + 0.1 × kₖ)  
  
subject to:  
 0.1 × kₖ ≤ min(Δₖ, x\_maxₖ - x₀ₖ), ∀k ∈ K  
 Σ(k∈K) cₖ × 0.1 × kₖ ≤ B  
 kₖ = 0, якщо k ∉ selected\_indicators або cₖ = ∞  
 kₖ ∈ ℤ₊, ∀k ∈ K

#### 3.2.3 Пояснення обмежень LP

**1. Обмеження на максимальний приріст:**

0.1 × kₖ ≤ min(Δₖ, x\_maxₖ - x₀ₖ), ∀k ∈ K

* kₖ - кількість кроків 0.1 для показника k
* 0.1 × kₖ - загальний приріст показника
* Не можемо перевищити максимальний приріст або абсолютний максимум

**Приклад для AR:**

0.1 × k\_AR ≤ min(1.0, 15 - 6.5) = min(1.0, 8.5) = 1.0

Тобто k\_AR ≤ 10 (максимум 10 кроків по 0.1)

**2. Бюджетне обмеження:**

Σ(k∈K) cₖ × 0.1 × kₖ ≤ B

* cₖ × 0.1 × kₖ - витрати на покращення показника k
* Сума всіх витрат не може перевищити бюджет

**Приклад:**

50 × 0.1 × k\_AR + 45 × 0.1 × k\_ER + ... ≤ 100

**3. Обмеження на заморожені показники:**

kₖ = 0, якщо k ∉ selected\_indicators або cₖ = ∞

* Якщо показник не обраний або занадто дорогий
* То kₖ = 0 (не робимо жодних кроків)

**4. Цілочисельність:**

kₖ ∈ ℤ₊, ∀k ∈ K

* kₖ - ціле невід’ємне число
* Не можемо зробити часткові кроки (наприклад, 2.5 кроки)

### 3.3 Порівняння формалізацій

| Аспект | Генетичний алгоритм | Лінійне програмування |
| --- | --- | --- |
| **Змінні** | xₖ (неперервні) | kₖ (дискретні) |
| **Крок** | Будь-який (0.1) | Фіксований (0.1) |
| **Обмеження** | Штрафи в фітнес-функції | Математичні обмеження |
| **Пошук** | Еволюційний | Симплекс-метод |
| **Результат** | Наближений оптимум | Точний оптимум |

## 4. Допоміжні функції

### 4.1 Розрахунок загальних витрат RU

compute\_total\_ru(QS\_INPUT, QS\_COST, solution) = Σ(k∈K) cₖ × max(0, solution[k] - x₀ₖ)

### 4.2 Розрахунок QS Score

compute\_qs\_score(solution, QS\_WEIGHTS, keys) = Σ(i=0 to |keys|-1) solution[i] × w\_{keys[i]}

### 4.3 Ефективність стратегії

efficiency = (QS\_Score\_new - QS\_Score\_current) / RU\_used

### 4.4 Відсоток покращення

improvement\_percent = ((QS\_Score\_new - QS\_Score\_current) / QS\_Score\_current) × 100%

## 5. Метрики якості

### 5.1 Основні метрики

* QS\_Score - загальний рейтинг
* RU\_used - використані ресурси
* efficiency - ефективність (приріст/витрати)
* improvement\_percent - відсоток покращення