Пример решения задачи оптимизации портфеля через целочисленное программирование в Julia

Целочисленное программирование (ЦП) — задача оптимизации с дискретными переменными. Основная форма:

$$maxc^T x$$

при условии

$$Ax \leq b$$

$$x \in \mathbb{Z}^n$$

В инвестиционной задаче:

Цель: максимизация доходности

Ограничения: квоты распределения в фиксированных единицах

Постановка задачи

Инвестор хочет распределить $10,000 между 3 активами:

Актив	Ожидаемая доходность	Риск (σ)	Минимальная доля	Максимальная доля
A	8%	10%	10%	50%
B	12%	15%	10%	50%
C	6%	5%	10%	50%

Ковариационная матрица: - Cov(A,B) = 0.5% - Cov(A,C) = -0.2% - Cov(B,C) = 0.3%

Инвестировать можно только тысячами (1000).

[ ]:
```
ASSETS = [:A, :B, :C]
RETURNS = [0.08, 0.12, 0.06]  # Процентная доходность
TOTAL_AMOUNT = 11000  # Общий бюджет в долларах
UNIT_VALUE = 1000     # Шаг инвестиций
MIN_PERCENT = 10      # Минимальная доля в %
MAX_PERCENT = 50      # Максимальная доля в %

MIN_UNITS = (MIN_PERCENT/100) * TOTAL_AMOUNT / UNIT_VALUE
MAX_UNITS = (MAX_PERCENT/100) * TOTAL_AMOUNT / UNIT_VALUE
TOTAL_UNITS = Int(TOTAL_AMOUNT / UNIT_VALUE)
```

[ ]:
```
11
```

Инициализация окружения

Подключим необходимые пакеты:

[ ]:
```
# import Pkg
# Pkg.add("JuMP")
# Pkg.add("Cbc")
```

[ ]:
```
using JuMP, Cbc
```

Инициализация модели

Создаем модель с целочисленным решателем Cbc

[ ]:
```
model = Model(Cbc.Optimizer)
set_optimizer_attribute(model, "logLevel", 0)
```

Переменные решения

Целочисленные переменные для каждого актива с явными границами

[ ]:
```
@variable(model, MIN_UNITS <= x_A <= MAX_UNITS, Int, base_name = "A")
@variable(model, MIN_UNITS <= x_B <= MAX_UNITS, Int, base_name = "B")
@variable(model, MIN_UNITS <= x_C <= MAX_UNITS, Int, base_name = "C")
```

[ ]:
C

Базовые ограничения

Полное инвестирование капитала

[ ]:
```
@constraint(model, x_A + x_B + x_C == TOTAL_UNITS)
```

[ ]:
A + B + C = 11

Целевая функция

Максимизация ожидаемой доходности портфеля

[ ]:
```
@objective(model, Max,
    RETURNS[1] * x_A +
    RETURNS[2] * x_B +
    RETURNS[3] * x_C
)
```

[ ]:
$0.08A + 0.12B + 0.06C$

Решение задачи

Запуск оптимизации и проверка корректности решения

[ ]:
```
optimize!(model)

if termination_status(model) != MOI.OPTIMAL
    error("Решение не найдено: ", termination_status(model))
end
```

Результаты оптимизации

Форматированный вывод с преобразованием единиц

[ ]:
```
function format_currency(amount)
    return "\$" * replace(string(amount), "." => ",")
end

println("\nОптимальное распределение:")
for (var, asset) in [(x_A, "A"), (x_B, "B"), (x_C, "C")]
    units = round(Int, value(var))
    amount = units * UNIT_VALUE
    exact_percent = units / TOTAL_UNITS * 100
    println(
        "$asset: ", lpad(units, 2), " ед. → ",
        lpad(format_currency(amount), 7),
        " (", round(exact_percent, digits=1), "%)"
    )
end

total_return = objective_value(model)
println("\nИтоговая доходность: ", total_return * UNIT_VALUE / 100,␣
  ↪"%")
```

```
Оптимальное распределение:
A:  4 ед. →   $4000 (36.4%)
B:  5 ед. →   $5000 (45.5%)
C:  2 ед. →   $2000 (18.2%)

Итоговая доходность: 10.4%
```