

ОПТИМИЗАЦИЯ РАСПРЕДЕЛЕНИЯ СТИМУЛИРУЮЩИХ ВЫПЛАТ НАУЧНО- ПЕДАГОГИЧЕСКИМ РАБОТНИКАМ: МНОГОКРИТЕРИАЛЬНЫЙ ПОДХОД

Соловьева А.А.

Ключевые слова: стимулирующие выплаты, многокритериальная оптимизация, эффективный контракт, python.

Для эффективного распределения стимулирующих выплат научно-педагогическим работникам предлагается многокритериальный подход на основе моделей эффективных контрактов.

Ежемесячно задаётся фонд стимулирующих выплат Φ , требуется определить индивидуальные выплаты H_j по трём критериям: эффективность профессиональной деятельности, исполнительная дисциплина и справедливость распределения.

Математическая модель

Вводится нормированный показатель эффективности (1)

$$P = \sum_{j=1}^N P_j \cdot H_j \rightarrow \max; \quad (1)$$

показатель дисциплины (2)

$$D = \sum_{j=1}^N D_j \cdot H_j \rightarrow \max; \quad (2)$$

и справедливости распределения (3)

$$S = \sum_{j=1}^N \left(H_j - \frac{\Phi}{N} \right)^2 \rightarrow \min; \quad (3)$$

После линейной свертки задача (4) выглядит так

$$\alpha \sum_{j=1}^N P_j \cdot H_j + \beta \sum_{j=1}^N D_j \cdot H_j - \gamma \sum_{j=1}^N \left(H_j - \frac{\Phi}{N} \right)^2 \rightarrow \max, \quad (4)$$

$$\sum_{j=1}^N H_j \leq \Phi,$$

$$H_j \leq H_{max},$$

$$H_j \geq 0, \quad j = 1, 2, \dots, N,$$

$$\alpha + \beta + \gamma = 1.$$

Разработана программа на Python с использованием библиотек Pandas и Puomo решатель Ipopt для нелинейного случая и Cbc для линейного, которая автоматически рассчитывает распределение выплат по введенным данным. Ниже приведен код программы.

```

# Показатели эффективности
E = pd.read_csv('E.csv', sep=";", header=None)
E_array = E.values
# Количество сотрудников и количество критериев
Nrows = len(E_array) # число сотрудников
cols = len(E_array[0])
# Весовые коэффициенты для критериев эффективности
K = pd.read_csv('K.csv', sep=";", header=None)
K_array = K.values[0]
# Количество взысканий по каждому сотруднику
N = pd.read_csv('N.csv', sep=";", header=None)
N_array = N.values[0]
# Ненормированный показатель эффективности
R = [sum(E_array[i][j] * K_array[j] for j in range(cols)) for i in
range(Nrows)]
Pmax = max(R)
# Нормированный показатель эффективности
P = [r / Pmax for r in R]
# Показатель исполнительной дисциплины
D = [1 / (1 + n) for n in N_array]
F = int(10000) # общий фонд стимулирующих выплат
FN = F / Nrows
alp = 0.5
bet = 0.3
gamm = 0.2
model = ConcreteModel()
model.P = Param(range(Nrows), initialize=P)
model.D = Param(range(Nrows), initialize=D)
model.FN = Param(initialize=FN)
model.H = Var(range(Nrows), domain=NonNegativeReals)
model.obj = Objective(
    expr= alp * sum(model.P[i] * model.H[i] for i in range(Nrows)) +
    bet * sum(model.D[j] * model.H[j] for j in range(Nrows)) -
    gamm * sum((model.H[k] - model.FN) ** 2 for k in
range(Nrows)),
    sense=maximize
)
model.con = Constraint(expr=sum(model.H[i] for i in range(Nrows)) <=
F)
solver = SolverFactory('ipopt')
result = solver.solve(model)

```

Список литературы

1. Аргучинцев А.В. Оптимизация распределения стимулирующих выплат научно-педагогическим работникам: многокритериальный подход / А.В. Аргучинцев, Н.Н. Шеломенцева. — DOI 10.17150/2500-2759.2026.36(1).45-52. — EDN JSICMB // Известия Байкальского государственного университета. — 2026. — Т. 36, № 1. — С. 45–52