

УДК 532.5

Определение собственной частоты колебаний жидкости с помощью метода интегральных коэффициентов

А. И. Брунеткин, М. В. Максимов

Одесский национальный политехнический университет, Украина

С помощью метода интегральных коэффициентов получено простое алгебраическое выражение для определения собственной частоты колебаний жидкости со свободной поверхностью в емкости сложной формы (между коаксиальными цилиндрами). Необходимый эмпирический коэффициент определен экспериментально. Отклонение вычисляемых по полученной формуле значений собственной частоты колебаний от экспериментальных величин составляет менее 4% во всем рассматриваемом диапазоне соотношений диаметров, образующих рассматриваемую емкость цилиндров. Такая погрешность приемлема не только для оценочных, но и для инженерных расчетов.

Ключевые слова: частота колебаний жидкости, метод интегральных коэффициентов, безразмерный вид.

За допомогою методу інтегральних коефіцієнтів отримано просте алгебраїчне вираз для визначення власної частоти коливань рідини з вільною поверхнею в ємності складної форми (між коаксіальними циліндрами). Необхідний емпіричний коефіцієнт визначений експериментально. Відхилення обчислюваних за отриманою формулою значень власної частоти коливань від експериментальних величин становить менше 4% у всьому розглянутому діапазоні співвідношень діаметрів, що утворюють розглянуту ємність циліндрів. Така похибка прийнятна не тільки для оціночних, але і для інженерних розрахунків.

Ключові слова: частота коливань рідини, метод інтегральних коефіцієнтів, безрозмірний вигляд.

Using the method of integral coefficients, a simple algebraic expression to determine the natural frequency of fluid with free surface in the capacity of complex shape (between coaxial cylinders) obtained. Necessary empirical coefficient determined experimentally. Deviation calculated by the formula obtained values of the natural frequency of the experimental values is less than 4% over this range diameter ratio, due consideration to the capacity of the cylinders. This error is acceptable, not only for assessment but also for engineering calculations.

Key words: oscillation frequency of fluid, the method of integral coefficients, the dimensionless form.

При проектировании технических систем в ряде случаев достаточно выполнить лишь оценочные расчеты, не прибегая к точному дорогостоящему решению. В первую очередь это относится к начальным стадиям проектирования. На этой стадии для описания учитываемых при проектировании процессов используются математические модели (ММ), относящиеся к верхним уровням иерархической структуры ММ [1]. На этих уровнях законы сохранения, переноса и состояния, а также замыкающие соотношения и краевые условия входят в интегральной форме. Для решения записанных в интегральной форме уравнений может быть использован метод интегральных коэффициентов.

Интегральный коэффициент $\zeta_{P_i, Q}$ некоторой искомой переменной P_i в пространстве Q (геометрические размеры – время) определяется соотношением:

$$\zeta_{P_i, Q} = \frac{\tilde{\delta P}_i}{P_i^\Delta} \in [0, 1], \quad (1)$$

$$\text{где } \tilde{\delta P}_i = \frac{1}{V_{\text{эф}}} \cdot \int_{Q_{\text{эф}}} \delta P_i(q) dq,$$

$$\text{при } \delta P_i = P_i - P_{i \min},$$

$$P_i^\Delta = P_{i \max} - P_{i \min};$$

$$Q_{\text{эф}} \in Q$$

где $\tilde{P}_i$ – среднеинтегральная величина в пространстве Q ;

$V_{\text{эф}}$ – объем подпространства $Q_{\text{эф}}$, в котором δP_i имеет существенное значение;

$P_{i \min}, P_{i \max}$ – минимальное и максимальное значения величины P_i соответственно.

Интегральный коэффициент пространства или отдельных его подпространств определяется следующим образом:

$$\xi_{Q_{\text{эф}}} = \frac{V_{\text{эф}}}{\prod_j q_{j \text{эф}}^\Delta} \in [0, 1]; \quad q_j^\Delta = q_{j \text{эф} \max} - q_{j \text{эф} \min}, \quad (2)$$

где $V_{\text{эф}}$ – объем эффективного пространства;

$\prod_j q_{j \text{эф}}^\Delta$ – объем куба, охватывающего эффективное пространство.

Из вышеизложенного следует, что интегральные коэффициенты ζ и ξ есть нормированные величины, характеризующие полноту переменной P_i в пространстве $V_{\text{эф}}$ и полноту эффективного пространства в пространстве куба с гранями $q_{j \text{эф}}^\Delta$.

Используя интегральные коэффициенты, уравнения сохранения для любой области пространства Q можно записать в форме алгебраических уравнений относительно масштабов переменных P_i^Δ и координат q_j^Δ , с помощью которых можно выразить любой интеграл, входящий в рассматриваемую ММ:

$$\sum_i P_i^\Delta \cdot \zeta_{P_i, Q} \cdot \xi_Q \cdot \prod_j q_j^\Delta = 0 \quad (3)$$

Для предварительных результатов интегральные коэффициенты могут быть оценены на интуитивном уровне или методом экспертных оценок. Для более точных результатов интегральные коэффициенты (а точнее их комплексы) могут быть определены из немногочисленных экспериментов.

В качестве примера рассмотрим задачу определения частоты собственных колебаний жидкости, частично заполняющей емкость, свободный объем которой

образован двумя вертикально расположенными коаксиальными цилиндрами (цилиндрическая емкость с центральным телом). Как известно [3], задачи такого класса, даже для емкостей более простых форм, формулируются в виде уравнений в частных производных и имеют непростые решения.

На Рис.1 проиллюстрированы колебания жидкости в вертикальной цилиндрической емкости без центрального тела, а так же характерные размеры и соотношения важных для процесса параметров. Рассматриваются свободные колебания. Из экспериментальных исследований вытекает, что существует такое $h \geq h_{эф}$, при котором величина заполнения емкости (глубина) перестает оказывать влияние на характер движения свободной поверхности (частоту колебаний). Например, рассматривая колебания свободной поверхности жидкости без ее разрушения в вертикальной цилиндрической емкости, что соответствует отклонению поверхности от положения равновесия до 40° , $h_{эф}$ приблизительно равно радиусу емкости. Тогда в обезразмеренном виде можно считать, что

$$\overline{h_{эф}} = \frac{h_{эф}}{R} = \begin{cases} \bar{h} \text{ при } \bar{h} < 1 \\ 1 \text{ при } \bar{h} > 1 \end{cases} \quad (4)$$

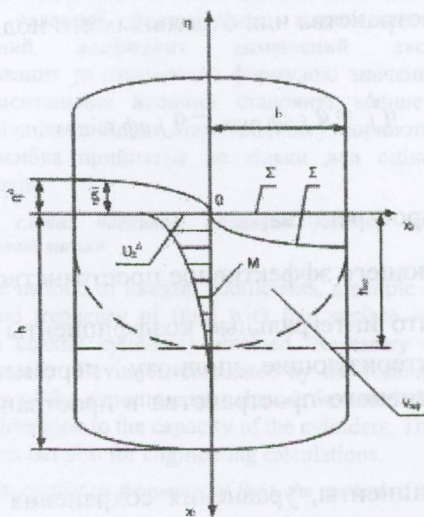


Рис. 1. Иллюстрация колебаний жидкости в емкости,

где η^Δ, v^Δ – масштабы амплитуды и скорости колебаний жидкости;

$V_{эф}, h_{эф}$ – эффективные объем и глубина колебаний жидкости;

l_2 – характерный размер свободной поверхности;

Σ, Σ' – следы свободной поверхности при движении и невозмущенном положении жидкости;

M – след меридианальной плоскости.

Для емкостей других форм также существует величина подобная $\overline{h_{эф}}$.

Пренебрегая диссипацией энергии при колебании жидкости, можно получить верхнюю оценку частоты колебаний. Закон сохранения энергии для колебаний можно записать в виде

$$\Delta K = \Delta U, \quad (5)$$

где ΔK – изменение кинетической энергии;

ΔU – изменение потенциальной энергии.

Полагая, что минимум потенциальной энергии при $\eta=0$, а минимум кинетической при $v=0$, получим:

$$\Delta K = \rho \cdot \int_V \frac{v^2}{2} dV; \quad (6)$$

$$\Delta U = \rho n g \iint_{S_\Sigma} \frac{\eta^2}{2} dx_2 dx_3$$

С учетом (1) и (2) выражения (6) принимает следующий вид:

$$\rho \int_V \frac{v^2}{2} \cdot dV = \frac{\rho}{2} \left(v^\Delta \cdot \zeta_{v_2 V_{\text{эф}}} \right)^2 \cdot \xi_{V_{\text{эф}}} \cdot h_{\text{эф}} \cdot (R-r) \cdot 2\pi \cdot \frac{(R+r)}{2} \quad (7)$$

$$\rho \cdot \frac{n_1 g}{2} \iint_{S_\Sigma} \eta^2 \cdot dx_2 \cdot dx_3 = \frac{\rho n_1 g}{2} \cdot \left(\eta^\Delta \zeta_{\eta S_\Sigma} \right)^2 \cdot \xi_{S_\Sigma} \cdot (R-r) \cdot 2\pi \cdot \frac{(R+r)}{2} \quad (8)$$

В уравнении (7) учтено, что интегрировать практически нужно не по всему объему жидкости, а только по его части – $V_{\text{эф}}$, где скорость жидкости имеет существенное значение.

Приравнявая уравнения (7) и (8) в соответствии с выражением (5), получим:

$$\left(\frac{\eta^\Delta}{v^\Delta} \right)^2 = \frac{\zeta_{v_2 V_{\text{эф}}}^2 \cdot \xi_{V_{\text{эф}}} \cdot h_{\text{эф}}}{n_1 g \cdot \zeta_{\eta S_\Sigma}^2 \cdot \xi_{S_\Sigma}} \Rightarrow \frac{\eta^\Delta}{v^\Delta} = \frac{\zeta_{v V_{\text{эф}}}}{\zeta_{\eta S_\Sigma}} \cdot \sqrt{\frac{\xi_{V_{\text{эф}}}}{\xi_{S_\Sigma}}} \cdot \frac{h_{\text{эф}}}{n_1 g} \quad (9)$$

или

$$\frac{\eta^\Delta}{v^\Delta} = \frac{\zeta_{v V_{\text{эф}}}}{\zeta_{\eta S_\Sigma}} \cdot \sqrt{\frac{\xi_{V_{\text{эф}}}}{\xi_{S_\Sigma}}} \cdot \frac{h_{\text{эф}}}{n_1 g}$$

Закон сохранения массы может быть записан в виде

$$\int_t^{t+T/4} \iint_{S_m} v \cdot dt \cdot dx_1 \cdot dx_3 = \frac{1}{2} \iint_{S_\Sigma} |\eta| \cdot dx_2 \cdot dx_3 \quad (10)$$

где S_m – сечение массы жидкости меридианальной плоскостью

Из уравнения (10) следует, что за четверть периода колебаний через контрольную поверхность S_m пройдет объем жидкости, равный половине суммы увеличения объема ее, с одной стороны, и уменьшения с другой. С использованием интегральных коэффициентов это уравнение можно записать в виде:

$$v_2^\Delta \cdot \zeta_{v_2 S_m} T/4 \cdot \frac{T}{4} \cdot \xi_{S_m} \cdot h_{\text{эф}} \cdot 2(R-r) = \frac{1}{2} \eta^\Delta \cdot \zeta_{\eta S_m} \cdot \xi_{S_\Sigma} \cdot (R-r) \cdot 2\pi \cdot \frac{(R+r)}{2} \quad (11)$$

или

$$\frac{\eta^\Delta}{\nu_2^\Delta} = \frac{\zeta_{\nu_2 S_M T/4} \cdot T \cdot \xi_{S_M} \cdot h_{\phi} \cdot (R-r)}{\zeta_{\eta S_M} \cdot \xi_{S_\Sigma} \cdot (R-r) \cdot \pi \cdot (R+r)} \quad (12)$$

Учитывая (9) из выражения (12) можно выразить:

$$\frac{\pi}{T} = \sqrt{\frac{n_1 g}{h_{\phi}}} \cdot \frac{h_{\phi}}{(R+r)} \cdot \frac{\zeta_{\nu_2 S_M T/4}}{\zeta_{\nu V_{\phi}}} \cdot \frac{\xi_{S_M}}{\sqrt{\xi_{V_{\phi}} \xi_{S_\Sigma}}},$$

или

$$\omega = \sqrt{\frac{n_1 g}{R+r}} \cdot \sqrt{\frac{h_{\phi}}{R+r}} \cdot k_\omega, \quad (13)$$

при

$$k_\omega = 2 \cdot \frac{\zeta_{\nu_2 S_M T/4}}{\zeta_{\nu V_{\phi}}} \cdot \frac{\xi_{S_M}}{\sqrt{\xi_{V_{\phi}} \xi_{S_\Sigma}}}, \quad (14)$$

где k_ω – коэффициент формы емкости.

Исходя из (4) для емкости с центральным телом так же может быть найдена такая граничная величина. Положим, что она определяется соотношением $\sqrt{h_{\phi}/(R+r)}$ из (13). Тогда для «глубокой» воды должно быть

$$\bar{h} = \sqrt{\frac{h_{\phi}}{R+r}} = 1 \quad (15)$$

Теперь выражение (13) в обезразмеренном виде можно записать так:

$$\bar{\omega} = \frac{\omega}{\sqrt{n_1 g/(R+r)}} = \sqrt{\frac{h_{\phi}}{R+r}} \cdot k_\omega \quad (16)$$

Учитывая условие (15) выражение (16) примет вид:

$$\bar{\omega} = \frac{\omega}{\sqrt{n_1 g/(R+r)}} = 1 \cdot k_\omega \quad (17)$$

Если предположение (15) верно, можно ожидать, что коэффициент формы k_ω будет величиной постоянной. Это может быть подтверждено или опровергнуто экспериментально.

Проверка проводилась на емкости из органического стекла $D = 0.195$ м. ($R = 0.0975$ м.) с пятью сменными центральными телами $d = 0.088...0.166$ м. ($r = 0.044...0.083$ м.). Еще одно значение получено для емкости без центрального тела (цилиндрическая вертикальная емкость). Для каждого соотношения диаметра емкости и диаметра центрального тела выполнялось по десять испытаний. В каждом испытании замерялось время 8...10 полных колебаний свободной поверхности. Замер времени проводился с помощью секундомера с ценой деления 0.01 с. Засечка времени вручную. Колебания фиксировались по отраженному от свободной поверхности жидкости световому блику. На основании полученных данных определена оценка математического ожидания и среднее квадратичное отклонение собственной частоты колебаний жидкости ($\bar{m}$, $\bar{\sigma}$) для всех соотношений размеров. Результаты эксперимента сведены в

Табл.1.

Табл. 1. Результаты натурных испытаний

№	$d, \text{ м} / d$	$\omega, 1/c$		$\bar{\omega}$
		$\tilde{m}$	$\tilde{\sigma}$	
1	2	3	4	5
1	0/0	13,64	$0,1 \cdot 10^{-1}$	1,36
2	0,088/0,451	11,446	$1,104 \cdot 10^{-1}$	1,37
3	0,109/0,559	11,02	$1,855 \cdot 10^{-1}$	1,37
4	0,133/0,682	10,771	$1,754 \cdot 10^{-1}$	1,39
5	0,147/0,754	10,469	$3,203 \cdot 10^{-1}$	1,38
6	0,166/0,851	10,629	$3,008 \cdot 10^{-1}$	1,44

Результаты расчета (Табл.1, столбец 5) показывают, что величина $\bar{\omega}$, а, следовательно, и k_{ω} близки к постоянной величине. Усреднив значения $\bar{\omega}$ в результате получим $k_{\omega} = 1,386$. Эта величина менее чем на 4% отличается от $\bar{\omega}$ на всем рассмотренном диапазоне соотношения диаметров емкости и центрального тела, что допустимо не только для оценочных расчетов, но и для инженерных.

На основе вышеизложенного материала можно сделать следующие выводы:

- 1) метод интегральных коэффициентов позволил получить простое алгебраическое выражение для определения собственной частоты колебаний жидкости со свободной поверхностью в емкости сложной формы (между коаксиальными цилиндрами);
- 2) необходимый эмпирический коэффициент определен экспериментально;
- 3) отклонение вычисляемых по полученной формуле значений собственной частоты колебаний от экспериментальных величин составляет менее 4% на всем рассматриваемом диапазоне соотношений диаметров, образующих рассматриваемую емкость цилиндров. Такая погрешность приемлема не только для оценочных расчетов, но и для инженерных.

ЛИТЕРАТУРА

1. Накозин В.Н. Системный подход в построении моделей процессов. - В сб.: Динамика машин и рабочих процессов. №237. Челябинск: ЧПИ, 1979.
2. Накозин В.Н. Метод интегральных коэффициентов в построении и решении математических моделей процессов. //Сб. науч. тр.: Динамика машин и рабочих процессов/ Челябинск: ЧПИ, 1983. - с. 72-78.
3. Моисеев Н.Н., Румянцев В.В. Динамика тела с полостями, содержащими жидкость. - М.: Наука, 1965.

УДК 519.862:866

Об одном подходе к построению модели управления активами и обязательствами (ALM)

С. В. Волкотруб

Харьковский национальный университет радиоэлектроники, Украина

В статье проведен анализ подходов к решению задач управления активами и обязательствами (ALM), выделены преимущества и недостатки существующих подходов, разработана структура модели ALM и предложен новый подход к решению данной задачи для страховой компании по страхованию жизни. Выведена рекуррентная формула для расчета страховых обязательств в предложенной модели ALM.

Ключевые слова: динамическая модель управления активами и обязательствами, оптимальное стохастическое управление, инвестирование.

В статті проведено аналіз підходів до вирішення завдань управління активами та зобов'язаннями (ALM), виділені переваги та недоліки існуючих підходів, розроблено структуру моделі ALM та запропоновано новий підхід до вирішення даного завдання для страхової компанії зі страхування життя. Виведена рекуррентна формула для розрахунку страхових зобов'язань у запропонованій моделі ALM.

Ключові слова: динамічна модель керування активами та обов'язками, оптимальне стохастичне керування, інвестування.

The paper analyzes the approaches to solving the problems of asset and liability management (ALM), highlighted the advantages and disadvantages of existing approaches, developed a model structure ALM and proposed a new approach to solve this problem for the life insurance company. The recurrent formula for calculating insurance liabilities in the proposed model ALM is derived.

Keywords: dynamic model of assets and liability management, optimal stochastic control, investment..

1. Общая постановка задачи и её актуальность

На сегодняшний день модели управления активами и обязательствами являются актуальными для многих инвесторов, например, для инвестиционных компаний, банков, пенсионных фондов, страховых компаний. Поскольку каждый из перечисленных инвесторов имеет свою специфику деятельности, особенности государственного регулирования и влияющие на деятельность факторы, то для каждого из них необходимо построить уникальную модель, отвечающую заданным требованиям. Настоящая работа посвящена построению модели оптимального управления инвестиционной деятельностью страховщика. Поставлены задачи: анализ известных подходов к решению динамических моделей управления активами и обязательствами (ALM); разработка структуры модели управления активами и обязательствами страховой компании по страхованию жизни.

2. Истоки исследования автора. Анализ существующих подходов к решению задач управления активами и обязательствами

Можно выделить 4 основных подхода к решению данных задач: решающие правила, наращивание капитала, стохастическое управление и стохастическое программирование.

Подход, заключающийся в выборе решающих правил, предусматривает выбор стратегий перераспределения активов в виде функции для расчета финансовых показателей (например, текущего значения активов, выплат) в каждый период времени t [2]. Преимущество подхода: решающие правила легко реализуемы и интуитивно понятны для большинства инвесторов, могут быть без труда протестированы на выбранных сценариях, обеспечивают приемлемые доверительные интервалы для рекомендаций. Недостатки подхода: поставленные оптимизационные задачи часто приводят к невыпуклым моделям и к необходимости решения глобально-оптимальных задач.

Второй подход основан на выборе локально-оптимальной стратегии наращивания капитала. Инвестор принимает некоторую функцию полезности и, стремясь максимизировать долгосрочный рост активов, оптимизирует на каждом шаге разность между активами и пассивами. Данные модели описывают набор однопериодных статических представлений о выборе из одних активов, меняющихся с течением времени.

Основная идея подхода, основанного на построении моделей стохастического управления [3, 4], состоит в использовании уравнений в непрерывном времени для описания динамики изменения финансовых переменных. В качестве критерия оптимальности рассматривается интегральный показатель математического ожидания функции полезности в полном соответствии с классическими аксиомами Неймана-Моргенштерна [2]. Недостатки подхода: сложность генерирования доверительных интервалов; могут возникать ошибки моделирования из-за аппроксимации в пространстве состояний; тяжело точно определить общие ограничения на процесс.

Основная идея задач стохастического программирования состоит в генерировании множества сценариев реализации случайных параметров в виде дерева и выборе управлений в вершинах дерева. Любой сценарий должен описывать отдельный, последовательный набор значений параметров, объективно отражающий возможности изменения влияющих факторов, которые должны быть внутренне согласованы в рамках одного сценария, на протяжении всего периода планирования и объективно отражать возможности изменения экономических факторов, учитывая все взаимосвязи, которые могут повлиять на динамику портфеля [1,2]. Преимущество подхода: построение моделей наиболее приближенных к реальности. Недостатки: от выбора множества сценариев зависит адекватность модели; вычислительные трудности, так как число переменных возрастает экспоненциально.

3. Нерешенные проблемы и цели работы

Однако среди существующих подходов к решению задач динамического управления активами и обязательствами (ALM) нет явного лидера, поскольку каждый из них имеет как достоинства, так и недостатки. При использовании

подхода наращивания капитала и стохастического управления политика распределения капитала по активам весьма чувствительна к входным параметрам. При стохастическом управлении и при стохастическом программировании, трудно сгенерировать доверительные пределы, и возможны ошибки моделирования из-за аппроксимации в пространстве состояний. Однако метод стохастического управления имеет концептуальное превосходство над стохастическим программированием, потому что нет необходимости в выборочных сценариях. Целью данной работы является разработка структуры модели ALM, в которой замечен синтез нескольких подходов к построению рассматриваемых моделей, что позволяет гибко использовать достоинства и недостатки каждого из используемых подходов.

4. Предлагаемый подход к построению модели управления активами и обязательствами для страховой компании по страхованию жизни

Выделим основные требования, которые будем учитывать при построении модели:

- 1) необходимость согласования множеств активов и обязательств на каждом шаге;
- 2) особенности государственного регулирования размещения и управления активами, которые идут на покрытие страховых обязательств;
- 3) обеспечение гарантированной доходности инвестиций и влияние инфляции на изменение страховой суммы (индексация обязательств);
- 4) возможность увеличения страховой суммы в зависимости от полученного инвестиционного дохода;
- 5) влияние экономических факторов на динамику изменения активов;
- 6) изменение общего количества действующих договоров страховщика, а соответственно и изменение страховых обязательств.

Построение динамической модели управления активами и обязательствами предложено разделить на три этапа, как показано на рис. 1.

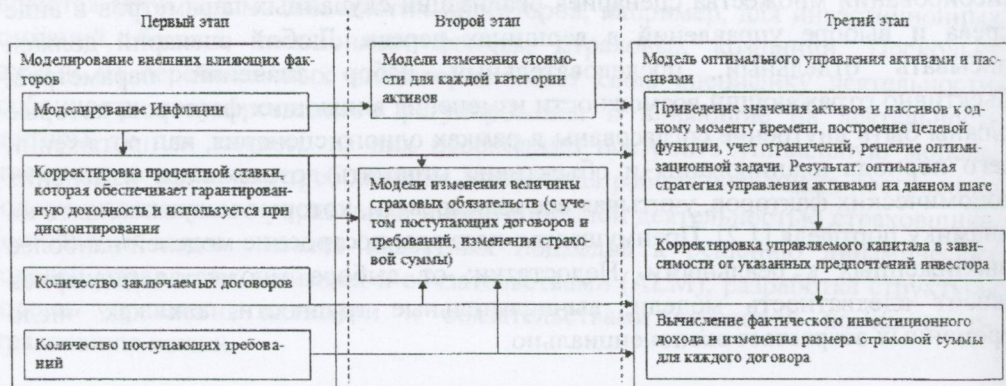


Рис. 1 Этапы построения модели ALM.

Первый этап включает моделирование влияющих стохастических параметров. Затем на втором этапе происходит моделирование изменения

стоимости капитала управляемого портфеля [5] и моделирование изменения величины страховых обязательств. Третий этап заключается в построении модели управления активами и обязательствами. Оптимальное управление активами и принятие управленческого решения о перераспределении средств осуществляется на каждом шаге (в начале каждого года) на основании имеющейся на начало года информации и прогнозируемом изменении факторов неопределенности и динамики изменения активов и обязательств.

Рассмотрим параметры и переменные предлагаемой модели. Обозначим множество моментов принятия решений $t \in T = \{0, 1, \dots, \infty\}$, $(t; t+1)$ - интервал управления (год). Параметры и переменные дискретной модели ALM:

θ – константа, заданная инвестором, определяет максимальную желаемую долю средств, размещенных на банковском счете, которая не участвует в перераспределении средств между активами;

$q^{(m)}$ – количество страховых случаев, предусмотренных в m -м договоре страхования. Страховые случаи – независимые случайные события, вероятность наступления каждого из них изменяется на каждом шаге в зависимости от возраста и состояния застрахованного;

R – заданная инвестором симметричная положительно определенная матрица размерности $(n+1) \times (n+1)$, регулирующая затраты на операции управления вложениями;

$\gamma_i, i = 1, n+2, \sum_{i=1}^{n+2} \gamma_i = 1$ – доля страховых резервов, размещенная в i -ый вид активов, не может превышать установленных государством нормативных значений;

r_t – процентная ставка, учитывающая минимальный гарантированный доход и влияние инфляции;

$t p_x$ – вероятность того, что человек в возрасте x лет проживет еще t лет;

$t V$ – нетто-резерв по одному договору страхования жизни, равный разности актуарной стоимости потока будущих страховых выплат и актуарной стоимостью потока будущих нетто-премий;

$N_{\sum t}^*(1)$ – прогнозируемое количество действующих в момент времени $t+1$ договоров с учетом заключенных $N_t^{new} \sim P(\lambda_t)$ за год $(t; t+1)$ и количества договоров $N_t^{closed} \sim P(\eta_t)$, по которым настали страховые случаи и осуществлена страховая выплата за год $(t; t+1)$;

N_t^* – количество фактически действующих договоров на момент времени t ;

$\pi_t^m, m = 1, N_t^* + N_t^{new}$ – величина нетто-премии по m -му договору страхования, внесенная в начале t -го года;

$b_t^{m,j}(1)$ – величина страховой выплаты по m -му договору, рассчитанная на основании инвестиционного дохода за $(t-1; t)$ год с упреждением в единицу,

выплачиваемая в конце $(t; t+1)$ года при наступлении страхового случая $j = \overline{1, q}$ в течение этого года;

$u_t^i \in R(-\infty; \infty)$, $i = \overline{1, n+1}$ – управляющая компонента, которая определяет количество денежных средств, перераспределяемых за счет банковских средств в i -й вид активов, вычисляемая в момент времени t с упреждением в единицу. При $u_t^i > 0$ перемещаем средства за счет безрискового актива (банковский счет) в i -й вид активов, при $u_t^i < 0$ изымаем средства и перемещаем их в безрисковый вид актива, $u_t^i = 0$ изменений вложений в i -й вид активов не происходит;

$x_i(\omega, t, u)$, $i = \overline{1, n}$, $\omega \in \Omega$, $x_{n+1}(t, u)$, $x_{n+2}(t, u)$ – вложения в выбранные виды активов: акции n эмитентов, кредитование страхователей, банковский счет (депозит) соответственно. Динамика изменения каждой категории активов приведена в [5].

$Act(\omega, t, u_t)$ – величина активов с учетом поступлений и выплат за год $(t; t+1)$ [5];

$$K(\omega, t, u_t) = \sum_{i=1}^{n+2} x_i(t) - \text{капитал управляемого портфеля};$$

$^*CP(t+1)$ – величина страховых обязательств в момент времени $t+1$ по всем действующим договорам страхования жизни.

$$M_{\omega \in \Omega} [Act(\omega, t+1, u_t) | F_t] = Act_t(1) - \text{величина активов, рассчитанная с}$$

упреждением в единицу, где F_t – информация о состоянии активов на момент времени t .

Вид целевой функции обусловлен необходимостью обеспечения вышеперечисленных требований, которые учитывают особенности функционирования страховщика, ограничения на инвестиционную деятельность и затраты на управление:

$$F(\overline{u}_t, t+1) = Act_t(1) - \overline{u}_t^T \cdot R \cdot \overline{u}_t - ^*CP(t+1), \quad t = 0, 1, 2, \dots, \infty;$$

$$F(\overline{u}_t, t+1) \rightarrow \max_{u_t \in G}, \quad (1)$$

$$G: P\{x_i(t+1) \leq \gamma_i K(t+1)\} \geq \alpha_i, \quad 0 \leq \alpha_i \leq 1, \quad i = \overline{1, n+1},$$

$$P\left\{\theta \cdot x_{n+2}(t) \geq \sum_{j=1}^{n+1} u_t^j\right\} \geq \alpha_i, \quad x_i(t+1) \geq 0,$$

при этом $\overline{x(0)}$, $K(0)$, N_0^* – заданные величины.

Ограничения регулируют объемы вложений согласно законодательным нормам и перераспределение средств таким образом, чтобы суммарный объем перераспределенных средств из банковского счета не превышал заданной доли θ средств на этом счету [5].

Построение модели формирования активов описано в работе [5], поэтому рассмотрим формирование страховых резервов и определение общего количества страховых обязательств страховщика.

Рассмотрим m -й договор переменного страхования жизни общего вида сроком на n лет для страхователя, заключившего его в возрасте x :

- страховые выплаты $b_t^{m,q}(1)$ по q -му страховому случаю (дожитие, смерть, несчастный случай и др.) осуществляются в конце года, вычисляемая по итогам года $(t-1; t)$:

$$b_t^{m,q}(1) = \begin{cases} b_{t-1}^{m,q}(1) + \varphi(t), \max_{u_{t-1}} [F(\overline{u_{t-1}}, t)] > 0; \\ b_{t-1}^{m,q}(1), \max_{u_{t-1}} [F(\overline{u_{t-1}}, t)] \leq 0; \end{cases}$$

где $\varphi(t)$ зависит от полученного инвестиционного дохода за год и от количества застрахованных на момент распределения дохода;

- премии размером π_j^m выплачиваются ежегодно в начале каждого года действия договора.

Пусть на момент времени h у страховщика $N_h^*(1)$ действующих договоров переменного страхования общего вида, заключенные на n лет, предусматривающие страховые выплаты размером $b_h^{m,q}(1)$ по $q^{(m)}$ страховым случаям.

Перспективные потери страховщика по одному договору переменного страхования общего вида за год $(j; j+1)$:

$$C_j = v \cdot b_{j+1}^1 \cdot I_1 + \dots + v \cdot b_{j+1}^q \cdot I_q - \pi_j = \sum_{k=1}^q v \cdot b_{j+1}^k \cdot I_k - \pi_j,$$

где I_k - индикатор наступления страхового случая k с изменяющейся на каждом шаге в зависимости от возраста застрахованного вероятностью, v - множитель дисконтирования.

Сумма настоящих стоимостей на момент времени h будущих ежегодных чистых потерь страховщика: ${}_h L = \sum_{j=h}^n v^{j-h} \cdot C_j$.

Обозначим $K(x)$ - пошаговая продолжительность предстоящей жизни (дискретная случайная величина), определяющая число полных будущих лет, прожитых лицом в возрасте x до смерти. При $h > K(x)$ ${}_h L = 0$.

Получим рекуррентное соотношение для нетто-резервов:

$${}_h V = M[{}_h L | K(x) \geq h] = M \left[\sum_{k=1}^q v \cdot b_h^k(1) \cdot I_k - \pi_j + \sum_{j=h+1}^n v^{j-h} \cdot C_j | K(x) \geq h \right] =$$

$$\begin{aligned}
 &= \sum_{k=1}^q v \cdot b_h^k(1) \cdot M[I_k | K(x) \geq h] - \pi_h + p_{x+h} \cdot M[v_{h+1} L | K(x) \geq h+1] = \\
 &= v \cdot p_{x+h} \cdot v_{h+1} V - \pi_h + \sum_{k=1}^q v \cdot b_h^k(1) \cdot M[I_k | K(x) \geq h]
 \end{aligned}$$

Умножим обе части на ${}_h p_x$ и с учетом, что ${}_{h+1} p_x = {}_h p_x \cdot p_{x+h}$, получим размер нетто-резерва для одного договора:

$${}_{h+1} V^{(1)} = \frac{1}{{}_{h+1} p_x} \left({}_h p_x (1 + r_t) ({}_h V^{(1)} + \pi_h^1) - \sum_{k=1}^q b_h^{1,k}(1) \cdot {}_h p_x \cdot M[I_k^1 | K^1(x) \geq h] \right).$$

Верхний индекс (1) в данной формуле обозначает номер договора. Тогда расчет размера страховых обязательств страховщика по всем договорам определяется следующей рекуррентной формулой:

$$\begin{aligned}
 {}^* CP(h+1) &= \sum_{m=1}^{N_h^*(1)} \frac{1}{{}_{h+1} p_x^{(m)}} \left({}_h p_x \cdot ({}_h V^{(m)} + \pi_h^{(m)}) (1 + r_t) - \right. \\
 &\quad \left. - \sum_{k=1}^q b_h^{m,k}(1) {}_h p_x \cdot M[I_k^{(m)} | K^{(m)}(x) \geq h] \right).
 \end{aligned}$$

Данная формула определяет объем обязательств страховщика в каждый момент времени и используется в задаче максимизации (1).

5. Выводы по результатам и направления дальнейших исследований

Проанализированы основные существующие подходы к решению задач управления активами и обязательствами инвесторов, в следствие чего можно сделать вывод, что актуальным для инвесторов является построение моделей, комбинирующих элементы всех четырех проанализированных в данной статье подходов.

Выделены преимущества и недостатки подходов к построению моделей ALM; автором предложен подход к построению моделей ALM для страховых компаний по страхованию жизни, который позволяет рассматривать построение модели в три этапа. Таким образом, учтены не только изменения капитала управляемого портфеля (активов) и обязательств, но и взаимодействие данных множеств с целью оптимального управления денежными средствами путем перераспределения их между выбранными категориями активов, выведена рекуррентная формула для расчета страховых обязательств в предложенной модели.

Практическая значимость работы: анализ структуры и подходов к построению моделей ALM актуален для крупных инвесторов, заинтересованных в оптимальном управлении долгосрочными инвестициями; предложенный подход может использоваться страховщиками для построения моделей ALM.

В дальнейшем автор предложит аналог данной модели в непрерывном времени и методы решения сформулированных задач.

ЛИТЕРАТУРА

1. Carino D.R., Kent T., Myers D.H. The Russel-Yasuda Kasai model: An asset/liability model for a Japanese insurance company using multistage stochastic programming // *Interface*. 1994. V.24. P. 29-49.
2. Worldwide Asset and Liability Modeling / Ziemba W.T., Mulvey J.M. Cambridge: University Press, 1998.
3. Merton R.C. Lifetime portfolio selection under uncertainty: the continuous time case // *Review of economics and statistics*. 1969. N3. P.373-413.
4. Samuelson P. Lifetime portfolio selection by dynamic stochastic programming // *Review of economics and statistics*. 1969. N8. P.239-246.
5. Волкотруб С.В., Герасин С.Н. Пошаговая дискретная модель оптимального управления динамическим балансом страховой компании // *Вісник Харківського національного університету ім. В.Н. Каразіна. Математичне моделювання. Інформаційні технології. Автоматизовані системи управління*. 2009. №863. С.53-61.

УДК 004.021

Складання розкладів для мультипроцесорних систем в оперативному й автономному режимі роботи

Л. С. Глоба, Д. С. Лисенко

*Національний технічний університет України "Київський політехнічний інститут",
Україна*

Ефективність роботи мультипроцесорної системи значною мірою залежить від методів побудови розкладу – способу за яким задачі розподіляються між процесорами з метою зниження часу роботи всієї системи. Ця стаття присвячена частині цього процесу – побудові розкладу на прикладі розробки програмного забезпечення для 4G базової станції.

Ключові слова: мультипроцесорна система, складання розкладу, генетичний алгоритм.

Эффективность работы мультипроцессорной системы в значительной мере зависит от метода построения расписания – способа распределения задач между процессорами с целью снижения времени работы всей системы. Данная статья посвящена части этого процесса – построению расписания на примере разработки программного обеспечения для 4G базовой станции.

Ключевые слова: мультипроцессорная система, составление расписаний, генетический алгоритм.

Efficiency of multiprocessor system usage is strongly dependent on methods of schedule design – the way of task distribution on each processor to decrease overall schedule time. This article is devoted to the part of this process - schedule design on example of software development for 4G base stations.

Key words: multiprocessor system, schedule design, genetic algorithm.

Вступ

Активний розвиток засобів телекомунікацій ставить перед розроблювачам задачу постійного оновлення існуючого програмного забезпечення (ПЗ) і розробки нового, що відповідає новим більш жорстким стандартам. Застосування мультипроцесорних систем особливо ускладнює це завдання.

У зв'язку із цим, метою представленої розробки є зниження часу розробки ПЗ телекомунікаційного обладнання, шляхом автоматизації визначення оптимальних рішень, пошуку помилок, спрощення налагодження й підвищення наочності опису функціонування ПЗ.

Для підвищення ефективності роботи ПЗ телекомунікаційного обладнання пропонується оригінальний метод складання ефективного розкладу для роботи ПЗ систем на кристалі (СНК). Апаратна платформа являє собою СНК, що має кілька типів різних процесорів (далі ресурсів). Алгоритм перетворення цифрового сигналу заданий у вигляді списку завдань із відносинами передування. Для кожного завдання вказується тип ресурсу, на якому він повинен бути виконаний й час необхідний для виконання.

При пошуку способу виконання алгоритмів ПЗ побудованого на основі пропонованого методу складання розкладів враховуються вимоги щодо максимально ефективного використання всіх доступних апаратних ресурсів. При цьому додатковою вимогою є мінімізація часу необхідного для пошуку такого рішення.

З метою зниження часу розробки ПЗ в складі запропонованого методу складання ефективного розкладу вирішуються наступні завдання:

- створення оптимального розкладу виходячи з вимог, які висуваються до ПЗ 4G базових станцій,
- мінімізація часу складання розкладу в оперативному режимі роботи.

2. Архітектура ПЗ СНК.

Архітектура розглянутої СНК представлена на рис.1. Розглянемо докладніше процес її роботи. Генерація списків завдань відбувається на макрорівні, потім завдання надходять на рівень прикладних програм нижньої підсистеми або фізичний рівень. Після обробки на фізичному рівні, списки завдань обробляються ПЗ, що працює на системному рівні. ПЗ, використовуючи інтерфейси операційної системи (ОС), забезпечує виконання отриманих списків завдань із використанням усіх доступних ресурсів, що знаходяться на рівні апаратних абстракцій.

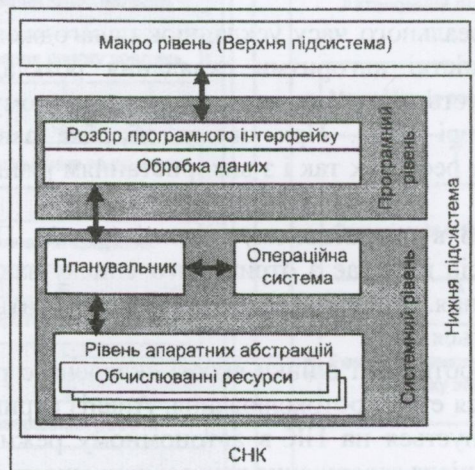


Рис.1. Структурна схема ПЗ СНК.



Рис.2. Емуляція роботи ПЗ у режимі налагодження.

Робота ПЗ у режимі реального часу ускладнює налагодження й не дозволяє застосовувати ресурсномісткі алгоритми складання розкладів. Тому аналіз списку завдань виконується автономно на програмному комплексі чи персональному комп'ютері (ПК). При цьому обробка завдань може бути виконана як на реальних ресурсах так і з використанням різних рівнів емуляції (рис.2)

Розглянемо роботу ПЗ в оперативному режимі (рис.3). Планувальник при наявності вільних ресурсів вибирає з отриманого списку завдань відповідні й направляє їх на виконання. Коли черга завдань не містить завдань, робота планувальника завершується.

Для налагодження роботи ПЗ і оцінки якості складеного розкладу в процес обробки завдань додається етап протоколювання. Аналіз отриманого протоколу і його оптимізація виконується на ПК в автономному режимі (рис. 4) у два етапи. На першому етапі після завершення всіх завдань, що отримуються від ПЗ, яке виконується на СНК знімається протокол роботи. Після аналізу даних протоколу, у разі отримання висновків про необхідність внесення змін у список завдань, обладнання перезапускається й на другому етапі, у момент, коли список завдань готовий до обробки, в нього вносяться необхідні зміни. Після обробки зміненого списку завдань знову формується протокол роботи, він аналізується з точки зору корисності внесених змін.

При пошуку способу виконання алгоритмів ПЗ побудованого на основі проміжного методу складання розкладів враховуються вимоги щодо максимального ефективного використання всіх доступних апаратних ресурсів. При цьому додатково вимога до мінімізації часу необхідного для такого рішення.

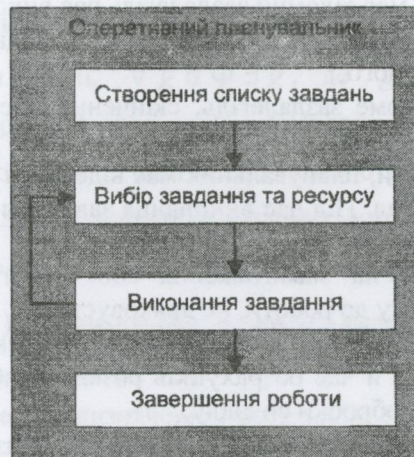


Рис.3. Схема роботи планувальника в оперативному режимі.



Рис.4. Спільна робота автономного й оперативного планувальників.

Припущення при розв'язанні задачі.

При аналізі й розв'язанні задачі використовувалися наступні припущення, що впливають зі специфіки предметної області.

1. Виконання завдання не переривається
2. Виконання завдання не розділяється між ресурсами
3. Кожен ресурс може виконувати тільки одне завдання (немає поділу часу)
4. Критерієм оптимальності складеного розкладу є його довжина. Якщо завдання W_i виконується за час, C_i , то загальний час виконання

$$\sum_{i=0}^N C_i \text{ для } N \text{ завдань.}$$

5. Кожне завдання має відомий заздалегідь час виконання.
6. Кожному завданню співвіднесений список завдань, які повинні бути виконані раніше його.
7. Система має відоме заздалегідь, скінченне число ресурсів кожного типу.
8. На початку роботи, планувальник має відомості про всі завдання, що становлять розклад. Під час виконання завдання не видаляються й не створюються нові.
9. Час, необхідний на завантаження або вивантаження завдання й підготовку ресурсу до роботи, не враховується.
10. В оперативному режимі, задача складання розкладу виконується на окремому ресурсі й час розрахунків розкладу не повинний впливати на загальний час обробки сигналу.
11. Кожне завдання має рівень пріоритету, який призначений попередньо. При інших рівних завдання, що має більш високий пріоритет, повинно бути виконано раніше.

3. Побудова математичної моделі обчислювального процесу.

Математична модель обчислювального процесу може бути представлена в вигляді графів «завдання - вершини» та «завдання - дуги».

Вхідні дані.

Вхідні дані для алгоритму перетворення цифрового сигналу в процесі роботи БС складаються з наступного набору даних:

- Множина завдань W і задане в ньому відношення передування: кожному із завдань $\omega \in W$ зіставляється деяка підмножина P її попередників. Причому для деяких завдань $P = \emptyset$.
- Кожному завданню зіставляється ціле ненегативне значення пріоритету φ_ω і індекс типу ресурсу, якому вона відповідає r_ω . Час виконання завдання t_ω . Унікальний у межах списку індекс k_ω використовується для запобігання невизначеності при сортуванні завдань.
- Множина доступних ресурсів R задана заздалегідь і містить у собі дані про кількість доступних ресурсів кожного типу $R = \{R_0, R_1, \dots, R_l\}$.

Вхідні дані можна представити у вигляді графу $G = (S_n, S_a)$, де $S_n = \{n_i, i = 1..N\}$ – множина вузлів, які представляють завдання, та $S_a = \{(n_i, n_j) | n_i \rightarrow n_j\}$ – скінченна множина дуг, які представляють відношення передування завдань. При цьому кожному завданню n_i ставиться у відповідність час виконання w_i .

Формально описати складання розкладу можна як множину Φ функцій g , які дозволяють відобразити граф G на архітектуру ПЦОС $S_p = \{p_k, k = 1..P\}$,

де P – кількість ресурсів. Функція $g(n_i)$ повертає множину ресурсів p_k , на яких n_i виконується. Далі, $\forall g \in \Phi \exists \bar{g}$, де $\bar{g}(p_k)$ – множина вузлів, призначених на процесор p_k функцією g . Аналогічно $\forall g \in \Phi \exists S_g$, де S_g – множина допустимих розкладів S , які можуть бути отримані за допомогою g . Таким чином, $S_g \subseteq S_i$, де S_i – множина існуючих допустимих відображень G на S_p – так званий *простір рішень*. Для заданого розкладу $s \in S_g$, якщо $n_i \notin \bar{g}(p_k)$, то $s(n_i, p_k)$ не визначене, в іншому випадку воно відображає час початку виконання n_i на ресурсі p_k .

Використовуючи наведені визначення, розклад s , який створюється g є допустимим, якщо наступні вирази істинні для усіх пар вузлів $(n_i, n_j) \in S_a$.

$$\forall p_l \in g(n_i) : \text{якщо } p_l \in g(n_i) \text{ то } s(n_i, p_l) + w_i \leq s(n_j, p_l),$$

$$\text{якщо } p_l \notin g(n_i) \text{ то } \exists p_k \in g(n_i) \mid s(n_i, p_k) + w_i + s(n_j, p_l)$$

$$\forall p_k \in S_p, \forall n_i \in S_n, n_j \in \bar{g}(p_k) :$$

$$\text{якщо } n_i \neq n_j \wedge s(n_j, p_k) \geq s(n_i, p_k) \text{ , то } s(n_j, p_k) \geq s(n_i, p_k) + w_i$$

Граф «завдання - вершини».

Найбільш очевидним є представлення списку завдань у вигляді орієнтованого графа $\langle M, N, T \rangle$ де M – завдання, N – умови передування, T – відображення множини у прямий добуток $M \times M$, що ставить у відповідність кожній залежності $j \in N$ упорядковану пару завдань, перша з яких повинна передувати другий.

Крім візуальної оцінки топології списку завдань, граф $G_1 = \langle M, N, T \rangle$ дозволяє провести перевірку графа на відсутність контурів. Завдання, що становлять контур, не можуть початися й чекають один одного.

Граф «завдання - дуги».

На практиці представлення списку завдань, у вигляді графа «завдання - вершини» не завжди зручно. Як приклад можна привести наступний граф, отриманий на основі реальних даних, що й становить собою частина розрахунків фрейму WIMAX (рис.5).

На ньому практично неможливо простежити залежності між завданнями, і він не придатний до використання.

Щоб уникнути таких незручностей, було вирішено представляти граф залежностей між завданнями, вважаючи завдання дугами. У цьому випадку вершинами графа стають події початку й закінчення завдань.

$$G_2 = \langle M', N', T' \rangle$$

Застосування таких перетворень до наведеного вище графа завдань дає результат більш наочний і зручний у використанні (рис. 6).

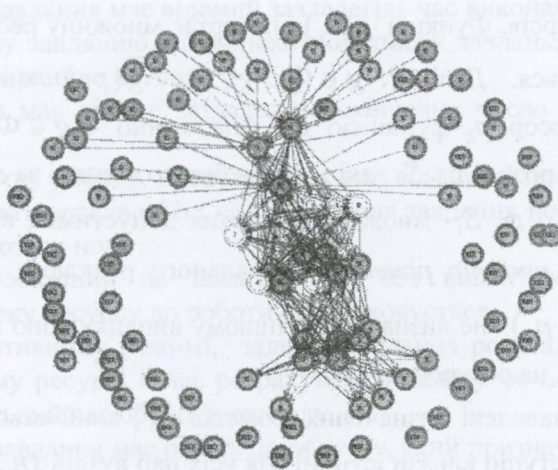


Рис. 5. Приклад графа "завдання - вершини" з більшою кількістю зв'язків.

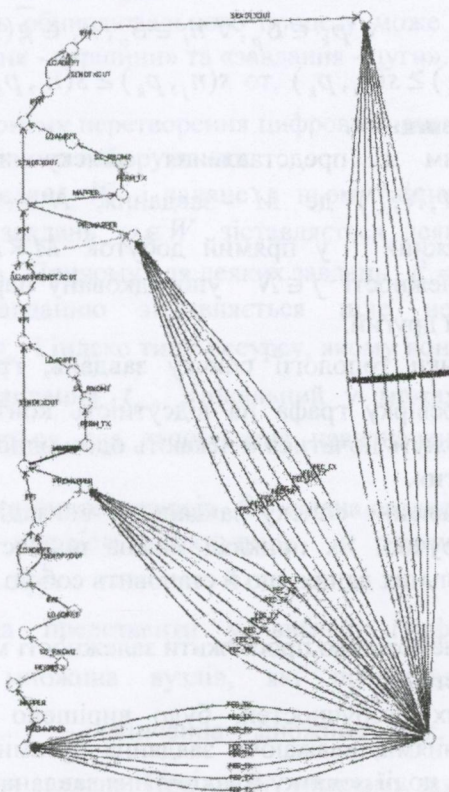


Рис. 6. Приклад графа "завдання - дуги".

Для оцінки стану виконання списку завдань у будь-який момент часу використовується діаграма Гантта (стрічкова діаграма).

4. Аналіз обчислювального процесу.

Для аналізу обчислювального процесу використовуються 4 основних методи: пошук контурів, побудова графа «завдання - дуги», приведення графа до ярусно паралельного вигляду й розрахунок критичного шляху (КШ).

Пошук контурів.

Пошук контурів здійснюється шляхом перебору, у якому будується постійно нарощуваний шлях. Він або замикається на вершину, що вже міститься в ньому або зупиняється в тупиковій вершині, у цьому випадку, дуга до такої вершини може бути вилучена з розгляду й зі шляху. Відзначимо, що формалізувати процес розмикання контурів неможливо. Апріорно припустимо будь-яке упорядкування завдань, і єдина можливість вирішення колізії - експертний аналіз проблеми поза рамками математичної моделі.

Побудова графа «завдання - дуги».

Будемо вважати, що на цьому етапі граф завдань вільний від контурів. У цьому випадку першим етапом буде побудова транзитивного замикання $\bar{P}$ відносини P для знаходження всіх попередників для кожного із завдань.

Для визначення завдань, що мають спільний набір попередників, розглянемо сукупність множин $\bar{P}_\omega, \omega \in W$ і вилучимо із цієї сукупності повтори, у результаті виходить деякий набір множин $\pi_\alpha, \alpha \in A$, і кожному $\omega \in W$ буде відповідати індекс α , такий що $\bar{P}_\omega = \pi_\alpha(\omega)$.

Для розбивки множини завдань на групи розглянемо для кожного π_α розбивку $\Pi_\alpha = \pi_\alpha \cup \left(\frac{W}{\pi_\alpha}\right)$. Розбивка $\bar{\Pi}_\alpha = \{\rho_\beta\}_{\beta \in B}$, дрібніше кожного з розбивок співмножників, отже кожне з множин π_α є об'єднанням деякого набору множин ρ_β . Позначимо через $B(\alpha) \subset B$, набір індексів для яких $\pi_\alpha = \bigcup_{\beta \in B(\alpha)} \rho_\beta$. Тому що об'єднання всіх множин ρ_β збігається з W і ці множини не перетинаються, то кожний елемент W належить рівно одному з них. Іншими словами, кожному $\omega \in W$ однозначно зіставляється індекс $\beta(\omega) \in B$, що $\omega \in \rho_{\beta(\omega)}$.

Уважаючи на те, що обрані нами множини A і B диз'юнктивні, будемо граф $\langle M, N, T \rangle$, де $M = A \cup B$; $N = W \cup F$; $T(\omega) = (\alpha(\omega), \beta(\omega))$. Щоб зібрати разом усі фрагменти отриманого двочасткового графа, необхідні додаткові дуги. Визначимо множину фіктивних дуг F як дуги відповідні до всіх пар (α, β) , при чому початок дуги відповідає $\beta \in B(\alpha)$, а кінець $\alpha \in A$.

Отриманий граф може бути істотно спрощений, якщо дотримуватися наступним правил:

- Якщо існує обхідний шлях для дуги $j_f \in j$ не утримуючий дуги j_f , те така дуга може бути вилучена із графа.

• Якщо якась дуга $j_f \in F$ єдина дуга вихідна з вершини j_f або вхідна у вершину, то така дуга може бути вилучена шляхом склеювання її початку й кінця в одну вершину.

Приведення графа до ярусно паралельного вигляду.

При оцінці топології графа, важливо подати інформацію в зручному для користувача вигляді для цього граф приводиться до ярусно паралельного вигляду. Це така розбивка вершин графа на яруси при яким:

1. Вершини першого ярусу не мають попередніх вершин, а вершини останнього ярусу наступних
2. Вершини будь-якого ярусу не мають попередніх у групі вершин наступного ярусу.
3. У кожному ярусі повинна бути хоча б одна дуга з попереднього ярусу.
4. Вершини в межах одного ярусу не з'єднуються дугами.

Така розбивка завжди можлива, і проводиться з використанням наступного алгоритму:

1. Будується матриця суміжності вершин.
2. У черговий ярус вибираються вершини з нульовим напівступенем заходу, вони відповідають нульовим стовпцям. Рядки відповідні до обраних стовпців зануляються.
3. Другий крок повторюється доти поки не будуть упорядковані всі вершини.

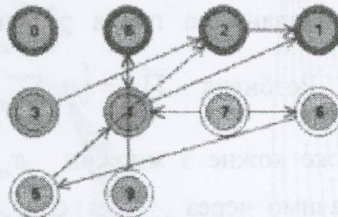


Рис. 7: Приклад графа "завдання - вершини" до приведення до ЯПВ.

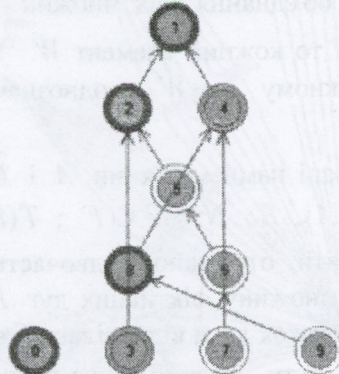


Рис. 8: Приклад графа "завдання - вершини" після приведення до ЯПВ.

Розрахунок критичного шляху.

Граф отриманий у результаті перетворення списку завдань може бути використаний для одержання важливих для розрахунків кількісних характеристик, насамперед пов'язаних із часом виконання.

Позначимо час виконання кожного завдання $\omega \in W$ як $t_\omega > 0$. При розгляді графа «завдання - дуги», для фіктивних завдань вважаємо $t_\omega = 0$. Сума часу завдань для шляху назвемо довжиною шляху. Пошук критичного шляху, це необхідна частина аналізу списку завдань при перевірці можливості паралельного виконання й оцінки часових характеристик розглянутого списку.

Алгоритм пошуку критичного шляху (КШ) схожий з алгоритмом Дейкстри з поправкою на те, що шукається шлях не мінімальної а максимальної довжини.

Алгоритм пошуку критичного шляху.

Кожній події $i \in M$ зіставляється число $v(i)$, називане раннім настанням події i і рівне найбільшій довжині шляхи від початкової події до i .

Ранні початки подій можуть бути визначені зі співвідношення

$$v = \max \{t_j + v(\text{початок } j) \mid \text{кінець } ji\},$$

при чому для початкової події $i_0, v(i_0) = 0$.

Усі розрахунки не можуть бути завершені раніше ніж відбудеться заключна подія, у такий спосіб довжина критичного шляху це мінімальний час завершення розрахунків.

Аналогічно алгоритму Дейкстри розіб'ємо множину вершин $M = M_0 \cup M_1 \cup M_2$, де M_0 - множина вершин з розрахованим часом настання ранньої події, M_1 - множина вершин з розрахованим часом настання ранньої події й M_2 - множина вершин для яких обчислення ще не починалися. Крім того кожній вершині $i \in M$ зіставляється ціле ненегативне число $r(i)$ - кількість ще не переглянутих дуг з кінцем в i та число $v(i)$ - раннє настання події i .

Початковий стан:

$$M_0 = \emptyset, M_1 = \{i\}, M_2 = \frac{M}{M_1}, M = 0, r = \{j \mid \text{кінець } j = i\}, j \in M$$

Крок алгоритму:

1. Якщо $M_1 = \emptyset$, то обчислення завершуються.
2. Інакше, вибирається будь-яка дуга $i_1 \in M_2$, яка переноситься в M_0 .
3. Для кожної дуги j вихідної з i_1 , та її кінця i_2 , порівняти $v(i_2)$ і $v' = t(j) + v(i_1)$ та збільшити $v(i_2)$ до v' якщо v' більше. Зменшити $r(i_2)$ на 1 і при $v(i_2) = 0$ перенести i_2 з M_2 в M_1 .

Аналогічно раннім настанням подій можуть обчислюватися пізні настання. Взявши в якості вихідних моментів для кожної події критичний час проекту й вважаючи цей час уже обчисленим для заключної події, обчислюємо заново пізні настання для всіх інших подій по формулі:

$$V = \min \{-t_j + V(\text{кінець } j) \mid \text{початок } ij\}.$$

Знаходження пізніх моментів аналогічно знаходженню ранніх початків.

Для кожного завдання обчислимо величину $d = V(\text{кінець } i) - t(j) - v(\text{початок } j)$ і назвемо її резервом часу для завдання. Очевидно, що завдання лежачі на критичному шляху мають нульовий резерв часу.

Нижче наведений приклад розрахунків критичного шляху.

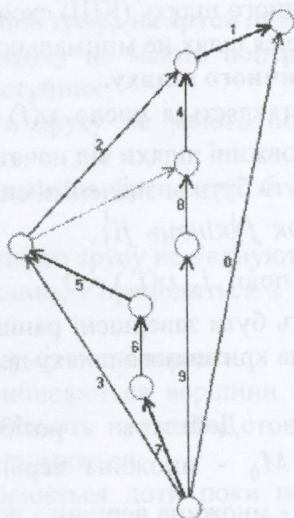


Рис. 9: Приклад розрахунків критичного шляху.

Довжина критичного шляху 46 тактів.

Завдання, що лежать на критичному шляху, відзначені сірим.

Табл. 1: Приклад розрахунку критичного шляху.

Ім'я	Вага	Раннє настання події	Пізнє настання події	Резерв часу
0	10	0	36	36
8	10	12	26	14
2	2	2	10	8
1	2	0	0	0
3	10	4	36	32
4	10	2	2	0
7	2	44	44	0
6	2	42	42	0
5	30	12	12	0
9	10	22	36	14

З наведеного графа видно, що завдання 1, 4, 5, 6 і 7 лежать на критичному шляху й відсутність затримок у них виконанні важливо для складання оптимального розкладу. Так само можна відзначити, що завдання 2 має невеликий резерв часу й слід звернути увагу на забезпечення його своєчасного виконання.

5. Алгоритми побудови розкладу

Жорсткі вимоги, пропоновані до алгоритмів побудови розкладу в оперативному режимі роботи, унеможливають застосування евристик. Замість цього необхідно використовувати алгоритми, що вимагають мінімальної кількості процесорного часу й ресурсів. Тому запропоновано 4 типу алгоритмів. Алгоритми 1, 2 і 3 можуть використовуватися в оперативному режимі, хоча алгоритм 3 вимагає більшу кількість ресурсів. Через високі вимоги до обчислювальних ресурсів алгоритм 4 може бути реалізований тільки на ПК. Питання про вибір використовуваного алгоритму в кожному конкретному випадку може бути вирішено залежно від кількості доступних апаратних ресурсів.

Алгоритм 1: «Швидкий».

Алгоритм, наведений нижче використовує мінімальну кількість ресурсів для складання розкладу, він запускається щораз при старті планувальника й при завершенні завдання.

1. Для кожного завдання $\omega \in W$, де W - розглянута множина завдань, що відповідає наступним умовам :
 - а. Призначена на конкретний тип ресурсу H_k .
 - б. Що має список залежностей P_ω .
2. Множина завдань W поділяється на K підмножин відповідних до кожного типу ресурсів.
3. Для $k = \overline{1, K}$ множин, M_k завдань.
4. Поки $M_k^i, i = \overline{1, N_k}$ обробляємо елементи множини k .
5. Якщо ресурс P_k^i вільний.
6. Нехай $j = 1$.
7. Поки $j < M_k$.
8. Якщо завдання T_j не має активних залежностей.
9. Виконати завдання T_j на ресурсі P_k^i .
10. Інакше, наступне i .
11. Інакше, наступне j .
12. Наступний k .

Алгоритм 2: «Критичний шлях» - модифікація «Швидкого» алгоритму використанням результатів розрахунків критичного шляху.

Будь-які затримки на критичному шляху приводять до збільшення загального часу розкладу. Щоб не допустити цього встановимо пріоритет кожного завдання рівним резерву часу. Це приведе до того, що завдання, що лежать на критичному шляху, будуть виконуватися в першу чергу, що знижує ймовірність затримок для критичних завдань.

Розрахунки критичного шляху для вихідних даних виконується заздалегідь за допомогою автономного ПЗ.

1. Для кожного завдання $\omega \in W$, де W - розглянута множина завдань, що відповідає наступним умовам :
 - a. Призначена на конкретний тип ресурсу H_k .
 - b. Що має список залежностей P_ω .
 - c. Що має пріоритет Π_ω .
2. Множина завдань W поділяється на K підмножин відповідних до кожного типу ресурсів.
3. Для $k = \overline{1, K}$ множин, M_k завдань.
4. Поки $M_k^i, i = \overline{1, N_k}$ обробляємо елементи множини k .
5. Якщо ресурс P_k^i вільний.
6. Нехай $j = 1$.
7. Поки $j < M_k$.
8. Якщо завдання T_j не має активних залежностей.
9. Виконати завдання T_j на ресурсі P_k^i .
10. Інакше, наступне i .
11. Інакше, наступне j .
12. Наступний k .

Алгоритм 3: «Критичний шлях з резервуванням» - модифікація алгоритму «Критичний шлях» веденням резервування ресурсів для критичних завдань.

Розглянемо окремий випадок вихідних даних (Таблиця 2). Усі завдання використовують той самий тип ресурсів.

Табл. 2: Вихідні дані.

Номер завдання	Час виконання	Залежності
1	10	-
2	5	1
3	5	1
4	9	-
5	5	2,3
6	3	4

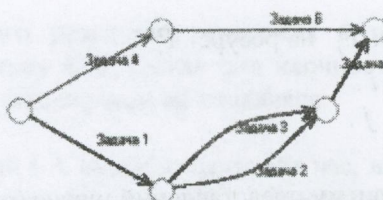


Рис. 10: Граф "завдання - дуги".

Граф відповідний цим вихідним даним наведений на малюнку (рис. 10). Критичний шлях позначений червоним. Застосування алгоритму 2 для випадку із двома доступними ресурсами дає результат в 22 такту й розклад наступного вигляду (рис. 11).

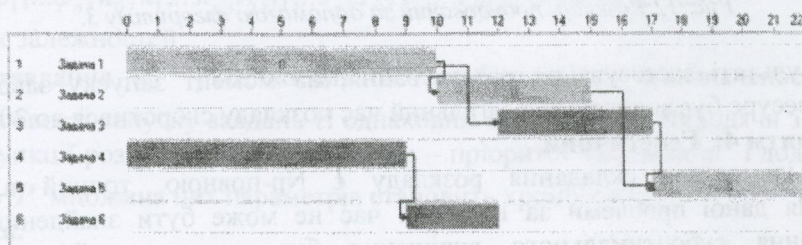


Рис. 11: Розклад побудоване за допомогою алгоритму 2.

Нескладно помітити, що виконання завдання 6 відразу після завдання 4 призвело до затримки виконання завдання 3, що лежить на критичному шляху і як наслідок до збільшення часу розкладу в цілому. Щоб уникнути таких ситуацій і знизити ймовірність затримок на критичному шляху введемо в алгоритм етап резервування ресурсів під критичні завдання. Таким чином, алгоритм здобуває наступний вид:

1. Для кожного завдання $\omega \in W$, де W - розглянута множина завдань, що відповідає наступним умовам :
 - a. Призначена на конкретний тип ресурсу H_k .
 - b. Що має список залежностей P_ω .
 - c. Що має пріоритет Π_ω .
2. Множина завдань W поділяється на K підмножин відповідних до кожного типу ресурсів.
3. Для $k = 1, \overline{K}$ множин, M_k завдань.
4. Поки $M_k^i, i = 1, \overline{N_k}$ обробляємо елементи множини k .
5. Якщо ресурс P_k^i вільний.
6. Нехай $j = 1$.
7. Поки $j < M_k$.
8. Якщо завдання T_j не має активних залежностей.

9. Виконати завдання T_j на ресурсі P_k^i .

10. Інакше, наступне i .

11. Інакше, наступне j .

12. Наступний k .

Результат роботи алгоритму представлений нижче (рис. 12).

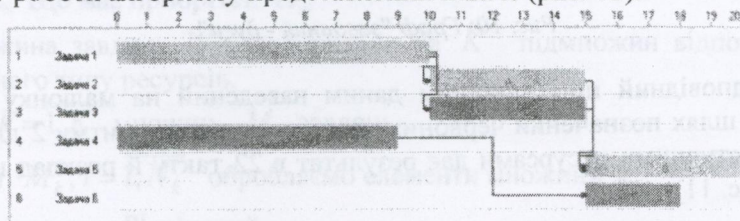


Рис. 12: Розклад, побудований за допомогою алгоритму 3.

У результаті застосування резервування, на момент запуску завдання 3, вільний ресурс був знайдений, і загальний час розкладу скоротився до 20 тактів.

Алгоритм 4: Генетичний.

Оскільки задача складання розкладу є Np-повною, точний алгоритм вирішення даної проблеми за кінцевий час не може бути знайдений. Для знаходження субоптимального вирішення був використаний генетичний алгоритм (ГА) – один з видів стохастичних алгоритмів, який містить у собі евристичні методи, що моделюють еволюційні механізми відбору й пошуку можливих рішень. Цей метод добре зарекомендував себе в вирішення задач складання розкладів [8],[9],[10].

Опис алгоритму.

Схема алгоритму наведена на рисунку 13.

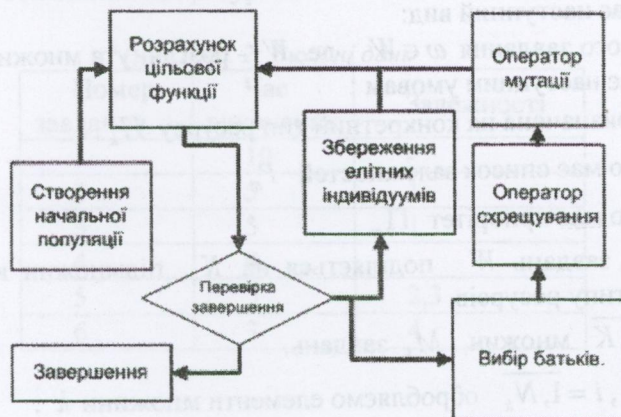


Рис. 13: Схема генетичного алгоритму.

Створення початкової популяції.

Створення початкової популяції проводиться шляхом випадкового розміщення пріоритетів завданням і внесенням випадкової невеликої кількості додаткових залежностей. Так само дозволяє знизити кількість ітерацій до

одержання субоптимального результату внесення в популяцію індивідуума створеної на основі алгоритму КШ. Однак для наочності результатів подібний індивідуум при проведенні розрахунків не додався.

Цільова функція.

У якості цільової функції ГА використовувався час, що вийшов у результаті складання розкладу за допомогою «швидкого» алгоритму розглянутого раніше. У такий спосіб цільова для списку з завдань може бути представлена у вигляді $\Theta(W, P_0, P_1, \dots, P_n)$

У цьому випадку кожне з розглянутих розкладів презентовано у вигляді множини пріоритетів завдань, що входять у розклад і набору додаткових залежностей P'_ω

$\Theta(\varphi_0, \varphi_1, \dots, \varphi_n, k_0, k_1, \dots, k_n, r_0, r_1, \dots, r_n, P_0, P_1, \dots, P_n, P'_0, P'_1, \dots, P'_k)$, де k - число додаткових залежностей

Індекс завдання k_ω уведено для того щоб уникнути невизначеності при виборі черговості запуску завдань із однаковим пріоритетом. Виходячи з опису цільової функції розглянемо два типи генів – пріоритет завдання φ і додаткова залежність P множина цих параметрів становить хромосому.

Мутація.

Під мутацією розуміється введення одиничної зміни в хромосому індивідуума. У розглянутому випадку це відповідає додаванню або видаленню додаткового зв'язку або зміні пріоритету завдання на 1. Вибір типу мутації здійснюється випадковим образом із імовірністю 0.5. Щоб уникнути появи петель у графі при додаванні нових залежностей необхідно враховувати ярус вузла. Для цього граф завдань приводиться до ярусно-паралельної форми й при додаванні нової залежності використовується множина завдань, що мають більш високий номер ярусу стосовно поточної.

Вибір батьків.

На етапі генерації наступного покоління необхідно забезпечити ефективний відбір батьківських індивідуумів для схрещування. Виходячи з рекомендацій до створення ГА була обрана наступна функція розподілу батьківських індивідуумів:

80% відібраних з 20 % популяції, що мають найкращу цільову функцію, 10% з 10% популяції, що мають найгірше значення цільової функції й 10% інших.

Схрещування.

При схрещуванні для одержання індивідуума нащадка використовується комбінація генів батьківських індивідуумів, кількість генів від кожного з батьківських індивідуумів визначається випадковим образом.

Збереження елітних індивідуумів.

Щоб уникнути регресії при роботі алгоритму, частина індивідуумів, що мають найкращий результат, фіксується й не піддається мутаціям. Дослідним шляхом установлене, що їх число повинне бути невеликим приблизно 0.05% від розміру популяції, але не менше 1.

Завершення алгоритму.

Час роботи алгоритму може, обмежуватися за часом, кількості розглянутих

покоління, або ознаці виродження популяції – коли значення цільової функції середнє по популяції починає дорівнювати кращому значенню.

Вибір оптимальних параметрів генетичного алгоритму.

В процесі роботи було помічено, що час потрібний ГА для знаходження субоптимального результату значною мірою знаходиться в залежності від наступних параметрів: розмір популяції, кількість пар для схрещування та кількість мутацій. Для знаходження оптимальних значень цих параметрів було виконане дослідження, результати якого представлені на малюнку нижче (рис. 14).

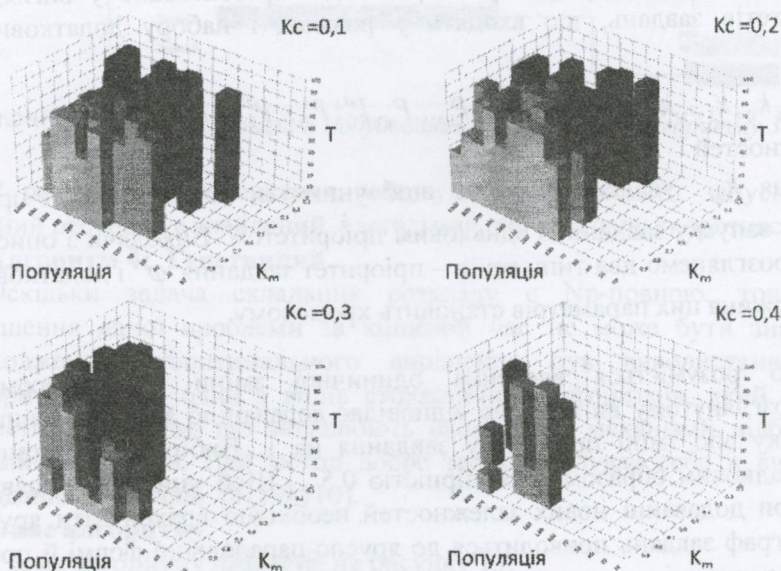


Рис. 14: Результати вибору оптимальних параметрів генетичного алгоритму.

На малюнку позначено K_c – коефіцієнт кількості пар для схрещування відносно розміру популяції, K_m – коефіцієнт кількості мутацій відносно розміру популяції. Для оцінки ресурсомісткості (тобто часу роботи) ГА використовувалася кількість розрахунків цільової функції у процентах до її мінімального значення (T). На діаграмах позначені лише результати в яких субоптимальне значення було досягнуте за менше чим 800 000 розрахунків цільової функції. Таких результатів для $K_c > 0.4$ не було знайдено жодного.

Для проведення дослідження був вибраний зразок №1. Таким чином, для розглядаємої множини зразків були вибрані наступні параметри ГА: розмір популяції 70, кількість пар для схрещування 7 кількість мутацій 35 за одну ітерацію алгоритму.

Оптимізація складання розкладу в оперативному режимі роботи планувальника.

Для оптимізації довгі розклади використовується ГА у зв'язку з тим що його результатом роботи є не тільки складений розклад але й параметри індивідуума

з найкращою цільовою функцією: набір пріоритетів для кожного завдання й список додаткових зв'язків. Застосування цих параметрів до вихідного списку завдань викличе зменшення загального часу розкладу при використанні алгоритму 2 в оперативному режимі.

Розглянемо цей процес на прикладі наведеному в (Таблиця 1). Як було показано вище в даному окремому випадку з використанням алгоритму 2 створюється неоптимальний розклад довжиною 22 такту. Результат роботи ГА представлений у таблиці (Таблиця 2).

Табл. 2: Результати роботи ГА.

Номер завдання	Пріоритет	Додаткові залежності
1	9	
2	9	
3	9	
4	8	
5	9	
6	0	1

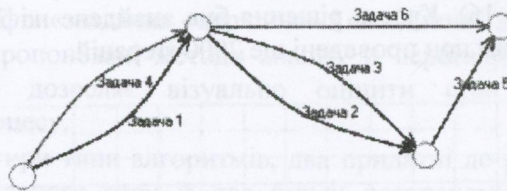


Рис. 15: Результати роботи ГА

Граф «завдання - дуги», відповідний до знайденого рішення, наведений вище на рисунку 15. Використовуючи алгоритм 2 до зміненого списку завдань, ми одержуємо рішення аналогічне отриманому за допомогою алгоритму 3 (див. рис. 12) менше на 2 такта в порівнянні з вихідним.

6. Результати експериментів.

Запропонований алгоритм реалізований мовою програмування C# у рамках створення комплексу «Offline Sceduler». Були проведені експерименти з метою перевірки ефективності алгоритму й налаштування його основних параметрів: імовірності появи мутації, імовірності застосування схрещування, розміру популяції й числа елітних індивідумів.

У якості прикладів використовувалася серія з 32 списків завдань виникаюча в ході формування WIMAX фрейму. Останній список завдань був узятий із процедури формування фрейму LTE і характеризується високою можливістю до распаралеливанню. Усі списки використовують два типи різних ресурсів. Усі зразки вихідних даних містять від 97 до 1087 завдань і від 116 до 87317 зв'язків.

Диаграма (рис. 16) показує результати розрахунків формування розкладу при кількості ресурсів першого типу 2 і другого типу 4 для наступних алгоритмів

«швидкий», «критичний шлях» (КШ), «критичний шлях з резервуванням» (КШР) і генетичний. Для генетичного алгоритму використовувалася популяція розміром 200 і розрахунки закінчувалися при досягненні 150ї ітерації. Результати наведені відносно «швидкого» алгоритму.

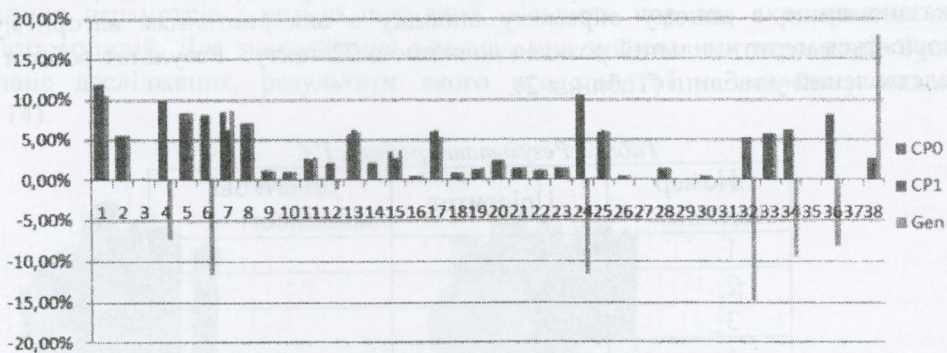


Рис. 16: Результати розрахунків трьох, що перевіряються алгоритмів стосовно "швидкого".

Ресурсів 1го типу 2, ресурсів 2го типу 4.

Як видно для зразка 4 генетичний алгоритм показує найгірший результат, але при збільшенні популяції до 1000 і кількості ітерацій до 500 значення порівнюються (рис. 16). Краще рішення був знайдене на 378 ітерації й надалі залишалося постійним при проведенні ще 2000 ітерацій.

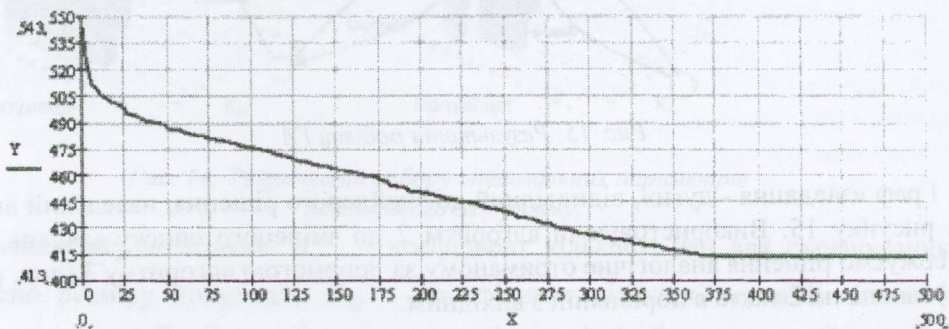


Рис. 17: Зразок 4, популяція 1000, найкращий результат 413.

Розрахунки, що дали схожі результати були проведені й для інших зразків.

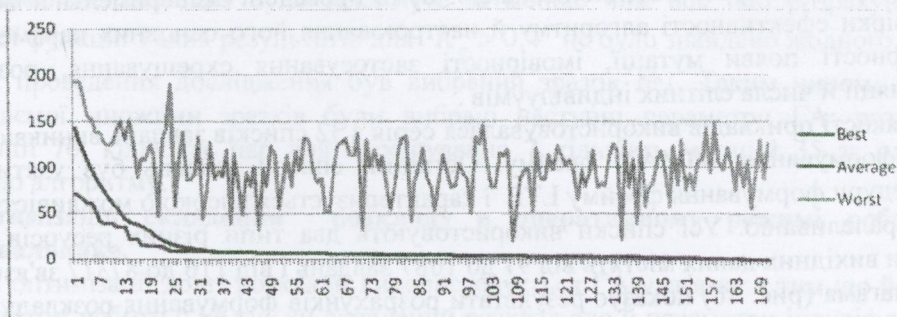


Рис. 18: Зразок 13, популяція 200, найкращий результат 173.4.

Для зразка 13 так само наведений гірший і середній результат по популяції на всіх етапах обчислень (рис.17).

Таким чином, можна зробити висновок про те, що серед чотирьох розглянутих алгоритмів генетичний дозволяє одержати найкращий результат при непорівнянному часі розрахунків, наприклад для зразка 4 рішення 413 було отримано за допомогою алгоритму «критичний шлях» менше ніж за секунду, знаходження цього ж результату за допомогою генетичного алгоритму зайняло більш 3х годин.

Обчислення проведені для кількості ресурсів першого типу 4 і другого типу 4 дозволяють зробити висновок про те, що при збільшенні кількості доступних ресурсів результати застосування різних алгоритмів розрахунків розклади прагнуть до значення критичного шляху.

Необхідно відмітити що особливо гарні результати навіть при невеликому числі ітерацій генетичний алгоритм показує на списках завдань із великою можливістю до распараллелюванню.

Висновки

1. Для вирішення завдання складання розкладу запропоновані й програмно реалізовані
 - Математичні моделі обчислювального процесу у вигляді графів «завдання - вершини» і «завдання - дуги»
 - Запропоновані методи аналізу й перетворення цих моделей, що дозволяє візуально оцінити стан обчислювального процесу.
 - Чотири типи алгоритмів, два придатні до реалізації в режимі реального часу й два більш ресурсномісткі для побудови субоптимального розкладу.
2. Реалізований генетичний алгоритм, що дозволяє не тільки скласти розклад але й зробити зміну розглянутого списку завдань, що дозволяє зменшити загальний час розкладу для алгоритмів працюючих в оперативному режимі.
3. Проведені експериментальні дослідження для більш ніж 30 варіантів черги завдань і різною кількістю доступних ресурсів. Виявлені параметри ГА які дають найкращий результат за мінімальний час для розглянутої множини завдань.

ЛИТЕРАТУРА

1. Э.Г. Коффман. Теория расписаний и вычислительные машины. Москва "Наука", 1984 г.
2. Mohammadi and S. Aki, "Scheduling Algorithms for Real-Time Systems", TR No.2005-499 Queen's University Publication. 2005.
3. Srinivasan, P. Holman, J. Anderson, and S. Baruah, "The Case for Fair Multiprocessor Scheduling", 2002.
4. Krishna and K. Shin, "Real Time Systems", McGraw-Hill, 1997.

5. И.В. Романовский. Дискретный анализ. Учебное пособие для студентов, специализирующихся по прикладной математике и информатике. СПб. "Невский диалект", 1999 г.
6. С.И. Зуховский, И.А. Радчик Математические методы сетевого планирования. Москва "Наука", 1965 г.
7. Phu D. Hoang and Jan M. Rabaey "Scheduling of DSP Programs onto Multiprocessors for Maximum Throughput" IEEE TRANSACTIONS ON SIGNAL PROCESSING, VOL. 41, NO. 6, JUNE 1993 p.2225
8. Robert W. Amphlett and David R. Bull "GENETIC ALGORITHM BASED DSP MULTIPROCESSOR SCHEDULING", Centre for Communications Research, University of Bristol, Bristol, UK.
9. Peter Koch, Niels Larsen, Thomas Bauer, Ole Ejlersen "GENETICAS: A Multi-DSP Scheduling Technique Based on Genetic Algorithms" .The DSP Research Group Institute for Electronic Systems, Aalborg University
10. Pavel Osmera, Bronislav Lacko, Masopust Petr "Parallel Evolutionary Algorithms", Pruceedinpr 2003 IEEE International Sympusium on Computational Intelligent in Robtics and Automatiun July 16-20,2003, Kobe, Japan
11. Shrutisagar C Sridhara, Krishnan N "Double Abstraction Level Heuristic Power Optimization for Digital Signal Processors Using GA's", Mepco Schienk Engg.

УДК 004.822

Систематизация информационных ресурсов Интернет-портала «Прочность материалов»

Л. С. Глоба, Р. Л. Новогрудская

Национальный технический университет Украины «КПИ»

В статье представлен подход к построению Интернет-портала знаний в области прочности материалов. Описан процесс построения модели представления знаний на портале. Приведена онтологическая модель портала, представлено формальное описание ее элементов.

Ключевые слова: Интернет-портал, знания, модель, онтология, предметная область

В статті наведено підхід до побудови Інтернет-порталу знань в області міцності матеріалів. Описано процес побудови моделі представлення знань на порталі. Наведена онтологічна модель порталу, представлено формальний опис її елементів.

Ключові слова: Інтернет-портал, знання, модель, онтологія, предметна область

The article presents the approach for the designing of knowledge Internet-portal in the field of materials strength. The process of portal knowledge representation model designing is described. Portal ontology model is shown, all its components formal description is given.

Key words: Internet-portal, knowledge, model, ontology, subject area.

Введение

В настоящее время существует огромное количество инженерного оборудования и программного обеспечения для решения специализированных задач разного рода. Однако они являются либо очень дорогими, либо очень закрытыми. Важные результаты, как теоретические, так и практические, которые получают исследователи, остаются сосредоточенными лишь на уровне конкретных исследовательских институтов. Происходит процесс сосредоточения знаний, они становятся доступными узкому кругу ученых и специалистов. Поэтому актуальной задачей становится предоставление возможности использования этих знаний большему количеству исследователей, для которых они предназначены.

Кроме того, недостатком процесса доступа к знаниям является плохая систематизированность и слабая структурированность больших объемов информации в сети Интернет, ее распределенность по разным Интернет сайтам, электронным библиотекам и архивам.

Из этого следует необходимость создания подхода направленного на построение специализированного Интернет портала знаний, ориентированного на работу с большим количеством разнообразных информационных и вычислительных ресурсов определенного направления, который не только даст возможность систематизировать и осуществлять поиск информации, но и реализовывать определенные расчетные задачи.

Для повышения эффективности доступа к инженерным знаниям за счет создания специализированного Интернет-портала инженерных знаний необходимо решить следующие задачи:

- Качественное представление знаний на портале
- Систематизация и структуризация информации
- Формализация инженерных знаний
- Эффективный содержательный поиск
- Описание сервисов, используемых для решения расчетных задач на портале.

Из всего вышесказанного следует необходимость построение качественной модели представления знаний на портале, которая позволит осуществить реализацию вышеперечисленных требований. В качестве модели представления знаний на портале использовались онтологии.

2. Онтология

Онтология портала представляет собой совокупность нескольких компонент, представленных на рис. 1. [1,2]

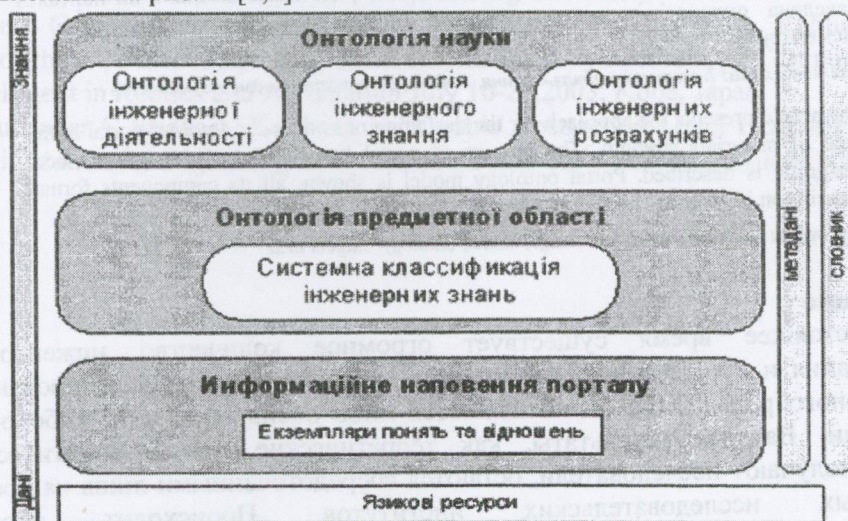


Рис. 1 Система онтологий портала.

2.1 Онтология науки

Она состоит из трех онтологий: Онтология инженерной деятельности, Онтология инженерного знания и Онтология инженерных расчетов $O_1 \supset \{O_2, O_3, O_4\}$.

2.2 Онтология инженерной деятельности:

Онтология инженерной деятельности (ОИД) включает десять общих классов понятий, относящиеся к организации научной деятельности, которые связаны отношениями различных типов. Различные свойства каждого понятия описываются на основе атрибутов понятий и ограничений, наложенных на область их значений. Исходя из этого ОИД может быть описана следующим образом:

$$O_2 = \{C_{o_2}, A_{o_2}, R_{o_2}, T_{o_2}, F_{o_2}, D_{o_2}\}, \quad \text{где}$$

$$C_{o_2} = \{C_1, C_2, C_3, C_4, C_5, C_6, C_7, C_8, C_9\},$$

$$A_{o_2} = (A_{c_1}, A_{c_2}, A_{c_3}, A_{c_4}, A_{c_5}, A_{c_6}, A_{c_7}, A_{c_8}, A_{c_9}), D_{o_2} = (D_1, D_2, D_3, D_4, D_5, D_6, D_7, D_8, D_9),$$

$R_{o_2} = (R_{AS_1}(O_2), \dots, R_{AS_{10}}(O_2), R_{IA_1}(O_2), \dots, R_{IA_6}(O_2), R_{n_1}(O_2), \dots, R_{n_6}(O_2), R_{CD_1}(O_2), \dots, R_{CD_6}(O_2)),$
 ОИД включает следующие классы понятий:

Персона - $C_1(O_2)$. К этому классу относятся понятия, связанные с субъектами научной деятельности: исследователями, сотрудниками и членами организаций. Атрибутами персоны являются: персональные данные, ученая степень, звание, направления научной деятельности, место проживания.
 $C_1(O_2) = (A_{C_1}, D_{C_1})$

Организация - C_2 . Понятия этого класса описывают различные организации, научные сообщества и ассоциации, институты, исследовательские группы и другие объединения. Атрибутами организации являются: название и место расположения.
 $C_2(O_2) = (A_{C_2}, D_{C_2})$

Мероприятие - C_3 . В этот класс входят понятия, описывающие научно-организационную или научно-исследовательскую деятельность – научные мероприятия, конференции, исследовательские поездки, проекты, программы и т.п. К атрибутам события относятся: название, место проведения, дата начала, дата окончания, степень завершенности.
 $C_3(O_2) = (A_{C_3}, D_{C_3})$

Деятельность - C_4 . Понятия класса Деятельность являются связующим звеном между методом и объектом исследования и полученным научным результатом. Класс описывает такие понятия, как проект, программа исследований.
 $C_4(O_2) = (A_{C_4}, D_{C_4})$

Публикация - C_5 . Этот класс служит для описания различного рода публикаций, (в периодических изданиях и тех, которые издаются в результате проведения конференций, научных мероприятий и т. д.). К атрибутам публикации относятся: название, описание, дата публикации и язык публикации.
 $C_5(O_2) = (A_{C_5}, D_{C_5})$

Местоположение - C_6 . Этот класс понятий позволяет описывать географическую и административно-территориальную локализацию объектов исследования, организаций и т.п. Атрибутами этого класса являются название местоположения и географический тип.
 $C_6(O_2) = (A_{C_6}, D_{C_6})$

Литература - C_7 . Этот класс служит для описания литературы (представленных в печатном или электронном форматах), которая используется в инженерной деятельности (монографии, статьи, отчеты, труды конференций, периодические издания, фото- и видеоматериалы и др.). К атрибутам публикации относятся: название, описание, дата публикации и язык публикации.
 $C_7(O_2) = (A_{C_7}, D_{C_7})$

Документация - C_8 . Этот класс служит для описания различного рода документации, такой как ГОСТЫ, ISO и т. д. К атрибутам документация относятся: название, описание, дата публикации и язык.
 $C_8(O_2) = (A_{C_8}, D_{C_8})$

Учебные материалы - C_9 . Этот класс содержит существующие учебники, справочники, учебные пособия, мануалы.... К атрибутам класса относятся: название, описание, дата публикации и язык. $C_9(O_2) = (A_{C_9}, D_{C_9})$

Для онтологии инженерной деятельности актуальны следующие отношения:

Ассоциативные отношения $R_{AS}(O_2) = \{C_i(O_2) \times C_j(O_2)\}$:

«*быть автором*» – используется для установления связи между персоной, являющейся автором публикации, и самой публикацией.

$$R_{AS_1} = \{C_1(O_2) \times C_5(O_2)\};$$

«*состоять в*» – связывает понятия организация и персона в случае, когда персона состоит в организации. $R_{AS_2} = \{C_1(O_2) \times C_2(O_2)\};$

«*быть участником*» – связывает мероприятие с персоной или организацией, участвующей в данном событии.

$$R_{AS_3} = \{(C_1(O_2) \times C_3(O_2)) \vee (C_2(O_2) \times C_3(O_2))\};$$

«*быть организатором*» – устанавливает связь между событием и персоной (или организацией), являющейся организатором события.

$$R_{AS_4} = \{(C_1(O_2) \times C_3(O_2)) \wedge (C_2(O_2) \times C_3(O_2))\};$$

«*научные труды*» – задает связь между мероприятием и публикациями, освещающими это мероприятие. $R_{AS_5} = \{C_3(O_2) \times C_5(O_2)\};$

«*Быть изданным в*» – связывает публикацию и организацию, являющуюся издателем. $R_{AS_6} = \{C_2(O_2) \times C_5(O_2)\};$

«*Располагаться*» - описывает место расположения организации.

$$R_{AS_7} = \{C_2(O_2) \times C_6(O_2)\}$$

«*описывает*» – связывает публикацию с любым понятием ОИЗ.

$$R_{AS_8} = \{(C_5(O_2) \times C_1(O_3)) \vee (C_5(O_2) \times C_2(O_3)) \vee (C_5(O_2) \times C_3(O_3)) \vee (C_5(O_2) \times C_4(O_3)) \vee (C_5(O_2) \times C_5(O_3))\}$$

«*часть деятельности*» – связывает деятельность с объектами, методами и результатами исследования

$$R_{AS_{10}} = \{(C_4(O_2) \times C_2(O_3)) \vee (C_4(O_2) \times C_1(O_3)) \vee (C_4(O_2) \times C_3(O_3))\};$$

«*применяется к*» - связывает метод и тип объектов исследования, к которым он применяется. $R_{AS_3} = \{C_1(O_3) \times C_2(O_3)\}.$

Структурные отношения («*общее-частное*», «*часть-целое*») $R_{IA}(O_2) = C_k(O_2) \subset C_m(O_2):$

Отношением «*часть-целое*» описывается связь между классом Литература и классами Документация, Учебные материалы и Публикация. Последние являются подклассом класса литература. $C_7 \subset C_8 \cup C_9 \cup C_5;$

«*общее-частное*» актуально для класса Персона и относящихся к нему Исследователей, Сотрудников и Членов организаций. $C_1 \subset C_{11} \wedge C_{12} \wedge C_{13};$

класс Организация является *общим* для Организаций, Научных сообществ и Ассоциаций, Институтов, Исследовательских групп и других объединений.

$$C_2 \subset C_{2_1} \wedge C_{2_2} \wedge C_{2_3} \wedge C_{2_4} \wedge C_{2_5} \wedge C_{2_6};$$

отношением «*общее-частное*» связывается класс Мероприятие с Конференциями, Исследовательскими Поездками, Проектами, программами.

$$C_3 \subset C_{3_1} \wedge C_{3_2} \wedge C_{3_3} \wedge C_{3_4};$$

связь класса Документация с документами вида ГОСТЫ, ISO и др. осуществляется с помощью отношения «*общее-частное*».

$$C_8 \subset C_{8_1} \wedge C_{8_2} \wedge C_{8_3};$$

отношение «*общее-частное*» описывает связь класса Учебные материалы с Учебниками, Справочниками, Учебными Пособиями, Мануалами.

$$C_9 \subset C_{9_1} \wedge C_{9_2} \wedge C_{9_3} \wedge C_{9_4}.$$

Отношение

наследования $R_n(O_2) = a_i, r_i \mid A_{C_m}(O_2) \rightarrow a_i, r_i \mid A_{C_k}(O_2)$:

Отношение наследования используется для передачи атрибутов и отношений от родительского класса к дочернему. Отношение используется в описанном выше смысле для:

Класс Литература его подклассы Документация, Учебные материалы и Публикация: $A(C_7), R(C_7) \rightarrow A(C_5), R(C_5)$.

$$A(C_7), R(C_7) \rightarrow A(C_8), R(C_8). A(C_7), R(C_7) \rightarrow A(C_9), R(C_9).$$

Класс Персона его подклассы Исследователей, Сотрудников и Членов организаций:

$$A(C_1), R(C_1) \rightarrow A(C_{1_1}), R(C_{1_1}),$$

$$A(C_1), R(C_1) \rightarrow A(C_{1_2}), R(C_{1_2}). A(C_1), R(C_1) \rightarrow A(C_{1_3}), R(C_{1_3}).$$

Класс Организация его подклассы Научных сообществ и Ассоциаций, Институтов, Исследовательских групп:

$$A(C_2), R(C_2) \rightarrow A(C_{2_1}), R(C_{2_1}),$$

$$A(C_2), R(C_2) \rightarrow A(C_{2_2}), R(C_{2_2}).$$

$$A(C_2), R(C_2) \rightarrow A(C_{2_3}), R(C_{2_3}).$$

$$A(C_2), R(C_2) \rightarrow A(C_{2_4}), R(C_{2_4}).$$

Класс Мероприятие его подклассы Конференциями, Исследовательскими Поездками, Проектами, программами:

$$A(C_3), R(C_3) \rightarrow A(C_{3_1}), R(C_{3_1}).$$

$$A(C_3), R(C_3) \rightarrow A(C_{3_2}), R(C_{3_2}).$$

$$A(C_3), R(C_3) \rightarrow A(C_{3_3}), R(C_{3_3}).$$

$$A(C_3), R(C_3) \rightarrow A(C_{3_4}), R(C_{3_4}).$$

Класс Учебные материалы его подклассы Справочниками, Учебными Пособиями,

$$\text{Мануалами: } A(C_9), R(C_9) \rightarrow A(C_{9_1}), R(C_{9_1}).$$

$$A(C_9), R(C_9) \rightarrow A(C_{9_2}), R(C_{9_2}). A(C_9), R(C_9) \rightarrow A(C_{9_3}), R(C_{9_3}).$$

Отношения «класс-данные» $R_{CD}(O_2) = C_j(O_2) \subseteq D_i(O_2)$:

Отношения вида «класс-данные» используется для связки конкретных экземпляров понятий с классом. Описанное отношение актуально для всех классов данной онтологии: $C_1(O_2) \subseteq D$, $A_{C_1} \subseteq A_D$, $C_2(O_2) \subseteq D$, $A_{C_2} \subseteq A_D$, $C_3(O_2) \subseteq D$, $A_{C_3} \subseteq A_D$, $C_4(O_2) \subseteq D$, $A_{C_4} \subseteq A_D$, $C_5(O_2) \subseteq D$, $A_{C_5} \subseteq A_D$, $C_6(O_2) \subseteq D$, $A_{C_6} \subseteq A_D$, $C_7(O_2) \subseteq D$, $A_{C_7} \subseteq A_D$, $C_8(O_2) \subseteq D$, $A_{C_8} \subseteq A_D$, $C_9(O_2) \subseteq D$, $A_{C_9} \subseteq A_D$, $C_{10}(O_2) \subseteq D$, $A_{C_{10}} \subseteq A_D$.

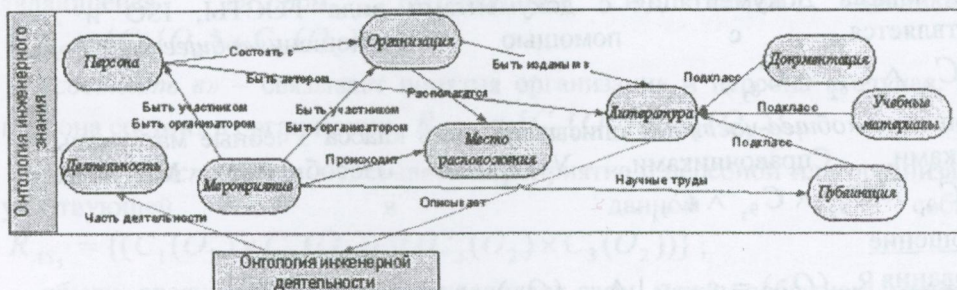


Рис 2 Классы и связи ОИЗ

2.3 Онтология инженерного знания:

Онтология инженерного знания (ОИЗ) состоит из пяти классов понятий, данные классы являются метапонятиями, задающими структуры для описания рассматриваемой предметной области. Существуют связи между классами заданные отношениями различных типов. Исходя из этого ОИЗ может быть описана следующим образом: $O_3 = \{C_{O_3}, A_{O_3}, R_{O_3}, T_{O_3}, F_{O_3}, D_{O_3}\}$,

где $C_{O_3} = \{C_1, C_2, C_3, C_4, C_5\}$, $A_{O_3} = (A_{C_1}, A_{C_2}, A_{C_3}, A_{C_4}, A_{C_5})$, $R_{O_3} = (R_{AS_1}(O_3), \dots, R_{AS_6}(O_3), R_{IA_1}(O_3), R_{\eta_1}(O_3), R_{CD_1}(O_3), \dots, R_{CD_5}(O_3))$,

$D_{O_3} = (D_{C_1}, D_{C_2}, D_{C_3}, D_{C_4}, D_{C_5})$. ОИЗ содержит следующие метапонятия:

Метод исследования - $C_1(O_3)$. Данный класс служит для описания методов исследования, применяемых в сопротивлении материалов. К атрибутам класса относятся название метода, его автор. $C_1(O_3) = (A_{C_1}, D_{C_1})$

Объект исследования - $C_2(O_3)$. Понятия этого класса задают типизацию объектов исследования и структуры для их описания. Объектами исследования могут выступать как материал, или группа материалов, так и различные свойства материалов. К атрибутам класса относятся название объекта, его параметры (характеристики). $C_2(O_3) = (A_{C_2}, D_{C_2})$

Результат исследования - $C_3(O_3)$. Понятия этого класса служат для описания результатов научной деятельности и их типизации. Класс результат содержит результаты уже полученные учеными. Научные результаты обычно находят свое отражение в публикациях. К атрибутам класса относится название, кем он был получен, требуемые характеристики. $C_3(O_3) = (A_{C_3}, D_{C_3})$

Назначение исследования - $C_4(O_3)$. Понятия этого класса отражают цель исследования и описывают "на что" проводится исследование (на сопротивление, на устойчивость к деформации, на устойчивость к нагрузкам). К атрибутам класса относится направление исследования. $C_4(O_3) = (A_{C_4}, D_{C_4})$

Исследовательское оборудование - $C_5(O_3)$. Понятия этого класса описывают оборудование, на котором проводятся исследования. К атрибутам класса относятся тип оборудования, название, год выпуска, размеры, некоторые нужные для проведения эксперимента параметры и характеристики. $C_5(O_3) = (A_{C_5}, D_{C_5})$

На классах данной онтологии реализованы следующие отношения:

Ассоциативные отношения $R_{AS}(O_3) = \{C_i(O_3) \times C_j(O_3)\}$:

«применяется к» - связывает метод и тип объектов исследования, к которым он применяется. $R_{AS_3} = \{C_1(O_3) \times C_2(O_3)\}$.

«проводится на» - связывает метод исследования с исследовательским оборудованием. $R_{AS_4} = \{(C_1(O_3) \times C_5(O_3))\}$.

«направлено на» - связывает назначение исследования с методом исследования. $R_{AS_5} = \{C_4(O_3) \times C_1(O_3)\}$.

«получен на» - связывает результат исследования с исследовательским оборудованием, на котором он был получен. $R_{AS_6} = \{C_3(O_3) \times C_5(O_3)\}$.

«проводимые с» - связывает результат исследования с объектом исследования. $R_{AS_1} = \{C_3(O_3) \times C_2(O_3)\}$.

Структурные отношения («общее-частное», «часть-целое») $R_{IA}(O_3) = C_k(O_3) \subset C_m(O_3)$:

Отношением «общее-частное» описывается связь между классом *Исследовательское оборудование* и установки для проведения опытов разного типа, измерительное оборудование, криогенное оборудование, тепловые установки. $C_5 \subset C_{5_1} \wedge C_{5_2} \wedge C_{5_3} \wedge C_{5_4}$.

Отношение наследования $R_n(O_3) = a_i, r_i | A_{C_m}(O_3) \rightarrow a_i, r_i | A_{C_k}(O_3)$:

Отношение наследования используется для передачи атрибутов и отношений от родительского класса к дочернему. Отношение используется в описанном выше смысле для:

Класс *Исследовательское оборудование* его подклассы установки для проведения опытов разного типа, измерительное оборудование, криогенное оборудование, тепловые установки. $A(C_5), R(C_5) \rightarrow A(C_{5_1}), R(C_{5_1}), A(C_5), R(C_5) \rightarrow A(C_{5_2}), R(C_{5_2}), A(C_5), R(C_5) \rightarrow A(C_{5_3}), R(C_{5_3}), A(C_5), R(C_5) \rightarrow A(C_{5_4}), R(C_{5_4})$.

Отношения «класс-данные» $R_{CD}(O_3) = C_j(O_3) \subseteq D_i(O_3)$:

Отношения вида «класс-данные» используется для связки конкретных экземпляров понятий с классом. Описанное отношение актуально для всех классов данной онтологии: $C_1(O_3) \subseteq D$, $A_{C_1} \subseteq A_D$, $C_2(O_3) \subseteq D$, $A_{C_2} \subseteq A_D$, $C_3(O_3) \subseteq D$, $A_{C_3} \subseteq A_D$, $C_4(O_2) \subseteq D$, $A_{C_4} \subseteq A_D$, $C_5(O_3) \subseteq D$, $A_{C_5} \subseteq A_D$.

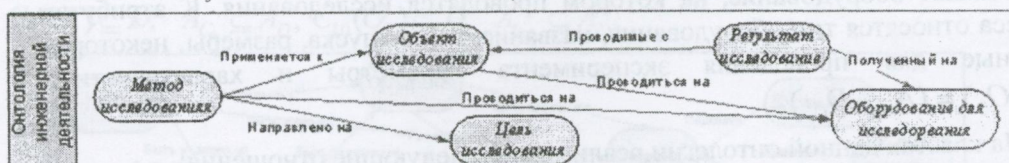


Рис 3 Классы и связи ОИЗ

2.4 Онтология инженерных расчетов

Онтология инженерных расчетов (ОИР) объединяет пять классов понятий, которые описывают расчетные возможности, реализованные на портале. Свойства и характеристики каждого понятия описываются с помощью атрибутов понятий и ограничений, наложенных на область значений. Исходя из этого ОИД может быть описана следующим образом: $O_4 = \{C_{o_4}, A_{o_4}, R_{o_4}, T_{o_4}, F_{o_4}, D_{o_4}\}$, где $C_{o_4} = \{C_1, C_2, C_3, C_4, C_5\}$, $A_{o_4} = (A_{C_1}, A_{C_2}, A_{C_3}, A_{C_4}, A_{C_5})$, $D_{o_4} = (D_{C_1}, D_{C_2}, D_{C_3}, D_{C_4}, D_{C_5})$, $R_{o_4} = (R_{AS_1}(O_4), \dots, R_{AS_7}(O_4), R_{IA_1}(O_4), R_{n_1}(O_4), R_{CD_1}(O_4), \dots, R_{CD_5}(O_4))$.

Онтология инженерных расчетов состоит следующие классы понятий:

Расчет (*Расчет_1, Расчет_2, Расчет_2*) – $C_1(O_4)$. К этому классу относятся типы и методы расчетов, которые используются для проведения исследование (эксперимент) на портале. К атрибутам класса относятся название, направление и время расчета. $C_1(O_4) = (A_{C_1}, D_{C_1})$

Результат – $C_2(O_4)$. В этот класс входят результаты расчетов. (Результаты, которые может получить пользователь, проведя исследование (эксперимент) на портале). Результаты могут быть изображены (представлены пользователю) в виде графиков, текстовых описаний, формул. Класс, как бы аналог класса “научный результат”, из онтологии инженерного знания, для онтологии инженерных расчетов. К атрибутам класса относятся название, описание вида представления. $C_2(O_4) = (A_{C_2}, D_{C_2})$

Сервис – $C_3(O_4)$. Этот класс описывает набор сервисов, которые используются для проведения расчетов. К атрибутам класса относятся название и характеристики сервиса. $C_3(O_4) = (A_{C_3}, D_{C_3})$

Интерфейс – $C_4(O_4)$. К этому классу относятся понятия, задействованные в реализации визуализации сервисов и расчетных задач. К атрибутам относится название интерфейса, его тип и вид. $C_4(O_4) = (A_{C_4}, D_{C_4})$

Параметры (для расчета) – $C_5(O_4)$. Этот класс содержит параметры, при которых совершается расчет (температура, жесткость и т.д.). К атрибутам класса относится название параметра, его численная характеристика.

$$C_1(O_4) = (A_{C_5}, D_{C_{51}})$$

Классы онтологии инженерных расчетов связаны следующими отношениями:

Ассоциативные отношения $R_{AS}(O_4) = \{C_i(O_4) \times C_j(O_4)\}$:

«используется для» - связывает сервис с расчетом.

$$R_{AS_1} = \{C_3(O_4) \times C_1(O_4)\}.$$

«находится в процессе» - связывает объект исследования из ОИЗ с расчетом.

$$R_{AS_2} = \{C_2(O_3) \times C_1(O_4)\}.$$

«быть результатом» - связывает результат и расчет.

$$R_{AS_3} = \{C_2(O_4) \times C_1(O_4)\}.$$

«быть полученным при» - связывает результат и параметры расчета.

$$R_{AS_4} = \{C_2(O_4) \times C_5(O_4)\}.$$

«служить параметрами» - связывает класс параметры и класс расчет.

$$R_{AS_5} = \{C_5(O_4) \times C_1(O_4)\}.$$

«быть полученным с помощью» - связывает класс результат с классом сервис.

$$R_{AS_6} = \{C_2(O_4) \times C_3(O_4)\}.$$

«использоваться для» - связывает класс объект исследования из ОИЗ с классом расчет

$$R_{AS_7} = \{C_1(O_4) \times C_2(O_3)\}$$

Структурные отношения («общее-частное», «часть-целое») $R_{IA}(O_4) = C_k(O_4) \subset C_m(O_4)$:

Отношением «общее-частное» связан класс Расчет с классами Расчет_1 (Расчет предельных состояний), Расчет_2 (Расчет прочностных характеристик), Расчет_3 (Расчет деформации). $C_1 \subset C_{1_1} \wedge C_{1_2} \wedge C_{1_3}$.

Отношение

наследования $R_n(O_4) = a_i, r_i \mid A_{C_m}(O_4) \rightarrow a_i, r_i \mid A_{C_k}(O_4)$:

Отношение наследования используется для передачи атрибутов и отношений от родительского класса к дочернему. Отношение используется в описанном выше смысле для:

Класс *Расчет* его подклассы Расчет_1 (Расчет предельных состояний), Расчет_2 (Расчет прочностных характеристик), Расчет_3 (Расчет деформации). $A(C_1), R(C_1) \rightarrow A(C_{1_1}), R(C_{1_1}), A(C_1), R(C_1) \rightarrow A(C_{1_2}), R(C_{1_2}), A(C_1), R(C_1) \rightarrow A(C_{1_3}), R(C_{1_3})$.

Отношения «класс-данные» $R_{CD}(O_4) = C_j(O_4) \subseteq D_i(O_4)$:

Отношения вида «класс-данные» используется для связки конкретных экземпляров понятий с классом. Описанное отношение актуально для всех

классов данной онтологии: $C_1(O_4) \subseteq D$, $A_{C_1} \subseteq A_D$, $C_2(O_4) \subseteq D$, $A_{C_2} \subseteq A_D$, $C_3(O_4) \subseteq D$, $A_{C_3} \subseteq A_D$, $C_4(O_4) \subseteq D$, $A_{C_4} \subseteq A_D$, $C_5(O_5) \subseteq D$, $A_{C_5} \subseteq A_D$.

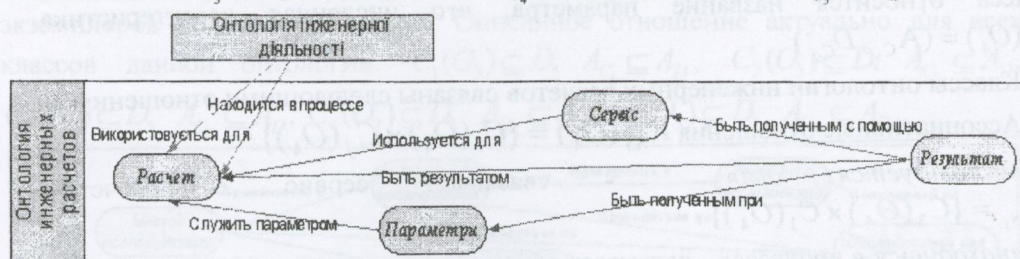


Рис 4 Классы и связи ОИР

3. Онтология предметной области

Онтология предметной области отражает общие знания о предметной области, такие как иерархия классов понятий, семантические отношения на этих классах. Основой онтологии предметной области для портала инженерных знаний в области сопротивления материалов послужила системная классификация[5].

Онтология предметной области (ОПО) описывает сопротивление материалов в целом как раздел науки и включает формальное и неформальное описание понятий и отношений между ними. Эти понятия являются реализациями метапонятий онтологии инженерного знания и могут быть упорядочены в иерархию общее-частное и часть-целое. Так, например, *Объектам* исследования в науке сопротивление материалов соответствуют материалы или определенные свойства материалов (Рис 3), а в качестве *Методов* исследования выступают такие понятия, как методы деформаций, методика перемещений, метод распределения напряжений и др. Основному классу онтологии инженерной деятельности *Расчет* в сопротивлении материалов соответствуют расчет предельных состояний, расчет деформации, расчеты на прочность и др.[4].

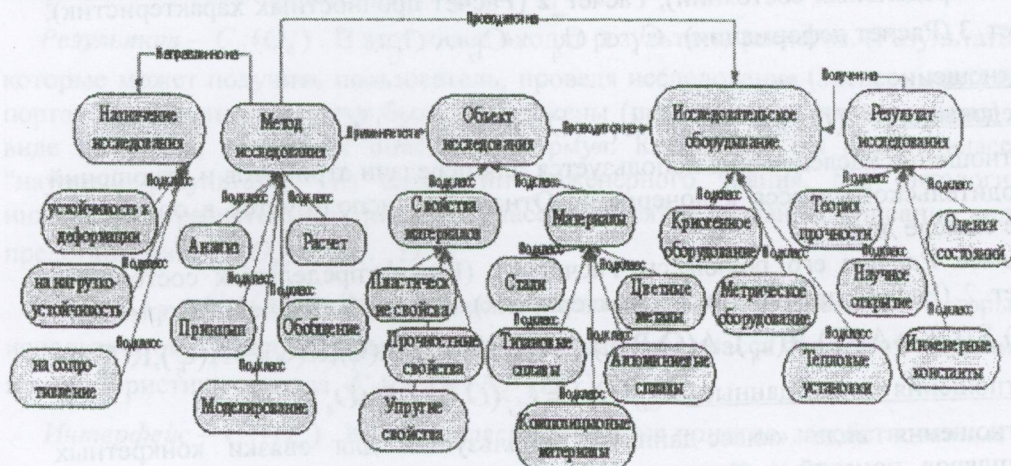


Рис 5 Фрагмент модели представления знаний (ОПО)

4 Информационное наполнение портала и языковые ресурсы

Экземпляры классов понятий и отношений, определенных в онтологии портала, образуют его *информационное наполнение*.

Исходными данными для системы знаний, характеризующими предметную область, являются *языковые ресурсы*, представленные в виде коллекции документов. Обеспечить автоматическое извлечение знаний из этих данных является главной задачей эксперта при наполнении и настройке системы знаний портала.

Выводы.

Рассмотрена возможность построения портала “Прочность материалов”. Информационную основу портала составляют онтологии, с помощью которой осуществляется систематизация и структуризация информации, организовывается эффективный поиск и навигация по информационному пространству портала инженерных знаний.

ЛИТЕРАТУРА

1. Загоруйко Ю. А., Попов И. Г. Описание сложных предметных областей на основе интеграции средств представления знаний // Тр. междунар. семинара «Диалог'97» по компьютерной лингвистике и ее приложениям. М.: 1997. С. 110–115.
2. Глоба Л. С., Новогрудская Р.Л. Подход к построению портала инженерных знаний // Труды конференции «Интеллектуальный анализ информации ИАИ - 2010». – Киев: КПИ; 2010. – С53-62.
3. Прочность материалов и конструкций / Редкол.: В. Т. Трощенко (отв. ред.) и др. – К.: Академперіодика, 2005. – 1088с.
4. А. С. Зубченко Марочник сталей и сплавов. – М. - Машиностроение, 2001. – 663с.
5. Боровикова О. И., Загоруйко Ю. А. Организация порталов знаний на основе онтологий // Тр. междунар. семинара «Диалог 2002» Компьютерная лингвистика и интеллектуальные технологии (Протвино, 6–11 июня 2002 г.). М.: Наука, 2002. Т. 2. С. 76–82.

УДК 004.655

До питання про створення мов запитів: узагальнена таблична алгебра, узагальнені реляційні числення

І. М. Глушко

Київський національний університет імені Тараса Шевченка, Україна

В статті проведено узагальнення табличної алгебри, побудованої на основі реляційних алгебр Кодда; узагальнення полягає в тому, що замість таблиць розглядаються пари, які складаються з таблиць та схем таблиць. Числення рядків (кортежів) та числення на домені поповнені довільними предикатними та функціональними сигнатурами. Доведено, що узагальнена таблична алгебра, узагальнене числення рядків та узагальнене числення на домені еквівалентні.

Ключові слова: реляційні (табличні) бази даних, числення рядків, числення на домені, таблична алгебра.

В статье проведено обобщение табличной алгебры, построенной на основе реляционных алгебр Кодда; обобщение состоит в том, что вместо таблиц рассматриваются пары, состоящие из таблиц и схем таблиц. Исчисление строк (кортежей) и исчисление на домене пополнены произвольными предикатными и функциональными сигнатурами. Доказано, что обобщенная табличная алгебра, обобщенное исчисление строк и обобщенное исчисление на домене эквивалентны.

Ключевые слова: реляционные (табличные) базы данных, исчисление строк, исчисление на домене, табличная алгебра.

In the paper generalization of table algebra built on the basis of Codd's relational algebras is conducted. Generalization is that instead of tables we consider pairs consisting of tables and scheme of tables. The classic tuple calculus and domain calculus are filled up arbitrary predicate and functional signatures on the universal domain. It is proved the equivalence between table algebra, tuple calculus and domain calculus.

Key words: relation databases, tuple calculus, domain calculus, table algebra.

1. Вступ

Уточнення реляції в термінах іменних множин було здійснене у монографії [1]. У даній роботі отримані результати узагальнено на таблиці, яким приписується певна схема. Класичне числення рядків та числення на домені поповнені довільними предикатними та функціональними сигнатурами на універсальному домені (у той час як, зазвичай, розглядають лише бінарні предикати, а функціональна сигнатура взагалі порожня, див., наприклад, [2]). Доведено, що узагальнена таблична алгебра, узагальнене числення рядків та узагальнене числення на домені еквівалентні.

2. Узагальнення табличної алгебри

Всі невизначені тут поняття та позначення розуміємо в смислі монографії [1]. Розглядаємо дві множини: A – множину атрибутів і D – універсальний домен. Довільну (скінченну) множину атрибутів $R \subseteq A$ назовемо схемою. Рядком схеми R називається іменна множина на парі R, D , проекція якої за першою компонентою рівна R , тобто рядок схеми R – це функція вигляду $s : R \rightarrow D$.

Під (узагальненою) таблицею розуміємо пару $\langle t, R \rangle$, де $t \in T(R)$ – таблиця фіксованої схеми R . Тоді $T(R) = \{\langle t, R \rangle \mid t \in T(R)\}$ – множина усіх (узагальнених) таблиць схеми R , а $T = \bigcup_R T(R)$ – множина всіх таблиць. Отже,

кожній таблиці приписується певна схема на відміну від визначення таблиці, яке подане у [1]. Це впливає тільки на випадок порожньої таблиці $t_\emptyset$, оскільки за непорожньою таблицею схема відновлюється однозначно. Запис $\langle t_\emptyset, R \rangle$ позначає порожню таблицю схеми R .

Під узагальненою табличною алгеброю розуміємо алгебру $\langle T, \Omega_{P, \Xi} \rangle$, де T – множина усіх таблиць, $\Omega_{P, \Xi} = \{\cup_R, \cap_R, \setminus_R, \sigma_{p, R}, \pi_{X, R}, \otimes_{R_1, R_2}^{R_1}, \div_{R_2}^{R_1}, R t_{\xi, R}, \sim_R\}$ – сигнатура, $p \in P, \xi \in \Xi$, $X, R, R_1, R_2 \subseteq A$, P, Ξ – множини параметрів. Операції сигнатури $\Omega_{P, \Xi}$ задано в [3]. Виразом табличної алгебри називається вираз, побудований з таблиць множини T при використанні операцій з сигнатури $\Omega_{P, \Xi}$.

Лема 1. Будь-який вираз табличної алгебри можна замінити еквівалентним йому виразом, який використовує сталі таблиці з єдиним атрибутом і єдиним рядком, операції селекції, з'єднання, проекції, об'єднання, різниці й перейменування

(доведення див., наприклад, в [2]).

3. Узагальнене числення рядків

Розглянемо множину атрибутів A і універсальний домен D ; множини предметних змінних (змінних рядків) $x_1, x_2, \dots$; множини предметних констант $d_1, d_2, \dots$; множини функціональних символів $f_1^{n_1}, f_2^{n_2}, \dots$; множини предикатних символів $p_1^{m_1}, p_2^{m_2}, \dots$. Областю інтерпретації предметних констант є універсальний домен D , предметних змінних – множина всіх рядків.

Введемо синтаксичні змінні як у [4]. Використовуємо x як синтаксичну змінну, областю зміни якої є предметні змінні; $f(p)$ як синтаксичну змінну, областю зміни якої є функціональні (відповідно предикатні) символи; d як синтаксичну змінну, областю зміни якої є предметні константи; A як синтаксичну змінну, областю зміни якої є атрибути.

Наступні вирази є термами (індукція за довжиною термів):

- d – терм;
- $x(A)$ – терм;
- якщо $u_1, \dots, u_n$ – терми, f – n -арний функціональний символ, то $f(u_1, \dots, u_n)$ – терм;
- вираз є термом, тільки тоді, коли це слідує з правил а), б), с).

Будемо використовувати u як синтаксичну змінну, областю зміни якої є терми. Сформулюємо правила побудови формул. Атомарні формули (атоми) можуть бути двох типів.

- a1. Нехай $\langle t, R \rangle$ – таблиця, а x – змінний рядок. Тоді $t(x)$ – атом, який (при інтерпретації) означає, що $x \in t$.
- a2. Нехай $u_1, \dots, u_m$ – терми, а p – m -арний предикат на універсальному домені D . Тоді $p(u_1, \dots, u_m)$ – атом.

Використаємо логічні зв'язки $\neg$, $\wedge$, $\vee$, квантори $\exists, \forall$ та дужки $(,)$ для побудови формул із атомів. Будемо використовувати P , Q та G як синтаксичні змінні, областю зміни яких є формули, визначені нижче.

- f1. Кожний атом – це формула.
- f2. Якщо P – формула, то $\neg P$ – теж формула.
- f3. Якщо P і Q – формули, то $(P \wedge Q)$, $(P \vee Q)$ – формули.
- f4. Нехай x – змінний рядок, P – формула, $R \subseteq A$ – схема, тоді $\exists x(R)P$ – формула.
- f5. Нехай x – змінний рядок, P – формула, $R \subseteq A$ – схема, тоді $\forall x(R)P$ – формула.
- f6. Якщо P – формула, то (P) – теж формула.
- f7. Інших формул немає.

В загальному випадку змінні рядки у формулах можуть бути вільні або зв'язані. Смысл цих понять такий самий як і у численні предикатів (див., наприклад, [4]). Використовуючи поняття вільних та зв'язаних змінних рядків, схеми $scheme(x, P)$ (скінченної множини атрибутів) та множини атрибутів $attr(x, P)$, з якими рядок x зустрічається у формулах, виділено клас дозволених формул. Точні означення див. в [3].

Вираз числення рядків має вигляд $\{x(R) \mid P(x)\}$, де

1. формула P – дозволена;
2. змінна x – єдина змінна, яка входить у формулу P вільно;
3. якщо $scheme(x, P)$ визначена, то $scheme(x, P) = R$, інакше, $attr(x, P) \subseteq R$.

Нехай $P(x)$ – дозволена формула, $R \subseteq A$. В результаті підстановки конкретного рядка s схеми R замість x у P отримаємо формулу $P(s/x)$.

Нехай $E = \{x(R) \mid P(x)\}$ – вираз числення рядків. Значенням виразу E назвемо таблицю схеми R , яка містить всі рядки $s \in S(R)$, такі, що формула $P(s/x)$ істинна.

4. Реляційне числення зі змінними на домені

Числення на домені відрізняється від числення рядків тим, що замість змінних рядків розглядаються змінні значення із домену, які представляють компоненти рядка. Також числення на домені підтримує умову приналежності (membership condition) (див., наприклад, в [5]).

Розглянемо множину атрибутів A і універсальний домен D ; множини предметних констант $d_1, d_2, \dots$; множини предметних змінних $x_1, x_2, \dots$; множини функціональних символів $f_1^{n_1}, f_2^{n_2}, \dots$; множини предикатних символів $p_1^{m_1}, p_2^{m_2}, \dots$. Областю інтерпретації предметних констант та предметних змінних є універсальний домен D . Як і раніше, використовуємо синтаксичні змінні x , $f(p)$, d , A , u , P , Q та G , областю зміни яких є предметні змінні, функціональні (відповідно предикатні) символи, константи, атрибути, терми та формули відповідно.

Наступні вирази є термами (індукція за довжиною термів):

- a) будь-яка предметна константа або предметна змінна є термом;
- b) якщо $u_1, \dots, u_n$ – терми і f – n -арний функціональний символ, то $f(u_1, \dots, u_n)$ – терм;
- c) вираз є термом, тільки тоді, коли це слідує з правил a), b).

Атомарні формули (атоми) мають вигляд:

- a1. нехай $\langle t, R \rangle$ – таблиця схеми $R = \{A_1, \dots, A_n\} \subseteq A$. Тоді $t(a_1, \dots, a_n)$ – атом, де a_i – константа або змінна на універсальному домені D ;
- a2. нехай $u_1, \dots, u_m$ – терми, а p – m -арний предикат на універсальному домені D . Тоді $p(u_1, \dots, u_m)$ – атом.

Для побудови формул із атомів використаємо логічні зв'язки $\neg$, $\wedge$, $\vee$, квантори $\exists, \forall$ та дужки $(,)$.

- f1. Кожний атом – це формула.
- f2. Якщо P – формула, то $\neg P$ – теж формула.
- f3. Якщо P і Q – формули, то $(P \wedge Q)$, $(P \vee Q)$ – формули.
- f4. Нехай x – змінна, P – формула, A – атрибут з множини A , тоді $\exists x (A) P$ – формула.
- f5. Нехай x – змінна, P – формула, A – атрибут з множини A , тоді $\forall x (A) P$ – формула.
- f6. Якщо P – формула, то і (P) – формула.
- f7. Інших формул немає.

Поняття вільних і зв'язаних змінних на домені визначаються в численні на домені так само, як і в численні рядків. Схема змінної x у формулі P , $scheme(x, P)$ – це або універсальний домен D , або невизначена величина. Поняття дозволеної формули визначається як і в численні рядків, а саме: формула дозволена, якщо змінна, зв'язана квантором, входить вільно у формулу, яка слідує за квантором.

Вираз числення зі змінними на домені має вигляд $\{x_1, \dots, x_n \mid P(x_1, \dots, x_n)\}$, де

1. формула P – дозволена, а $x_1, \dots, x_n$ – всі вільні змінні, які входять у формулу P ;
2. $R = \{A_1, \dots, A_n\}$, $R \subseteq A$ – схема, порядок атрибутів фіксований;
3. $scheme(x_i, P) = D$, $i = 1, \dots, n$.

Нехай $P(x)$ – дозволена формула. У результаті підстановки константи d з універсального домену D замість змінної x у P отримаємо формулу $P(d/x)$.

Нехай $E = \{x_1, \dots, x_n \mid P(x_1, \dots, x_n)\}$ – вираз числення на домені. Значенням виразу E назвемо таблицю схеми $R = \{A_1, \dots, A_n\}$, яка містить всі рядки $s \in S(R)$, $s = \langle A_1, d_1 \rangle, \dots, \langle A_n, d_n \rangle$ такі, що формула $P(d_1/x_1, \dots, d_n/x_n)$ істинна.

5. Еквівалентність узагальненої табличної алгебри, узагальненого числення рядків та узагальненого числення на домені

Покажемо, що для довільного виразу табличної алгебри можна побудувати еквівалентний вираз числення рядків. Вираз числення рядків E еквівалентний виразу табличної алгебри F , якщо значення виразу E співпадає із значенням виразу F , за умови, коли табличні константи і табличні змінні інтерпретуються однаково в цих двох виразах.

Теорема 1. Якщо F – вираз табличної алгебри, то можна ефективно (і рівномірно) побудувати еквівалентний йому вираз E числення рядків.

Доведення.

Згідно леми 1 розглядаємо вирази табличної алгебри, які містять лише операції об'єднання, різниці, селекції, проєкції, з'єднання та перейменування, причому допускаються тільки одноатрибутні сталі таблиці з одним рядком. Доведення проведемо індукцією за числом операцій в F .

База індукції (немає операцій). Можливі два випадки. По-перше, $F = t$, де t – таблиця схеми R , покладемо $E = \{x(R) \mid t(x)\}$. По-друге, F – стала таблиця $t = \{\langle A, d \rangle\}$. Покладемо $E = \{x(\{A\}) \mid x(A) = d\}$. □

Крок індукції. Припустимо, що твердження теореми виконується для будь-якого виразу табличної алгебри, який містить менше $k = 1, 2, 3, \dots$ операцій. Нехай вираз F містить k операцій.

Випадок 1. $F = F_1 \cup_R F_2$. Тоді F_1 і F_2 мають менше k операцій, тому існують вирази числення рядків $\{x(R) \mid P(x)\}$ і $\{x(R) \mid Q(x)\}$, еквівалентні F_1 і F_2 відповідно. Покладемо E рівним $\{x(R) \mid P(x) \vee Q(x)\}$. □

Випадок 2. $F = F_1 \setminus_R F_2$. Нехай $\{x(R) \mid P(x)\}$ і $\{x(R) \mid Q(x)\}$ – вирази числення рядків, еквівалентні F_1 і F_2 відповідно. Покладемо $E = \{x(R) \mid P(x) \wedge \neg Q(x)\}$. □

Випадок 3. $F = \sigma_{\tilde{p}, R}(F_1)$. Нехай $\{x(R) \mid P(x)\}$ – вираз числення рядків, еквівалентний F_1 . Покладемо $E = \{x(R) \mid P(x) \wedge p(x(A_1), \dots, x(A_m))\}$, де $R = \{A_1, \dots, A_m\}$ схема таблиці, яка є значенням виразу F_1 . Тут припускається, що предикат-параметр селекції заданий так $\tilde{p}(s) = \text{true} \Leftrightarrow p(s(A_1), \dots, s(A_m)) = \text{true}$, $s \in S(R)$, де p – m -арний сигнатурний предикатний символ. □

Випадок 4. $F = \pi_{X,R}(F_1)$. Нехай $\{x(R) | P(x)\}$ – вираз числення рядків, еквівалентний F_1 . Покладемо E рівним $\{y(X \cap R) | \exists x(R)(P(x) \wedge \bigwedge_{A \in X \cap R} y(A) = x(A))\}$. $\square$

Випадок 5. $F = F_1 \otimes_{R_1, R_2} F_2$. Нехай $\{x(R_1) | P(x)\}$ і $\{y(R_2) | Q(y)\}$ – вирази числення рядків, еквівалентні F_1 та F_2 відповідно. Покладемо E рівним $\{z(R_1 \cup R_2) | \exists x(R_1) \exists y(R_2)(P(x) \wedge Q(y) \wedge \bigwedge_{A \in R_1} z(A) = x(A) \wedge \bigwedge_{A \in R_2} z(A) = y(A))\}$. $\square$

Випадок 6. $F = Rt_{\xi,R}(F_1)$, де $\xi: A \rightarrow A$ – ін'єктивна функція, що здійснює перейменування атрибутів. Тоді існує вираз числення рядків $\{x(R) | P(x)\}$, еквівалентний F_1 . Покладемо E рівним $\{y(R_2) | \exists x(R)(P(x) \wedge \bigwedge_{C \in R \setminus \text{dom} \xi} y(C) = x(C) \wedge \bigwedge_{A \in R \cap \text{dom} \xi} x(A) = y(\xi(A)))\}$, де $R_2 = R \setminus \text{dom} \xi \cup \xi[R]$. $\square$

Проведемо редукцію числення рядків до числення на домені. Задамо відображення $\varphi: F \rightarrow E$, яке кожному виразу числення рядків ставить у відповідність еквівалентний вираз числення на домені. Нехай $F = \{y(R) | P(y)\}$ – вираз числення рядків, змінна y – єдина змінна, яка входить у формулу P вільно, а $R = \{A_1, \dots, A_n\}$. Причому y – синтаксична змінна, областю зміни якої є змінні числення рядків. Застосуємо спочатку відображення φ до термів:

1. Кожну предметну константу d числення рядків залишимо без змін.
2. Кожен терм $x(A_i)$ числення рядків замінимо змінною $x_i \in D$ числення на домені.
3. Кожен терм $f(v_1, \dots, v_n)$, де v_i – терми числення рядків, $i = 1, \dots, n$, замінимо на $f(u_1, \dots, u_n)$, де u_j – терми числення на домені, $j = 1, \dots, n$, отримані в результаті попередніх замі.

Застосуємо відображення φ до атомів:

- a1. Кожен атом $t(z)$ числення рядків замінимо на $t(z_1, \dots, z_m)$, де $R = \{A_1, \dots, A_m\}$ схема таблиці t .
- a2. Кожен атом $p(v_1, \dots, v_m)$, де v_i – терми числення рядків, $i = 1, \dots, m$, замінимо на $p(u_1, \dots, u_m)$, де u_j – терми числення на домені, $j = 1, \dots, m$, отримані в результаті попередніх замі.

Кожну формулу P числення рядків перетворимо на формулу P' числення на домені, де формула P' – це формула P , у якій для кожного атома виконані заміни a1-a2, а кожна вільна змінна z схеми $R = \{A_1, \dots, A_m\}$ числення рядків замінена m змінними $z_1, \dots, z_m$ числення на домені. Формулу з квантором $\exists z(R_2)G$, $R_2 = \{A_1, \dots, A_m\}$ перетворимо на $\exists z_1(A_1) \dots \exists z_m(A_m)G'$. Формулу з квантором $\forall z(R_2)G$, $R_2 = \{A_1, \dots, A_m\}$ перетворимо на $\forall z_1(A_1) \dots \forall z_m(A_m)G'$.

Виконавши вказані заміни, отримаємо вираз числення зі змінними на домені $E = \{y_1, \dots, y_m \mid P(y_1, \dots, y_m)\}$. Очевидно, що кожна змінна z_i числення на домені може приймати точно ті самі значення, що і $z(A_i)$. Отже, значення виразу E співпадає із значенням виразу F . Тому має місце наступна теорема.

Теорема 2. Якщо F – вираз числення рядків, то можна ефективно (і рівномірно) побудувати еквівалентний йому вираз E числення на домені.

Теорема 3. Для кожного виразу числення на домені E можна ефективно (і рівномірно) побудувати еквівалентний йому вираз F табличної алгебри.

Доведення.

Нехай $E = \{x_1, \dots, x_n \mid P(x_1, \dots, x_n)\}$ – вираз числення на домені, значенням якого є таблиця схеми $R = \{A_1, \dots, A_n\}$. Побудуємо для кожної підформули G формули P еквівалентний алгебраїчний вираз F_G . Доведемо індукцією по числу операторів у підформулі G з P , що якщо G містить вільні змінні на універсальному домені $y_1, \dots, y_m$, то для виразу $\{y_1, \dots, y_m \mid G(y_1, \dots, y_m)\}$ існує еквівалентний вираз табличної алгебри F_G . Тоді, якщо G є сама P , отримаємо алгебраїчний вираз для $\{x_1, \dots, x_n \mid P(x_1, \dots, x_n)\}$. Припускається, що $y_1, \dots, y_m$ – всі вільні змінні в G і значенням виразу $\{y_1, \dots, y_m \mid G(y_1, \dots, y_m)\}$ є таблиця схеми $R_G = \{A_1, \dots, A_m\}$.

Замінімо змінні у формулі P таким чином, щоб жодна змінна не входила одночасно і вільно, і зв'язано у P та не була зв'язана квантором у двох різних місцях. Тепер кожна змінна асоційована з атрибутом, який або входить у квантор $\forall x(A)$ чи $\exists x(A)$, якщо змінна зв'язана у формулі P , або, якщо змінна вільна в P , співставлений змінній x_i , що знаходиться у виразі E зліва від вертикальної лінії. Для будь-якого атрибута A можна виписати алгебраїчний вираз, значенням якого є одноатрибутна таблиця $\langle t, \{A\} \rangle$, що містить всі рядки $s = \langle A, d_i \rangle$, $d_i \in D$ – всі елементи множини D . Позначимо його через $[D]$.

База індукції (немає операторів, тобто логічних зв'язок і кванторів). Підформула G – атом вигляду $p(u_1, \dots, u_m)$ або $t(a_1, \dots, a_n)$.

Нехай G має вигляд $p(u_1, \dots, u_m)$, де u_i – терми числення на домені, причому $y_1, \dots, y_k$ – всі змінні цих термів. Покладемо F_G рівним $\sigma_{\tilde{p}}([D]_1 \otimes_{R_1, R_2} \dots \otimes_{R_1 \cup \dots \cup R_{k-1}, R_k} [D]_k)$, де R_i – одноатрибутні схеми таблиць, причому ці таблиці є значеннями алгебраїчних виразів $[D]_i$, $i = 1, \dots, k$. □

Нехай G має вигляд $t(a_1, \dots, a_k)$, де a_i – константа або змінна на універсальному домені D . Нехай $R = \{C_1, \dots, C_k\}$ – схема таблиці $\langle t, R \rangle$. Покладемо алгебраїчний вираз F_G рівним $\pi_X(\sigma_{\tilde{p}}(t))$, де $\tilde{p}$ – предикат-параметр селекції, який є кон'юнкцією порівнянь $C_i = a_i$ по всіх i , для яких a_i – константа; X – множина атрибутів $\{C_j \mid a_j \text{ – змінна}\}$. □

Крок індукції.

Припустимо, що підформула G містить принаймні один оператор і що гіпотеза індукції справедлива для всіх підформул формули P , що мають менше операторів, ніж G .

Випадок 1. $G = \neg Q$. Нехай F_Q – алгебраїчний вираз для Q , а значенням виразу F_Q є таблиця схеми R_Q . Покладемо $F_G = \sim_{R_Q} F_Q$. $\square$

Випадок 2. $G = Q \vee Q'$. Нехай вільні змінні з Q – це $z_1, \dots, z_k, v_1, \dots, v_p$, а з Q' – це $z_1, \dots, z_k, w_1, \dots, w_q$, де змінні $v_1, \dots, v_p$ та $w_1, \dots, w_q$ попарно не співпадають. Нехай F_Q і $F_{Q'}$ – алгебраїчні вирази для Q і Q' відповідно, причому C_i , $i = 1, \dots, p$ – атрибути, асоційовані зі змінними $v_1, \dots, v_p$, а K_j , $j = 1, \dots, q$ – атрибути асоційовані із змінними $w_1, \dots, w_q$.

Покладемо рівним $F_1 = F_Q \otimes_{R_Q, \{C_1\}} [D]_1 \otimes_{R_Q \cup \{C_1\}, \{C_2\}} \dots \otimes_{R_Q \cup \{C_1, \dots, C_{p-1}\}, \{C_p\}} [D]_p$, а $F_2 = F_{Q'} \otimes_{R_{Q'}, \{C_1\}} [D]_1' \otimes_{R_{Q'} \cup \{C_1\}, \{C_2\}} \dots \otimes_{R_{Q'} \cup \{C_1, \dots, C_{p-1}\}, \{C_p\}} [D]_p'$, де R_Q та $R_{Q'}$ – схеми таблиць, які є значеннями виразів F_Q та $F_{Q'}$ відповідно, а C_i та K_j – одноатрибутні схеми таблиць, які є значеннями алгебраїчних виразів $[D]_i$ та $[D]_j$ відповідно, $i = 1, \dots, p$, $j = 1, \dots, q$. Позначимо через R_{F_1} та R_{F_2} – схеми таблиць, які є значеннями виразів F_1 та F_2 відповідно. Для відповідним чином побудованих виразів F_1 та F_2 , їхні схеми R_{F_1} та R_{F_2} рівні. Тому покладемо $F_G = F_1 \cup_{R_{F_1}} F_2$. $\square$

Випадок 3. $G = Q \wedge Q'$. За законом де Моргана $G = \neg(\neg Q \vee \neg Q')$. $\square$

Випадок 4. $G = \exists x(A)Q$. Нехай F_Q – алгебраїчний вираз для Q . Покладемо $F_G = \pi_{X \setminus A, X}(F_Q)$, де X – схема таблиці, що є значенням виразу F_Q . $\square$

Випадок 5. $G = \forall x(A)Q$. Даний вираз числення на домені можна виразити через операцію заперечення та квантор існування $\forall x(A)Q = \neg(\exists x(A))(\neg Q)$. $\square \square$

Таким чином, встановлено основний результат

Теорема 4. Узагальнена таблична алгебра, (узагальнене) числення рядків і (узагальнене) числення на домені еквівалентні.

6. Висновки

Проведено узагальнення табличної алгебри на таблиці, яким приписується певна схема. Класичне числення рядків та числення на домені поповнено довільними предикатними та функціональними сигнатурами на універсальному домені D . Визначено синтаксис термів, атомів та формул числення рядків і числення на домені. Доведено, що узагальнена таблична алгебра, узагальнене числення рядків та узагальнене числення на домені еквівалентні.

ЛИТЕРАТУРА

1. Реляційні бази даних: табличні алгебри та SQL-подібні мови / В.Н. Редько, Ю.Й. Борона, Д.Б. Буй, С.А. Поляков. – Київ: Видавничий дім «Академперіодика», 2001. – 198 с.
2. Мейер Д. Теория реляционных баз данных. – М.: Мир, 1987. – 608 с.
3. Буй Д.Б., Глушко І.М.: Узагальнена таблична алгебра та узагальнене реляційне числення // Теоретичні та прикладні аспекти побудови програмних систем: міжнародна конференція, 4-8 жовтня 2010 року, Київ, Україна: тези доповідей. – Київ. – 2010. – С. 112-120.
4. Шенфилд Дж. Математическая логика. – М.: Наука, 1975. – 528 с.
5. Дейт К. Дж. Введение в системы баз данных: [8-е изд.: пер. с англ.]. – М.: Издательский дом «Вильямс», 2005. – 1328 с.