

СППР З КЕРУВАННЯ РОЗПОДІЛОМ ОБМЕЖЕНИХ РЕСУРСІВ

©Верес О.М., Верес Ю.О., Катренко А.А., 2008

Досліджено моделі задач розподілу обмежених ресурсів та проблеми оперативного управління в системі споживачів ресурсів. Запропонована структура системи підтримання прийняття рішень з розподілу ресурсів та підтримання їх запасів на належному рівні. Визначені функції основних складових СППР.

Task models of division of the limited resources and problems of operative management in the system of resource users are probed in the article. The structure of Decision Support System of the division of resources and support of their supplies on a due levels is offered. The functions of basic constituents of Decision Support System are certain.

Постановка проблеми у загальному вигляді

Проблема автоматизації різноманітних процесів управління і опрацювання даних на основі використання сучасних засобів комп'ютерної техніки є однією з головних народногосподарських проблем. Ці проблеми багатогранні, і їх успішне вирішення залежить від сумісних зусиль спеціалістів різного профілю: інженерів, технологів, математиків, програмістів, економістів, спеціалістів з питань управління та інших галузей науки і техніки.

Це пояснюється тим, що розроблення систем підтримання прийняття рішень (СППР) і оперативного управління складними процесами передбачає зазвичай необхідність системного вирішення низки завдань: розроблення і обґрунтування математичних моделей різних етапів автоматизації процесу, визначення комплексу першочергових задач, які повинні бути вирішені, обрання системотехнічних засобів автоматизації, економічного обґрунтування прийнятих рішень на різних етапах розроблення, створення програмного забезпечення системи автоматизації, вирішення питань забезпечення необхідної надійності функціонування процесів в умовах автоматизованого управління ними тощо.

Аналіз останніх досліджень і публікацій

Системи, в яких розв'язуються задачі розподілу обмежених ресурсів та управління запасами, є системами ієрархічного типу, а тому в загальному ці задачі розв'язуються розподілено за рівнями ієрархії: на вищих рівнях розв'язуються агреговані задачі, а на нижчих рівнях ступінь деталізації для кожного окремого елемента ієрархії зі збільшенням її глибини зростає. Загальна узгодженість такого системного рішення забезпечується процесами координації в межах одного рівня ієрархії та стабілізуючими зворотними зв'язками між сусідніми рівнями ієрархії. Однак, хоча задачі, що розв'язуються в ієрархічній системі, є задачами різного ступеня агрегування, за типом математичних моделей у більшості випадків локально вони є задачами математичного програмування.

Найзагальнішими задачами цього типу є задачі мішаного програмування, хоча у більшості задач розподілу ресурсів ці задачі з достатнім рівнем точності можуть бути представлені моделями лінійного мішаного програмування або апроксимовані сукупністю таких моделей.

Відомо, що ці задачі належать до класу NP (Nondeterministic Polynomial) – повних, тобто сподіватися на побудову ефективних алгоритмів знаходження точних оптимальних розв'язків для них не треба. Водночас точність первісної інформації є не стовідсотковою, що дає змогу застосовувати для розв'язання таких задач ϵ -оптимальні алгоритми.

Для розв'язання мішаних лінійних задач існують достатньо ефективні алгоритми, за якими можна отримати ϵ -оптимальні розв'язки за поліноміальний час, тобто у випадку пошуку субоптимальних розв'язків з заданою точністю ϵ задачі NP-класу переходять до класу P-повних.

Дискретні оптимізаційні задачі широко застосовуються в різних галузях, де використовуються математичні методи для аналізу процесів в системах керування. Тому знання методів дискретної та мішаної лінійно-дискретної оптимізації є важливим елементом освіти фахівців в галузі як комп'ютерних наук, так і системного аналізу.

Зазначимо, що ще близько двадцяти років тому розв'язання кожної серйозної задачі дискретного програмування було пов'язане з великими труднощами; це часто було мистецтвом, хоча основні алгоритми були вже розроблені. Сьогодні ситуація принципово змінилась. Ми вважаємо, що це пов'язано в основному з двома обставинами. По-перше, на порядки збільшилась потужність обчислювальної техніки. По-друге, вдалось зрозуміти тонкощі обчислювальної реалізації комбінаторних методів і алгоритмів, істотно пов'язаних з особливостями розв'язуваних задач. І хоча досі можуть зустрічатися дуже складні задачі, їх розв'язання вже перестало бути мистецтвом і може бути реалізоване у більшості випадків за допомогою стандартних засобів. Але труднощі, пов'язані з перебором великої кількості варіантів, зберігаються за будь-якої потужності комп'ютерної техніки [4]. Зрозуміло, що моделі такого типу є детермінованими, тобто не враховують випадкових збурень – що у повній моделі виявилось б у наявності некерованих випадкових змінних. Своєю чергою, спроби це зробити призводять до задач стохастичного програмування, які взагалі розв'язуються дуже погано.

Важливим і тонким моментом при прийнятті управляючих рішень є ідентифікація некерованих змінних, тобто змінних, зміна значень яких не залежить від суб'єкта управління. Ігнорування в реальних ситуаціях некерованих змінних може привести до побудови неадекватної моделі і, отже, до прийняття помилкових рішень.

Вплив некерованих змінних по суті є виявом рівня невизначеності навколишнього середовища.

Як відомо [4], у цьому випадку не існує кращого варіанта, ніж розв'язання задачі з рівнем точності (ϵ), що відповідає рівню залишкової ентропії навколишнього середовища. І це також зрозуміло: чим менший рівень хаотичності, тим доцільнішим є точніше розв'язання оптимізаційної задачі. І навпаки – при певному значному рівні невизначеності може бути обраний будь-який припустимий варіант.

Складність реальних систем може значно ускладнити подання цілі і обмежень в аналітичному вигляді. Тому дуже важливо зменшити розмірність задачі, зберігаючи необхідний рівень адекватності моделі. Незважаючи на надзвичайно велику кількість змінних і обмежень, котрі, на перший погляд, необхідно враховувати при аналізі реальних ситуацій, лише невелика їх частина виявляється істотною для опису поведінки досліджуваних систем. Тому при спрощеному описанні реальних систем, на ґрунті якого будуватиметься та чи інша модель, насамперед треба ідентифікувати домінуючі змінні, параметри і обмеження і обрати адекватний рівень абстрагування при моделюванні реального об'єкта [5].

Іншим аспектом, що сприяє зменшенню розмірності задач, є ієрархічна структура системи. Це дає змогу розв'язувати на верхньому рівні задачу невеликої розмірності, а також розв'язувати задачі порівняльної розмірності у всіх вузлах прийняття рішень ієрархічної системи. Основна проблема, що виникає при розв'язанні задачі розподілу ресурсів та управління запасами, полягає в створенні ефективної і надійної системи підтримання необхідного рівня запасів, а також розподілу та перерозподілу певних ресурсів у межах ієрархії системи.

Процес управління реалізується так: після чергового розподілу ресурсів ієрархічна система прагне підтримувати рівень запасів не меншим, ніж граничний. Однак наявний ресурс вичерпується, і в момент надходження певної кількості додаткового ресурсу розв'язується задача розподілу, а після цього шляхом оперативного регулювання підтримується необхідний рівень запасів у всіх елементах системи. Природно, що чим вищий рівень управління, тим більш вагомим та значущим є відповідне рішення. Спроби побудувати повністю автоматичні підприємства у 60–70-ті роки минулого сторіччя виявили наявність однієї непереборної проблеми, а саме: підприємство – це не паровий двигун, де революцією стало застосування регулятора Уатта, і взагалі в складних системах застосування методів теорії автоматичного регулювання є неефективним. Саме

тому виникла потреба в інтерактивних засобах підтримання прийняття рішень. Насамперед йдеться про комп'ютерну підтримку діяльності керівників різного рівня. Вирішувати це завдання покликані комп'ютерні системи підтримки прийняття рішень (СППР, в англomовному варіанті – Decision Support System (DSS)). СППР – це інтерактивні автоматизовані інформаційні системи, які допомагають децидентам використовувати інформацію та моделі для того, щоб вирішувати слабо структуровані проблеми.

На рівні користувача Haettenschwiler поділяє СППР на пасивні, активні і кооперативні [7]. Пасивною СППР називається система, яка допомагає процесу ухвалення рішення, але не може пропонувати, яке рішення прийняти. Активна СППР може зробити пропозицію, яке рішення обрати. Кооперативна дає змогу дециденту змінювати, поповнювати або покращувати рішення, запропоновані системою, надсилаючи потім ці зміни в систему для перевірки. Система змінює, поповнює або покращує ці рішення і надсилає їх знову користувачеві. Процес триває до отримання узгодженого рішення.

На концептуальному рівні Power вирізняє СППР керування повідомленнями (Communication-Driven DSS), СППР керування даними (Data-Driven DSS), СППР керування документами (Document-Driven DSS), СППР керування знаннями (Knowledge-Driven DSS) і СППР, керування моделями (Model-Driven DSS). СППР керування моделями характеризують в основному доступ і маніпуляції з математичними моделями (статистичними, фінансовими, оптимізаційними, імітаційними) [8]. Зазначимо, що деякі OLAP-системи, що дають змогу здійснювати складний аналіз даних, можуть належати до гібридних СППР, які забезпечують моделювання, пошук і опрацювання даних.

СППР керування повідомленнями (Communication-Driven DSS) (раніше групова СППР – GDSS) підтримує групу користувачів, що працюють над виконанням загального завдання.

СППР керування даними (Data-Driven DSS) або СППР, орієнтовані на роботу з даними (Data-oriented DSS), є орієнтованими на доступ і маніпуляції з даними. СППР, керування документами (Document-Driven DSS) керують, здійснюють пошук і маніпулюють неструктурованою інформацією, що може існувати в різних форматах. Нарешті, СППР, керовані знаннями (Knowledge-Driven DSS) забезпечують розв'язання задач у вигляді фактів, правил, процедур.

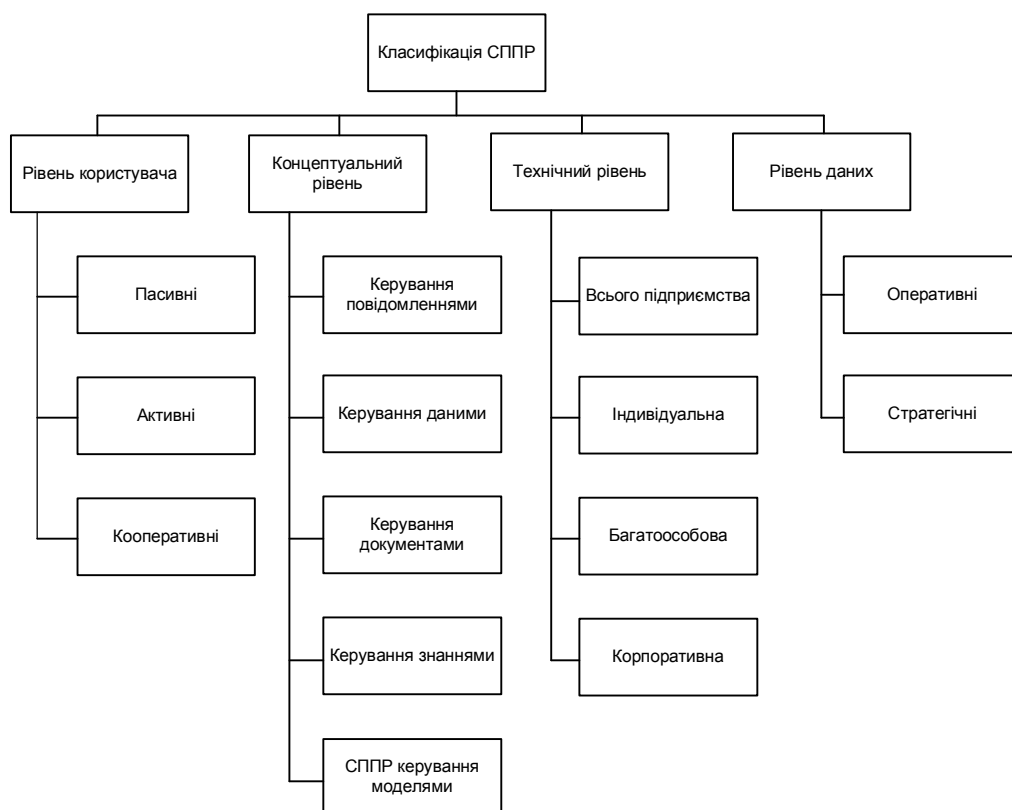


Рис. 1. Класифікація СППР

На технічному рівні Power розрізняє СППР всього підприємства, індивідуальну, багатособову та корпоративну СППР [9]. СППР підприємства загалом має можливість доступу до великих сховищ інформації і обслуговує багато менеджерів підприємства. Індивідуальна СППР – це мала система, що обслуговує лише один комп'ютер користувача. Існують і інші класифікації (Alter [10], Holsapple і Whinston [11], Golden, Hevner і Power [12]). Зазначимо лише, що досконала для свого часу класифікація Alter'a, яка розбивала всі СППР на 7 класів, сьогодні дещо застаріла.

Залежно від даних, з якими ці системи працюють, СППР умовно можна розділити на оперативні і стратегічні (див.рис.1). Оперативні СППР призначені для негайного реагування на зміни поточній ситуації в управлінні фінансово-господарськими процесами компанії. Стратегічні СППР орієнтовані на аналіз значних обсягів різномірної інформації, що збираються з різних джерел і знаходяться на різних рівнях.

Marakas [13] запропонував узагальнену архітектуру СППР, що складається з 5 різних частин, а саме: система управління даними (the data management system – DBMS); система управління моделями (the model management system – MBMS); машина знань (the knowledge engine (KE)); інтерфейс користувача (the user interface); користувачі (the user(s)).

Отже, ґрунтуючись на цьому аналізі, вирішеними не в повному обсязі є такі підпроблеми:

- проблема оптимального розподілу ресурсів між споживачами;
- проблема обрання оптимального маршруту постачання ресурсів;
- проблема оперативного перерозподілу ресурсів.

Цілі (завдання) статті

Основним завданням статті є дослідження моделей задач оптимального розподілу обмежених ресурсів та розроблення структури системи підтримки прийнятих рішень, яка розв'язуватиме такі задачі та реалізовуватиме процеси оперативного управління перерозподілу ресурсів.

Основний матеріал дослідження

Метою створення прикладної СППР є оптимальний розподіл обмежених ресурсів між споживачами, котрі об'єднані між собою у мережу та підтримання їх запасів на належному рівні. В загальному випадку це транспортна мережа, хоч можуть бути й інші варіанти. Під оптимальним розподілом розуміють такий, який мінімізує вартість доставки ресурсів до споживачів та задовольняє потреби споживачів. Задачі, що виникають, є оптимізаційними задачами для слабо структурованого середовища, а тому й поняття «оптимальності» є суб'єктивним.

Мету та функції системи зручно подати у вигляді дерева цілей, отриманого в результаті проведеного системного аналізу предметної області (див. рис. 1).

Виходячи із структури отриманого дерева цілей можна запропонувати декілька різних структур СППР залежно від типів розв'язуваних задач і методу їх розв'язання:

- СППР із використанням евристичних алгоритмів. Використання таких алгоритмів дасть змогу за короткий час отримати відповідь, близьку до оптимальної.
- СППР із використанням комбінованих алгоритмів. Дасть змогу отримати оптимальний розв'язок певної задачі або наближений залежно від потреб користувача системи, розміру задачі та обчислювальних потужностей. Із використанням комбінованих алгоритмів можна отримати близький до оптимального розв'язок задачі великої розмірності.
- СППР із використанням точних алгоритмів. За успішного розв'язання задачі завжди буде отриманий оптимальний розв'язок. Недоліком є те, що для того, щоб отримати розв'язок задачі великої розмірності, потрібно або дуже багато часу, або взагалі розв'язок може бути не знайдений.

Для того, щоб проаналізувати таку складну структуру, було використано метод аналітичної ієрархії з результатами опитування експертів. У результаті аналізу отримуємо вектор пріоритетів для структур СППР (див. табл. 1).

Таблиця 1

Вектор пріоритетів для структур СППР

Структура СППР	Значення вектора пріоритетів
СППР із використанням евристичних алгоритмів	0,079
СППР із використанням комбінованих алгоритмів	0,659
СППР із використанням точних алгоритмів	0,263

Отже, очевидно, що в результаті аналізу найкращою структурою цієї системи підтримки прийнятих рішень є СППР із використанням комбінованих алгоритмів, оскільки саме така альтернатива отримала найвище значення у векторі пріоритетів – 0,689.

Тобто, в результаті аналізу структури задачі система визначає метод декомпозиції та ітеративного процесу отримання бажаного розв'язку.

У загальному випадку задачі, що виникають, є задачами пошуку розташування елементів складної системи з одночасним визначенням структури матеріальних, енергетичних та інформаційних потоків між ними та їх оптимізацією.

Зрозуміло, що навіть у випадку лінійних залежностей в результаті виникають багатоіндексні задачі лінійного програмування, які в загальному випадку не можуть бути розв'язані безпосередньо. Саме тому для первісної задачі розподілу ресурсів визначається шлях декомпозиції на послідовність складових простіших задач, і остаточний розв'язок досягається в результаті ітеративної процедури розв'язання взаємопов'язаних складових. Такими складовими є задача про розміщення центрів у вершинах та на ребрах графу, що розв'язуються за допомогою різноманітних варіантів методу Хакімі та евристичного методу Сінтера [15], та різноманітних варіантів поточкових задач, які у більшості випадків можуть бути подані формально у вигляді поточної задачі загального типу.

Задачі поточкового типу можуть бути сформульовані як задачі лінійного програмування та розв'язані за допомогою симплекс-методу, але ця можливість є лише принциповою, тому що в цьому випадку розмірність відповідних задач лінійного програмування буде надзвичайно великою, і матриці коефіцієнтів обмежень дуже розрідженими – тобто кількість нульових елементів досягатиме 90–98%. Водночас спеціальна структура окремих класів поточкових задач дає змогу побудувати ефективні алгоритми їх розв'язування.

Розглянемо загальну задачу поточкового типу як мережу, в якій існує α джерел, β витоків та ϕ проміжних вузлів (див.рис.3). Однопродуктовий потік повинен проходити з джерел у виток через проміжні вузли. Характеристиками кожної дуги мережі $(i, j) \in A$ є пропускна здатність u_{ij} та вартість доставки нею одиниці потоку c_{ij} .

Необхідно визначити потік мінімальної вартості з джерел у виток за умови, що наявно не більш ніж $\sum_{i \in N_a} a_i$ одиниць продукту, а у виток повинно бути доставлено не менш ніж

$\sum_{i \in N_b} b_i$ одиниць продукту, де N_a, N_b, N_j — множини джерел, витоків та проміжних вузлів

відповідно, a_i — кількість продукту в джерелі a_i , b_j — потреби в продукті у витокі b_j .

Формальна постановка задачі про максимальний потік мінімальної вартості є такою.

$$\sum_i \sum_j c_{ij} f_{ij} \Rightarrow \text{Min} \quad , \quad (1)$$

$$\sum_j f_{ij} - \sum_j f_{ji} \leq a_i, \quad i \in N_a \quad , \quad (2)$$

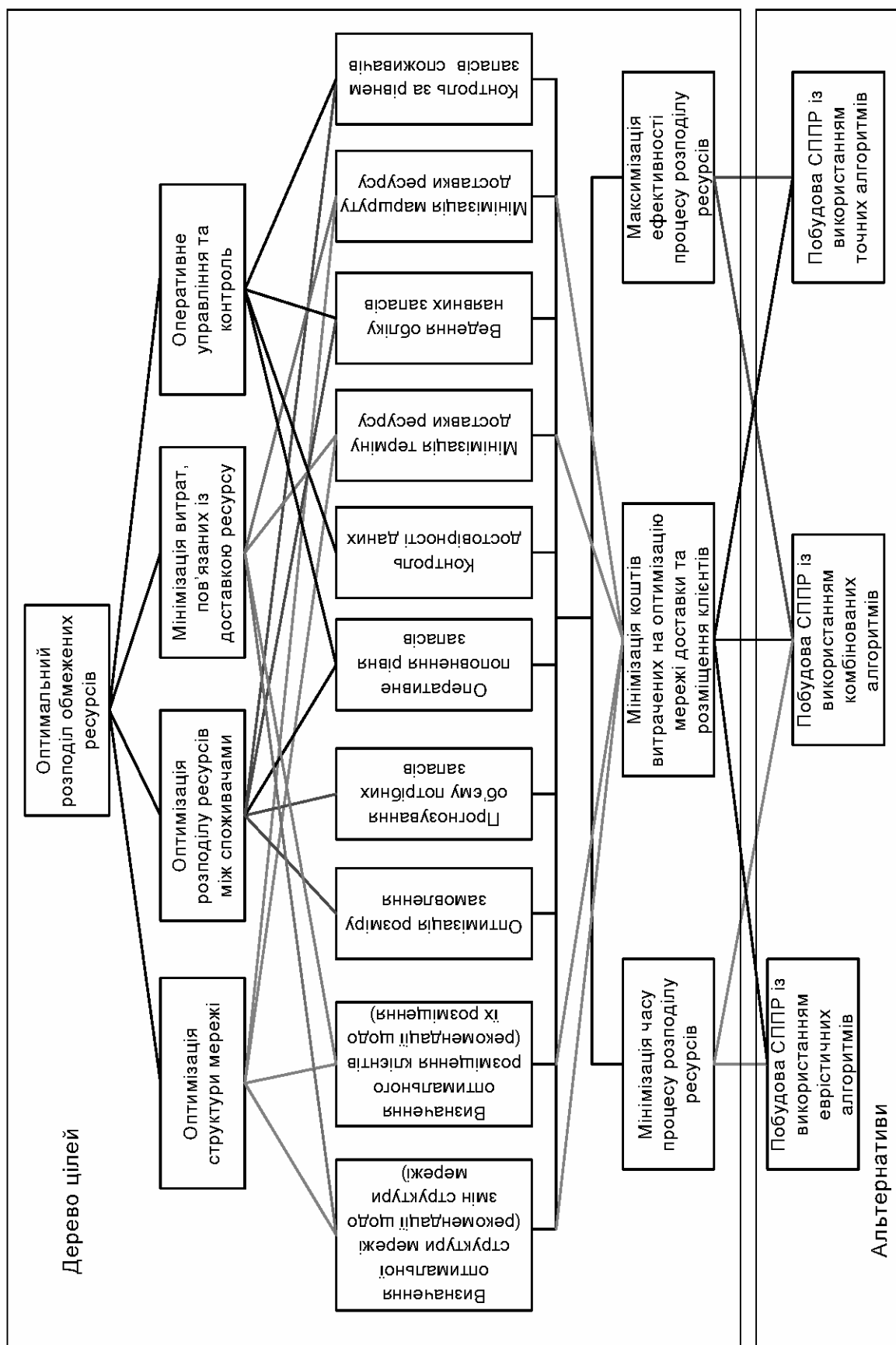


Рис. 2. Дерево цілей та альтернатив

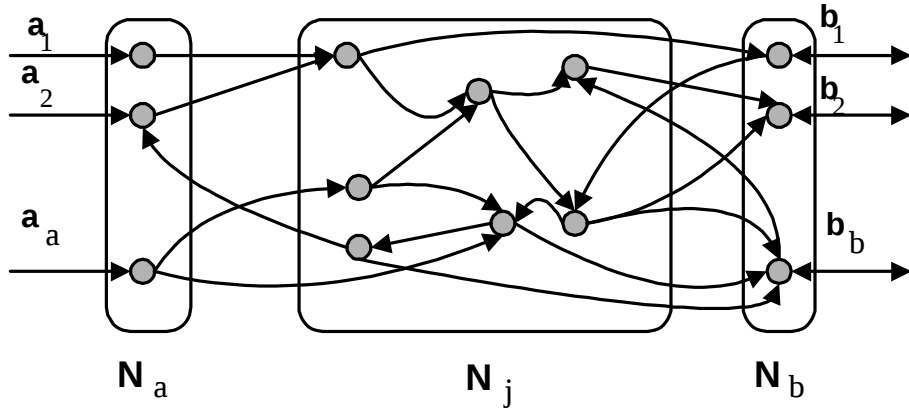


Рис. 3. Загальна потокова задача

$$\sum_j f_{ij} - \sum_j f_{ji} = 0, \quad i \in N_j, \quad (3)$$

$$\sum_j f_{ji} - \sum_j f_{ij} \geq b_i, \quad i \in N_b, \quad (4)$$

$$0 \leq f_{ij} \leq u_{ij}, \quad (i, j) \in A \quad (5)$$

Нерівності групи (2) відображають той факт, що в мережу з джерела не може надходити більша кількість продукту, ніж наявна, (3) — умови зберігання потоку для кожного з проміжних вузлів, (4) — попит на продукт у витоків повинен бути задоволений, (5) — потоки дугами є обмеженими пропускними можливостями дуг.

Окремими випадками цієї задачі є відомі задачі — транспортна, задача про призначення, задача про максимальний потік, задача про найкоротший ланцюг. Транспортна задача виникає у тому випадку, коли потік проходить безпосередньо з джерел до витоків без проміжних пунктів, тобто $N_j = \emptyset$, $u_{ij} = \infty$. Задачу про призначення отримаємо, якщо потік проходить безпосередньо з джерел до витоків без проміжних пунктів: $N_j = \emptyset$, $u_{ij} = \infty$. Крім того $\forall i \in N_a : a_i = 1$, та $\forall i \in N_b : b_i = 1$. Задача про комівояжера є задачею про призначення з наявністю додаткових обмежень, що істотно ускладнює її розв'язання порівняно з задачею про призначення. Задача про максимальний потік виникає, коли в мережі є лише одне джерело s та витік t , тобто $\text{card}(N_a) = \text{card}(N_b) = 1$. Кількість продукту, що надходить в мережу через джерело, не обмежена, а мінімальні потреби дорівнюють нулю — $a = \infty$, $b = 0$. Пропускні здатності дуг обмежені значеннями u_{ij} , вартості пересилання одиниці потоку дугами становлять $\forall ((i, j) \in A) \wedge (j \neq t) : c_{ij} = 0$, $\forall ((i, t) \in A) : c_{it} = -1$. У результаті отримаємо задачу максимізації значення сумарного потоку, що надходить з мережі до витоків:

$$\sum_{(i,t) \in A \wedge i \neq t} f_{it} \Rightarrow \text{Max}, \quad \sum_j f_{ij} - \sum_j f_{ji} = 0, \quad i \neq s, \quad i \neq t \quad (6)$$

$$0 \leq f_{ij} \leq u_{ij}, \quad (i, j) \in A.$$

Задачу про найкоротший ланцюг отримаємо за таких умов. Вважатимемо, що c_{ij} — вартість передачі одиниці потоку з вузла i в вузол j . Нехай $\text{card}(N_a) = \text{card}(N_b) = 1$, тобто в мережі є лише одне джерело та один витік, в мережу з джерела надходить $a = 1$ потоку, попит у витоків становить $b = 1$. Потік будь-якою дугою не може бути більшим за 1. Мережа не має контурів

від'ємної ваги. Задача знаходження потоку мінімальної вартості в такому випадку означатиме не що інше, як знаходження найкоротшого шляху з джерела до витоку — c_{ij} в цьому випадку розглядатимемо як віддалі — довжини дуг.

$$\sum_i \sum_j c_{ij} f_{ij} \Rightarrow \text{Min}, \quad \sum_j f_{sj} - \sum_j f_{js} = 1, \quad \sum_j f_{tj} - \sum_j f_{jt} = -1, \quad (7)$$

$$\sum_j f_{ij} - \sum_j f_{ji} = 0, \quad i \neq s, i \neq t, (i, j) \in A.$$

Ці задачі мають одну дуже важливу особливість: якщо значення a_i, b_i, u_{ij} — цілі, то будь-який базовий розв'язок загальної задачі про максимальний потік мінімальної вартості є цілочисельним внаслідок унімодулярності матриці коефіцієнтів обмежень.

Внаслідок цього розроблено низку ефективних алгоритмів розв'язання задач поточкового типу.

Однак розглянута загальна задача ((1)÷(5)) є задачею однопродуктового типу, задачі, що виникають у багатьох випадках, є багатопродуктивними. Для розв'язання багатопродуктивної задачі знову ж таки використовується принцип декомпозиції — багатопродуктова задача декомпозується до множини взаємопов'язаних однопродуктових задач, і остаточний розв'язок отримується в результаті певної кількості ітерацій.

Оскільки основними складовими задачами, що розв'язуються в системі, є задачі цілочисельного та мішаного програмування, то в загальному для їх розв'язання застосовуються методи, що належать одній з трьох основних груп.

Для розв'язання задач розподілу обмежених ресурсів можливе застосування таких (основних) груп методів:

- метод відтинання для задач (частково) цілочисельного лінійного програмування;
- комбінаторні методи;
- наближені методи.

Методи відтинання використовуються для розв'язання цілочисельних та мішаних задач лінійного програмування та будуються на ідеї поступового відтинання фрагментів області припустимих рішень зі збереженням всіх припустимих цілочисельних розв'язків задачі. Вперше ця ідея була запропонована Д. Данцігом і реалізована шляхом поповнення множини первісних обмежень на кожній ітерації до моменту отримання цілочисельного розв'язку, який і буде оптимальний. Ідея відтинання призводить до трьох проблем:

- знаходження універсального правила формування відтинань;
- доведення кінця процесу відтинання;
- боротьба з надмірним “розростанням” розмірності задач у разі додавання обмежень.

Тільки розв'язання цих проблем може привести до універсального і такого, що можна реалізувати в обчислювальному відношенні алгоритму. Р. Гоморі вдалось розв'язати ці проблеми, але було виявлено велику непередбачуваність у поведінці різних алгоритмів відтинання. Обчислювальний експеримент показав, що існують задачі порівняно невеликої розмірності, розв'язок яких не вдається отримати при великих затратах машинного часу. Надалі було знайдене теоретичне підґрунтя цього явища. Було показано, що для деяких варіантів алгоритму Гоморі існують задачі, в яких кількість відтинань швидко росте з ростом розмірності задачі і ростом коефіцієнтів, а в поточних симплекс-таблицях з'являються вельми великі числа.

Основна ідея комбінаторних методів полягає у використанні скінченної кількості допустимих розв'язків і заміні повного їх перебору скороченим (напрямленим перебором). Головними в скороченні перебору є оцінки (границі, межі) та критерії відтинання підмножин, які завідомо не містять оптимальних розв'язків. Ця ідея реалізується шляхом послідовного розбиття множини всіх припустимих розв'язків на підмножини, що породжує у процесі розв'язання динамічну деревоподібну структуру. Серед підмножин, що послідовно породжуються на кожному кроці процесу, можуть виявитися такі, що не містять припустимих розв'язків. Відтинання таких

підмножин дає змогу замінити повний перебір частковим і значно підвищити ефективність порівняно з методом повного перебору. Комбінаторні методи відрізняються способом розбиття і способом оцінювання, ці способи зазвичай пов'язані зі специфікою класів задач. До цієї групи належать такі методи: метод гілок і границь; схема послідовного аналізу варіантів; метод динамічного програмування; апроксимаційно-комбінаторний метод; метод послідовності розрахунків. Зазначимо, що універсальність комбінаторних методів обумовлює надзвичайну широту їх застосування до конкретних задач.

Сучасні наближені методи зазвичай є комбінованими, тобто містять елементи різних методів. У наближених методах розв'язання задачі виконується зазвичай в два етапи: побудова початкового розв'язку і покращання початкового розв'язку. При цьому на першому етапі широко використовуються *евристичні алгоритми* – алгоритми, що ґрунтуються на правдоподібних, але не обґрунтованих строго припущеннях про властивості оптимального розв'язку задачі. На другому етапі використовуються алгоритми локальної оптимізації, пов'язані в введеним поняттям околиці; при цьому можна використовувати декілька алгоритмів цього типу, змінюючи правила вибору околиці. Зазначимо також, що сьогодні поширення набули наближені методи, які є модифікаціями точних методів, з самого початку орієнтовані на пошук наближеного розв'язку з оцінкою відхилення від оптимального.

Підсумовуючи усе вищесказане, можна дійти висновку, що обрання того чи іншого методу залежить від типу задачі та від її розмірності. Для невеликих задач доцільно обирати точні методи розв'язання (комбінаторні), а для задач значної розмірності – методи, що дають приблизний (близький до оптимального із певною мірою точності) розв'язок (евристичні). Одним із найкращих варіантів є вибір комбінованих методів – після визначення типу і розміру задачі застосування до неї відповідного алгоритму, який дасть потрібний у даній ситуації розв'язок.

Отже, первісна задача аналізується, і якщо вона може бути безпосередньо розв'язана, залежно від додаткових умов обирається алгоритм оптимального (чи субоптимального) розв'язку. Головними з цих умов є обмеження у часі. Якщо планується майбутня поведінка, то час, що відводиться на отримання рішення, є доволі значним, і застосовуються складніші алгоритми, а якщо завдання полягає в оперативному корегуванні системи чи поверненні її на планову траєкторію, використовуються алгоритми пошуку наближених рішень (як варіант – обрання стратегії «відразу–вглиб» у методі границь та розгалужень).

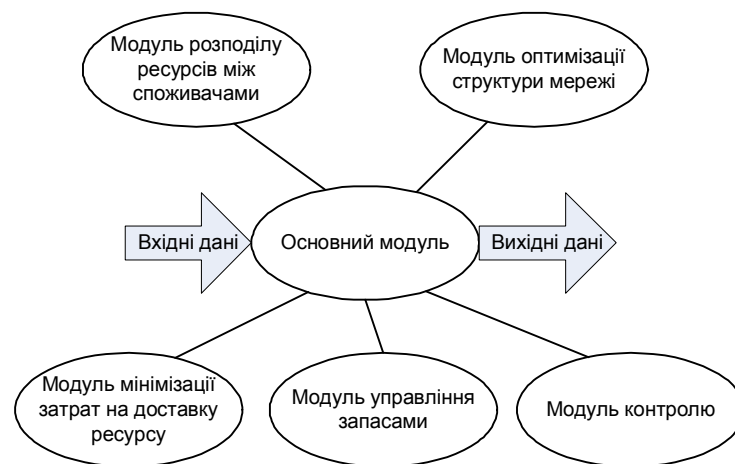


Рис. 4. Внутрішня модульна структура СППР

Результати аналізу дерева цілей дали змогу обґрунтувати структуру СППР (див.рис.4), яка містить такі модулі: модуль оптимізації структури мережі; модуль розподілу ресурсів між споживачами; модуль мінімізації затрат на доставку ресурсу; модуль контролю процесу оптимізації; модуль взаємодії, модуль управління запасами. Деякі модулі мають схожі функції, але розв'язують при цьому різні задачі з однією метою – оптимізувати розподіл ресурсів між споживачами. Кожен модуль має низку методів для розв'язання задач. Відповідно до цих методів він виконує операції, які властиві тільки йому (табл. 2).

Функції модулів

Назва модуля	Перелік функцій
Основний модуль (модуль взаємодії)	– забезпечення взаємодії між модулями; – керування модулями; – взаємодія із користувачем; – забезпечення введення/виведення; – збирання статистичних даних.
Модуль розподілу ресурсів між споживачами	– розв'язання задач, пов'язаних із розподілом ресурсів між споживачами; – взаємодія з головним модулем.
Модуль мінімізації затрат на доставку ресурсу	– розв'язання задач, пов'язаних із мінімізацією затрат на доставку ресурсу; – взаємодія із основним модулем.
Модуль оптимізації структури мережі	– розв'язання задач, пов'язаних із оптимізацією мережі; – взаємодія із основним модулем.
Модуль контролю	– перевірка вхідних і вихідних даних; – перевірка достовірності даних; – перевірка оптимальності розв'язку; – контроль часу роботи системи; – виявлення і усунення помилок.
Модуль управління запасами	– контроль за рівнем наявних запасів; – визначення кількості замовленого запасу; – визначення строку розміщення запасу

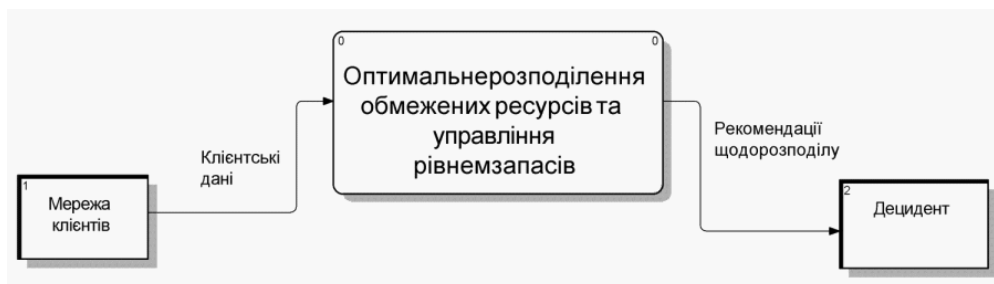


Рис. 5. Діаграма потоків даних першого рівня



Рис. 6. Діаграма потоків даних другого рівня

Взаємодію цієї СППР із зовнішнім середовищем зручно зобразити у вигляді DFD-діаграми за методологією Гайна–Сарсона. Наведемо діаграми потоків даних першого і другого рівнів для заданої СППР (див. рис. 5,6).

У потоці даних від клієнтів мережі до процесу розподілу ресурсів передається основна інформація про структуру мережі, кількість спожитих ресурсів (статистичні дані), вимоги на поповнення ресурсу. Від процесу розподілу до користувача передається інформація про рекомендовану структуру мережі, кількість ресурсу, яку потрібно надати кожному споживачу, маршрут доставки.

Діаграм потоків даних першого та другого рівнів достатньо для відображення основних процесів, які відбуваються в системі, та сховищ даних.

Висновки

1. На основі аналітичного огляду літератури та інших джерел подано формальні постановки і особливості задач дискретного програмування, розроблено класифікацію СППР за певними ознаками.

2. У результаті системного аналізу побудовано дерево цілей системи та виявлено можливі альтернативні варіанти побудови СППР. Результати системного аналізу сумісно з отриманими результатами опитування експертів дали змогу обрати оптимальний варіант побудови СППР, а саме – СППР з використанням комбінованих алгоритмів шляхом застосування методу аналітичної ієрархії.

3. Визначено структуру та функції основних модулів СППР, до складу якої входять: модуль оптимізації структури мережі; модуль розподілу ресурсів між споживачами; модуль мінімізації затрат на доставку ресурсу; модуль контролю процесу оптимізації; модуль взаємодії, модуль управління запасами.

4. Процеси функціонування СППР проілюстровані за допомогою діаграм потоків даних першого та другого рівнів. Описано модулі СППР та їхні функціональні особливості.

Перспектива розвитку даної СППР полягає у деталізації функцій системи, формування відповідних баз даних та знань і виконанні програмної реалізації.

1. *Модели и методы решения на ЭВМ комбинаторных задач оптимизации* / И.В. Сергиенко, М.Ф. Каспицкая. – К.: Наук. думка, 1981. – 288 с. 2. Сотник С.Л. Конспект лекций по курсу "Основы проектирования систем искусственного интеллекта". – 1997–1998. 3. Кофман А., Анри-Лабордер А. *Методы и модели исследования операций. Целочисленное программирование*. – Москва: Мир, 1977. – 432 с. 4. Сигал И.Х., Иванова А.П. *Введение в прикладное дискретное программирование: модели и вычислительные алгоритмы: Учеб. пособие*. – Изд. 2-е, испр. – М.: ФИЗМАТЛИТ, 2003. – 240с. 5. Косоруков О.А., Мищенко А.В. *Исследование операций: Учебник. Под общ. Н.П. Тихомирова*. – М.: Издательство "Экзамен", 2003. – 448 с. 6. Катренко А.В. *Дослідження операцій. Підручник*. – Львів: "Магнолія Плюс", 2007. – 549 с. 7. Haettenschwiler P. *Neues anwenderfreundliches Konzept der Entscheidungs-unterstützung. Gutes Entscheiden in Wirtschaft, Politik und Gesellschaft*. Zurich: Hochschulverlag AG, 1999. – S. 189–208. 8. Power D.J. *A Brief History of Decision Support Systems*. DSSResources.COM, World Wide Web, <http://DSSResources.COM/history/dsshistory.html>, version 2.8, May 31, 2003. 9. Power D. J. "What is a DSS?" // *The On-Line Executive Journal for Data-Intensive Decision Support*, 1997. – v. 1. – N3. 10. Alter S. L. *Decision support systems : current practice and continuing challenges*. Reading, Mass.: Addison-Wesley Pub., 1980. 11. Holsapple C.W., Winston A.B. *Decision Support Systems: A Knowledge-based Approach*. – Minneapolis: West Publishing Co., 1996. 12. Golden B., Hevner A., Power D.J. *Decision Insight Systems: A Critical Evaluation* // *Computers and Operations Research*, 1986. – v. 13. – N2/3. – p. 287–300. 13. Marakas G. M. *Decision support systems in the twenty-first century*. Upper Saddle River, N.J.: Prentice Hall, 1999. 14. Костюкова О.И. *Исследование операций: Учеб. пособие для студ. спец. 310304 «Информатика» всех форм обучения*. – Мн.: БГУИР, 2003. – 94с. 15. Кристофидес. *Теория графов. Алгоритмический подход*. – М.: Мир, 1978. – 432 с.