

**БАГАТОКРИТЕРІАЛЬНА ОПТИМІЗАЦІЯ
ПРИ ПРОЕКТУВАННІ СИСТЕМ
З ВИСОКОЮ ЗАВАНТАЖЕНОСТЮ**

Вступ. На сьогоднішній день сучасні програмні системи можна поділити на два класи. Перший клас, включає у себе програмні системи, які працюють і обробляють великі масиви даних їх ще називають системи навантажені даними. Другий клас включає у себе програмні системи, призначені для обчислення складних математичних задач – системи навантажені обчисленнями.

Поняття високої завантаженості – досить відносне, оскільки всі програмні системи специфічні й у залежності від своєї архітектури, структури можуть по різному реагувати на однакову кількість запитів. Відповідно навантаження також буде різним. Тому високе навантаження, визначається як навантаження, з яким дане апаратне забезпечення, з певних причин, не може справитись. Коли робота системи доходить до такої точки, є сенс починати масштабування і оптимізацію інфраструктури додатку. Щоб зрозуміти, що існує проблема перенавантаження, її потрібно діагностувати. Саме тому, при будь-якому навантаженні потрібна надійна система моніторингу, яка допоможе визначити момент, коли потрібно розпочати масштабування.

При створенні високонавантажених систем у першу чергу необхідно визначитися з тим, за допомогою яких інструментів і підходів найкраще вирішувати наявну проблему. Іноді бувають складності з підбором потрібної комбінації інструментів, коли потрібно реалізувати якийсь складний алгоритм, для цього одного інструмента окремо недостатньо.

При проектуванні інформаційної системи виникає безліч питань. Як забезпечити правильність і повноту даних, у тому числі при внутрішніх помилках? Як забезпечити однаково хорошу продуктивність для всіх клієнтів навіть у разі погіршення робочих характеристик деяких частин системи? Як забезпечити масштабування для обліку збільшеного навантаження?

Існує безліч факторів, що впливають на архітектуру інформаційної системи, включаючи навички та досвід залучених у проектування фахівців, успадковані системні залежності, терміни поставки, законодавчі обмеження і т. п. Ці фактори залежать від конкретної ситуації.

Сучасний етап розвитку науки і техніки характеризується значним ускладненням розв'язуваних завдань. Це призводить до необхідності скоротити терміни розробки і впровадження нових технологій. Системи управління вирішуються на основі багатокритеріальної оптимізації. Розроблено послідовність дій, що формалізують рішення задачі управління. Наведені кроки для побудови алгоритму багатокритеріальної оптимізації у проектуванні програмних систем, здійснює пошук рішень максимуму в багатокритеріальній задачі. Багатокритеріальна оптимізація ґрунтується на знаходженні рішення в задачах з великою кількістю варіантів.

Ключові слова: системи навантаженні даними, системи навантаженні обчисленнями.

Мета даної статті – надати математичний апарат для оптимізації всіх цих параметрів, адже за суттю перед нами постає багатокритеріальна задача з безліччю можливих сценаріїв яку можна вирішити за допомогою методів багатокритеріальної оптимізації.

Постановка задачі

Добре спроектована система повинна мати в собі такі якості.

Надійність [1]. Система повинна продовжувати працювати (здійснювати необхідні функції на необхідному рівні продуктивності) навіть при несприятливих обставинах (у разі апаратних або програмних збоїв, або помилок користувача). У програмному забезпеченні під надійністю зазвичай очікується наступне:

- додаток виконує очікувану користувачем функцію;
- додаток здатний витримати помилкові дії користувача;
- його продуктивність досить висока для поточного сценарія використання;
- система повинна запобігати будь-який несанкціонований доступ і неправильну експлуатацію.

Можливі проблеми називаються збоями [2], а системи, створені з розрахунку на них, називаються стійкими до збоїв. Збій зазвичай визначається як відхилення одного з компонентів системи від робочих характеристик, водночас як відмова – ситуація, коли вся система в цілому припиняє надання необхідного сервісу користувачеві. Надійність потрібна не тільки в керуючому програмному забезпеченні атомних електростанцій і повітряного сполучення, але й від звичайних додатків теж очікується надійна робота.

Масштабованість [3]. Система має передбачати розумні способи вирішення виникаючих проблем. Найчастіше проводять диференціацію між вертикальним масштабуванням – переходом на більш потужну апаратну частину і горизонтальним масштабуванням – розподілом навантаження за кількома системами об'єднаних в одну цілісну систему. Системи, які здатні працювати на окремій машині, звичайно простіші, а обчислювальні кластери призначені для складних обчислень можуть виявитися досить недешеві, так що при великому робочому навантаженні часто не вдається уникнути горизонтального масштабування. На практиці добре спроектована архітектура зазвичай являє собою прагматичну суміш цих підходів: наприклад, може виявитися простіше і дешевше використовувати кілька обчислювальних кластерів, ніж безліч маленьких віртуальних машин. Хоча розподіл сервісів без збереження стану на кілька систем особливих труднощів не представляє, перетворення інформаційних систем зі збереженням стану в одноузлові у розподілені може спричинити значні складності. Добре масштабована для конкретного додатка архітектура базується на таких припущеннях, які операції будуть виконуватися часто, а які – не часто, тобто на параметрах навантаження.

Зручність супроводу [4] забезпечує можливість ефективної роботи з системою кількох різних осіб. Зручність супроводу включає у себе:

- зручність експлуатації;
- простоту;
- можливість удосконалення.

З розвитком сучасних технологій будь-який WEB портал чи мобільний додаток можна вважати системою з високою завантаженістю, а якщо розглядати їх з точки зору моделювання математичних систем, то всіх їх можна вважати системами управління, які вирішуються за допомогою багатокритеріальної оптимізації.

В класичній постановці задачі математичного програмування, необхідно, в першу чергу, зазначити, що така задача передбачає тільки одну цільову функцію, яка кількісно визначена. Розглядаючи реальні системи, важливо розуміти, що на роль критерія оптимальності претендують кілька десятків показників. Так само, хорошою практикою є застосування декількох критеріїв синхронно, причому вони можуть бути взагалі несумісні, зокрема, вимога досягти максимальної

ефективності при мінімальних витратах з точки зору математичної постановки задачі є некоректною. Мінімальні витрати – це нульові витрати, які мають місце при повній відсутності будь-яких процесів. Подібно максимальна ефективність може бути досягнута тільки в разі використання ресурсів певних обсягів. Тому коректними є постановки завдань такого типу: досягти максимальної ефективності при заданих витратах або досягти заданого ефекту при мінімальних витратах.

Ефективне вирішення завдань управління стало допустимим завдяки розвитку засобів обчислювальної техніки, інформаційних технологій, методів системного аналізу, математичного моделювання та теорії оптимізації. Рішенням цих проблем, свого часу, займалися такі вчені, як: В.С. Михалевич, В.Л. Волкович [5], І.В. Сергієнко [6], О.Ф. Волошин, В.А. Заславський, І. Ушаков [7], В. Танаєв [8], В. Горелик [9], С.І. Наконечний, С.С. Савіна [10], Ю.М. Кузнецов, В.І. Кузубов, А.Б. Волощенко [11], А.М. Гупал, В.І. Норкин [12], Б. Муртаф [13].

На сьогоднішній день, в наукових працях висвітлюються різні аспекти проблем моделей, які вимагають прийняття оптимального рішення. При закономірності одиничного критерія оптимальності, пошук рішення здійснюється досить просто, так як явно є розроблений апарат методів оптимізації. Проблема вибору багатокритеріальної оптимізації у системах управління – складна і неоднозначна.

Підходячи до дослідження проблеми багатокритеріальної оптимізації систем управління, перш за все, потрібно розробити послідовність дій, які дозволять формалізувати і вирішити поставлену задачу.

Будемо вважати, що $\bar{F}_i = F_i(\alpha_1, \dots, \alpha_n)$ – критерії, за якими можна оцінити загальний стан системи, $(\alpha_1, \dots, \alpha_n)$ – множина керуючих дій. Тоді необхідно з множини вибрати такі дії, які за рахунок зміни можуть керувати системою. По закінченню цього етапу необхідно побудувати математичні моделі поведінки певних критеріїв, аналізуючи при цьому наявні в системі статистичні дані. По закінченню першого етапу, можна буде отримати набір цільових функцій $\bar{f}_i = f_i(\alpha_1, \dots, \alpha_n)$ з певним напрямом (максимізації або мінімізації), за яким вони повинні наближатися до своїх ідеальних значень $\bar{f}_i^* = f_i(\alpha_1, \dots, \alpha_n)$ в умовах, які розглядаються.

Після того, як отримано набір критеріїв оптимізації, необхідно побудувати загальну модель багатокритеріальної оптимізації або визначити процедуру, використання якої дозволить отримати найбільш ефективне або оптимальне рішення і визначити обмеження.

Рішення багатокритеріальної задачі також не представляє особливих складнощів, якщо критерії нейтральні по відношенню один до одного, тобто пошук рішення за одним критерієм ніяким чином не відбивається на пошук рішення за іншим критерієм. Наприклад, це має місце, коли при проектуванні системи ми ставимо собі за мету розробити таку систему, щоб вона була надійна і масштабована.

Загалом, складність вирішення багатокритеріальних задач полягає у тому, що критерії конкурують один з одним. У більшості практичних задач пошук більш пріоритетних рішень за одним критерієм призводить до того, що рішення стає менш привабливим за іншим критерієм. Наприклад, розглядаючи вартість розробки системи і масштабованість цієї системи як критерії при проектуванні, можна стверджувати, що при мініальному бюджеті на розробку (більш кращий за першим критерієм) система – менш масштабована (менш приваблива за другим критерієм). Аналіз таких ситуацій може бути здійснений за допомогою визначення множини Парето [14]. Припустимо, що при оцінці альтернатив використовувалися два критерії: – вартість і надійність.

З табл. 1 видно, що альтернатива 3 – найкраща, так як вона не гірша за інші альтернативи за всіма критеріями. Але треба зазначити, що дана альтернатива не підходить до критично важливих систем управління. Наприклад, дане рішення повністю задовольнить програмну систему за типом інформаційного порталу, але не задовольнить програмну систему, яка відповідає за охолодження реактора на атомній електростанції. Тому, що це взаємозалежні альтернативи і зрозуміло, що з малим бюджетом не вийде розробити повністю надійну систему, від якої залежить здоров'я людей.

Альтернатива α_i – домінуюча щодо альтернативи α_k , якщо за всіма критеріями оцінки альтернативи α_i не гірші, ніж альтернативи α_k , а хоча б за одним критерієм оцінка α_i краща. При цьому альтернатива α_k називається домінуючою. З визначення випливає, що альтернатива 3 з вищеведеного прикладу – домінуюча щодо альтернативи 1 і альтернативи 2. Це можна побачити з рисунку, де альтернатива 3 займає саме праве і верхнє положення щодо інших альтернатив.

ТАБЛИЦЯ 1. Значення критеріїв для двох альтернатив

Альтернативи	Вартість	Надійність
1	Мала	Низька
2	Велика	Висока
3	Мала	Висока

Розглянемо альтернативи 1 і 2. З рисунку також випливає, що альтернативи 1 і 2 не перебувають у домінуючому відношенні. Дійсно, за вартістю краще альтернатива 1, а за надійністю – альтернатива 2. Ці альтернативи є непорівняними щодо переваги між векторними оцінками, так як їх неможливо порівняти безпосередньо на основі критеріальних оцінок. Множина непорівняних альтернатив утворює множину ефективних рішень і називається множиною Парето, а альтернативи, що утворюють цю множину – Парето-оптимальними. Якщо повернутися до прикладу, то альтернативи 1 і 2 належать множині Парето.

Дати однозначну відповідь на питання, які з Парето оптимальних альтернатив слід вважати оптимальними, для загального випадку, не маючи додаткової інформації про критерії, неможливо. Загальна методика дослідження багатокритеріальних задач на основі математичного моделювання може бути реалізована в рамках одного з таких підходів.

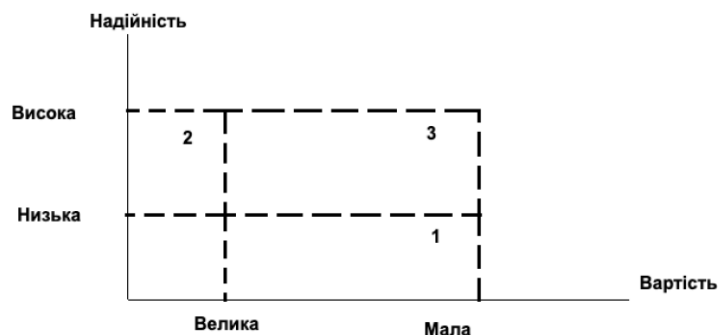


РИСУНОК. Подання альтернатив на графіку

Перший підхід. Для заданої багатокритеріальної задачі знаходиться множина Парето-оптимальних альтернатив, а вибір конкретної альтернативи з множини Парето-оптимальних надається спеціалісту.

Другий підхід. Проводиться звуження множини Парето-оптимальних альтернатив (в ідеалі – до однієї) за допомогою деяких формалізованих процедур, що полегшують остаточний вибір альтернативи для особи, яка приймає рішення. Зазначимо, що таке звуження може бути зроблено тільки за наявності додаткової інформації про критерії або про властивості оптимального рішення.

Розглянемо деякі найпростіші способи звуження Парето-оптимальної множини, акцентуючи при цьому увагу на необхідність додаткової інформації [15].

1. Визначити нижню межу критеріїв. При визначенні нижньої межі критеріїв оптимальною може вважатися тільки така Парето-оптимальна альтернатива, для якої оцінка за кожним з критеріїв не нижче визначеної оцінки. Це призводить до звуження Парето-оптимальної множини. При використанні цього способу остаточний вибір Парето-оптимальної альтернативи вибирається спеціалістом (на основі суб'єктивних міркувань).

2. Субоптимізація робиться у такий спосіб: виділяють один із критеріїв, а за всіма іншими критеріями визначають нижню межу. Оптимальним при цьому вважається результат, який максимізує виділений критерій на множині альтернатив, оцінки яких за іншими критеріями не нижче визначених. За допомогою методу субоптимізації задача багатокритеріальної оптимізації перетворюється в задачу скалярної оптимізації.

3. Лексикографічна оптимізація. Цей метод заснований на впорядкуванні критеріїв за їх відносною важливістю. Після цього процедуру знаходження оптимального рішення проводять наступним чином. На першому кроці відбирають результати, які мають максимальну оцінку. Якщо такий результат єдиний, то його і вважають оптимальним. Якщо ж таких випадків кілька, то серед них відбирають ті, які мають максимальну оцінку за наступними важливими критеріями і т. п. В результаті такої процедури завжди залишається (принаймні, в разі кінцевої множини випадків) єдиний результат, він і буде оптимальним. Термін «словниковий» відображає аналогію між цим методом і методом упорядкування слів у словнику. При лексикографічному підході потрібно ранжувати показники за важливістю, а значення показників розташовувати на шкалі порядку. Після того як найважливіший показник обраний, може бути визначена альтернатива, що має найвище значення за цим показником. Якщо така альтернатива одна, то її вибирають і процедура закінчується. Якщо за певним показником є кілька альтернатив з одним і тим же найвищим значенням, то вони порівнюються за наступним за важливістю показником. Процес триває до тих пір, поки не буде виявлена єдина альтернатива, або поки не будуть перевірені всі показники.

Розглянемо багатовимірну максимізацію, тобто знаходження максимуму функції більш ніж однієї незалежної змінної. Для досягнення поставленої задачі будемо використовувати Симплекс метод [16]. Метод вимагає лише оцінок функцій, а не похідних. Це не дуже ефективно з точки зору кількості необхідних обчислень функцій. Проте, Симплекс метод спуску часто може бути кращим методом для використання, так як він не вимагає великої кількості розрахунків.

Цей метод можна формалізувати у геометричному вигляді, тому його легко уявити у вигляді геометричної фігури. Симплекс – це геометрична фігура в N -мірному вимірі з $N + 1$ точок (або вершин), які з'єднуються прямими. У Декартовій системі координат симплекс – це трикутник. У тривимірному просторі це тетраедр, не обов'язково правильний тетраедр. Як правило, розглядають тільки невідроджені симплекси, тобто ті, які включають у себе кінцеву внутрішню N -мірну фігуру. Якщо будь-яка точка невідродженого симплекса береться за початок координат, то N інших точок визначають напрямки вектора, який охоплює N -мірний векторний простір.

У багатовимірному просторі, на відміну від одновимірного простору, ми не можемо відразу знайти максимум. Найкраще, що можна зробити для багатовимірної максимізації – це дати алгоритму початкову гіпотезу, тобто N -мірний вектор незалежних змін. Потім потрібно налаштувати алгоритм так, щоб він «спускався» нижче по N -мірній топографії до тих пір, поки не знайде локальний максимум. Алгоритм повинен починатися не просто з одної вершини, а з $N + 1$ вершин, що визначають початковий симплекс. Припустимо, що P_0 початкова відправна точка, тоді інші точки можна записати у такому вигляді:

$$P_i = P_0 + \Delta e_i,$$

де e_i – N -мірний вектор, Δ – просто константа, яка визначає розмірність задачі.

Алгоритм складається з ряду кроків, більшість з яких просто переміщує точку симплекса, де функція найбільша («найвища точка»), через протилежну сторону симплекса в нижню точку. Коли це можливо, метод повинен розширювати симплекс у тому чи іншому напрямку, роблячи більші кроки.

Критерії закінчення методу можуть бути слабкими в будь-якій процедурі багатовимірної максимізації. Зазвичай є сенс ідентифікувати один «цикл» або «крок» нашого багатовимірного алгоритму. Потім можна закінчити, коли векторна відстань, переміщена на цьому кроці, буде трохи меншою за величину, між деяким допуском. Як альтернативу ми могли б зробити так, щоб зменшення значення функції на завершальному етапі було трохи меншою деякого допуску.

Автором даної статі було розроблено комп'ютерну програму, яка використовує вищевикладені кроки для пошуку значення Z , яке максимізує рівняння $Z = X_1 + X_2 + 3X_3 - 0.5X_4$ з наступними обмеженнями:

$$\begin{aligned} X_1 + 2X_3 &\leq 740, \\ 2X_2 - 7X_4 &\leq 0, \\ X_2 - X_3 + 2X_4 &\geq 0.5, \\ X_1 + X_2 + X_3 + X_4 &= 9. \end{aligned}$$

Обчислення були виконані за підтримки обчислювального комплексу СКІТ Інституту кібернетики імені В.М. Глушкова НАН України [17]. Результати виконання програми наведені в табл. 2.

ТАБЛИЦЯ 2. Результат виконання програми

Z	X_1	X_2	X_3	X_4
17.025	0.0	3.325	4.725	0.95

Зазначені раніше кроки для побудови алгоритму багатовимірної максимізації дозволяють ефективно знайти оптимальне рішення, уникаючи простого перебору всіх можливих варіантів. Основний принцип метода полягає у тому, що обчислення починається з якогось «стартового» базисного рішення, а потім проводиться пошук варіантів, які покращують значення цільової функції. Це можливо лише у тому випадку, якщо зростання якоїсь змінної призведе до збільшення значення функції.

Висновки. При побудові високонавантаженої програмної системи виділяються деякі критерії, за яких система повинна функціонувати. Дані критерії можуть перетинатися, а в деяких випадках суперечити один одному і потрібно об'єднувати дані критерії у систему і оперувати ними. Дану систему можна розглядати як задачу з безліччю критеріїв, які потрібно оптимізувати. Виходячи

з проведеного дослідження, важливо зазначити, що системи управління засновані на багатокритеріальній оптимізації не мають універсального способу вирішення. А вибір і коректне застосування одного з методів багатокритеріальної оптимізації, які об'єднують до штучного злиття декількох обраних показників в один, залишається за суб'єктом прийняття рішень. Таким чином, рішення задачі управління за допомогою багатокритеріальної оптимізації полягає у забезпеченні необхідною кількістю науково обгрунтованої інформації, на підставі якої здійснюється вибір управлінського рішення.

Список літератури

1. Вікторова В.С., Степанянц А.С. Моделі і методи розрахунку надійності технічних систем. М.: Ле-Нанд, 2013. 256 с.
2. Шубинский І.Б. Структурна надійність інформаційних систем. М.: ТОВ «Журнал Надійність», 2012. 210 с.
3. Атчісон Л. Масштабування додатків. Пітер, 2017. 254 с.
4. Липа В.В. Супровід і управління конфігурацією складних програмних засобів. М.: СІН-ТЕГ, 2006. 357 с.
5. Михалевич В.С., Волкович В.Л. Обчислювальні методи дослідження і проектування складних систем. М.: 1982. 327 с.
6. Сергієнко І.В. Математичні моделі і методи розв'язання задач дискретної оптимізації. К.: Наук. думка, 1985. 384 с.
7. Волкович В.Л., Волошин О.Ф., Заславський В.А., Ушаков І. Моделі і алгоритми оптимізації надійності складних систем. К.: 1993. 423 с.
8. Танасв В. Декомпозиція та агрегування в задачах математичного програмування. М.: 1987. 523 с.
9. Горелик В.А., Ушаков І.А. Исследование операций. М.: Машиностроение, 1986. 288 с.
10. Наконечний С.І., Савіна С.С. Математичне програмування. К.: КНЕУ, 2003. 452 с.
11. Кузнецов Ю.Н., Кузубов В.И., Волощенко А.Б. Математическое программирование. М.: Высшая школа, 1980. 302 с.
12. Михалевич В.С., Гупал А.М., Норкин В.И. Методы невыпуклой оптимизации. М.: Наука, 1987. 282 с.
13. Муртаф Б. Современное линейное программирование. *Теория и практика*. М.: Мир, 1984. 224 с.
14. Халіна В.Г., Чернова Г.В. Системи підтримки прийняття рішень. М.: ЮРАЙТ, 2008. 478 с.
15. Ногін В.Д. Прийняття рішень в багатокритеріальному середовищі. М.: ФИЗМАТЛИТ, 2002. 142 с.
16. Струченков В.І. Методи оптимізації. Москва – Берлін: DirectMEDIA, 2015. 263 с.
17. Головинський А.Л., Маленко А.Л., Сергієнко І.В., Тульчинський В.Г. Енергоефективність супер-комп'ютера СКІТ-4. *Вісник НАН України*. 2013. № 2. С. 50–59. <https://doi.org/10.15407/vism2013.02.050>

Одержано 05.07.2021

Тупало Ярослав Олегович,
молодший науковий співробітник,
інженер програміст Інституту кібернетики імені В.М. Глушкова НАН України.
<https://orcid.org/0000-0003-3563-898X>
typaloyaroslav91@gmail.com

UDC 004.021

Yaroslav Tupalo

Multi-Criteria Optimization in the Design of High-Load Systems

V.M. Glushkov Institute of Cybernetics of the NAS of Ukraine, Kyiv
Correspondence: typaloyaroslav91@gmail.com

The current stage of development of science and technology is characterized by a significant complication of the tasks. The development of the economy to produce a situation where the development, implementation and operation of complex technical and socio-technical systems have to be in conditions of fierce competition. This is necessary to reduce the time of development and implementation of new technologies, especially in

high-load systems. Highly loaded systems are, by and large, the same websites, only with a very large audience, and as a consequence with a large load, which requires an optimized server part of the site. A qualitative characteristic for a highly loaded system is the bandwidth of this system, it describes the amount of work that must be able to perform the system per unit time. The development of highly loaded systems is time consuming and poorly formalized. Since the production of high-load systems is one of the most dynamically developing areas in the field of information technology, which is demonstrated by a significant annual increase in volumes. In the practical task of decision-making, there is often a situation where you cannot limit yourself to considering a single criterion for choosing a decision. An attempt at mathematical formalization of such problems has led to the creation of the Theory of Multicriteria Optimization, which is used in the development of methods, intended for support of decision makers, in the presence of several criteria. The steps for construction of algorithm of multicriteria optimization in designing of highly loaded systems, carries out search of solutions of a maximum in a multicriteria problem are resulted. The algorithm was based on the descent method in Simplex problems. Simplex method - an algorithm for solving the optimization problem of linear programming by searching the vertices of a convex polyhedron in multidimensional space. Multicriteria optimization is based on finding solutions in problems with a large number of options. Now the type of tasks is very resource-intensive and is calculated using computers.

Keywords: high-load data systems, high-load computing system.