

Вінницький національний технічний університет

(повне найменування вищого навчального закладу)

Факультет інформаційних технологій і комп'ютерної інженерії

(повне найменування інституту, назва факультету (відділення))

Кафедра обчислювальної техніки

(повна назва кафедри (предметної, циклової комісії))

Пояснювальна записка
до магістерської кваліфікаційної роботи
магістр
(освітньо-кваліфікаційний рівень)
на тему: Засоби аналізу ефективності моделювання мультипроцесорних
систем з розподіленою пам'яттю

Виконав: студент 2 курсу групи 1КІ-18м
спеціальності 123 «Комп'ютерна інженерія»

(шифр і назва напрямку підготовки)

_____Волосович О.С._____

(прізвище та ініціали)

Керівник: к.т.н., доц. Колесник І.С.

(прізвище та ініціали)

Рецензент: д.т.н, проф.Поплавський А.В.

(прізвище та ініціали)

Вінниця ВНТУ – 2019 року

АНОТАЦІЯ

Робота присвячена розробці і дослідженню засобів аналізу ефективності моделювання мультипроцесорних систем з розподіленою пам'ятю. Проаналізовано еволюцію змісту термінів «мультипроцесорність» і «розподілена пам'ять», вибрано як базову модель для оцінки ефективності комп'ютерної системи структуру з типового виробництва і типовою комп'ютерною системою, але з моделлю управління на базі методів оптимального агрегування як альтернативі стандартних систем на базі оптимального агрегування. Розроблено комп'ютерну інтегровану систему оптимального адаптивного управління для об'єкту в цілому і підсистем, що є концептуально розподіленою з інформаційного забезпечення і управління. Розроблені моделі імітаційного моделювання оптимальної та безпошукової адаптивної виробничої системи і модулі оцінювання комплексної ефективності. Наведені приклади тестування ефективності системи.

ABSTRACT

The work is dedicated to the exploration and development of new mathematical models of integrated process operation and development in modern production systems. Developed integrated system monitors the status and settings of subsystems of the production system. Complex measures the State of subsystems is complemented by nonlinear observer status and settings. Availability of information redundancy allows you to build analytical non-searching adaptive control system of production system. This system can provide optimal reconfiguration manufacturing system with subsystems failures. Production management system based on binary tree optimal aggregation consists of identical blocks: strategic management, operational management, information-measuring system, model-Predictor. Examples of modeling.

ЗМІСТ

ВСТУП.....	8
1 АНАЛІЗ МУЛЬТИПРОЦЕСОРНИХ СИСТЕМ ТА ПОКАЗНИКІВ ЇХ ЕФЕКТИВНОСТІ.....	11
1.1 Огляд математичних моделей і методів для мультипроцесорних систем з розподіленою пам'яттю.....	12
1.2 Багатопроцесорні системи.....	14
1.3 Показники ефективності мультипроцесорних систем з розподіленою пам'яттю.....	23
1.4 Існуючі моделі і засоби оцінки ефективності комп'ютерних систем.....	27
1.5 Висновки до розділу.....	28
2 УДОСКОНАЛЕННЯ КОМПЛЕКСУ МОДЕЛЕЙ І ПРОГРАМ ДЛЯ ОЦІНКИ ЕФЕКТИВНОСТІ МУЛЬТИПРОЦЕСОРНОЇ СИСТЕМИ.....	30
2.1. Аналіз структур і задач управління виробничих систем.....	30
2.2 Моделі процесів утворення розподілених виробничих систем.....	35
2.3 Постановка задачі моделювання розподілених систем класу «композиційна структура».....	36
2.4 Задачі моделювання розподілених систем класу «декомпозиція».....	39
2.5 Декомпозиційна структура для задач класу «декомпозиція».....	42
2.6. Синтез декомпозиційної структури для систем К-класу і Д-класу.....	45
2.7 Висновки до розділу.....	50
3 ПРАКТИЧНА РЕАЛІЗАЦІЯ ЗАСОБІВ АНАЛІЗУ ЕФЕКТИВНОСТІ МУЛЬТИПРОЦЕСОРНИХ СИСТЕМ.....	51
3.1 Структура і функції засобів для тестування і оцінки ефективності мультипроцесорних систем з розподіленою пам'яттю.....	51
3.2 Оптимальне агрегування виробничої системи з урахуванням впливу комп'ютерних систем.....	54
3.3 Моделі оцінювання ефективності комп'ютерних систем з урахуванням розвитку та інновацій.....	62
3.4 Аналіз результатів моделювання процесів розвитку системи обслуговування.....	72
4 ЕКОНОМІЧНА ЧАСТИНА.....	80
4.1 Оцінювання комерційного потенціалу розробки.....	80
4.2 Прогнозування витрат на виконання науково-дослідної, дослідно-конструкторської та конструкторсько-технологічної роботи.....	86
4.3 Прогнозування комерційних ефектів від реалізації результатів розробки.....	91
4.4 Розрахунок ефективності вкладених інвестицій та період їх окупності.....	93
Висновок до розділу.....	97
ВИСНОВКИ.....	98
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	99
Додаток А.....	101
Додаток Б.....	105
Додаток В.....	106

Додаток Д.	107
Додаток Е.	109
Додаток Ж.	110
Додаток К.	111
Додаток Л.	112

ВСТУП

Актуальність теми. Сьогодні в епоху виокремлених технологій в усіх областях науки і техніки вводяться комп'ютерні системи – для вбудовування інформаційно-вимірювальних модулів, комплексні розподілені системи контролю середовища та ін.. до управління атомними електростанціями, металургійними агрегатами та ін.. На початку комп'ютерної епохи вважалося, що комп'ютери спроможні автоматизувати все. На цьому фоні незрозумілими є ситуації, коли причиною малих в дуже великих відмов є програмне забезпечення комп'ютерів. Найбільш відомі завдяки засобам інформації, аварії літаків: інтелектуальне програмне забезпечення «перемогало» зусилля пілота і розбивало лайнери на 200 пасажирів. Супернадійний масовий винищувач може виконувати свої функції годину на добу, а основний час витрачає на перевірки, тестування, перезавантажування КС. Власне комп'ютерні засоби можуть працювати годинами без збоїв. Визнана причина – неякісні і неадекватні програмні продукти для технічних систем. Ще одна причина – недостатній захист комп'ютерних систем від зламів: сьогодні сьогодні можливі за допомогою відповідних програм і засобів можливо посадити лайнер, звести з орбіти супутник.

В рамках даної роботи планується розробити обчислювально ефективні і математичні математичні моделі для комп'ютерних систем і оцінки їх ефективності. Така можливість гарантована напрацюваннями в методах оптимального агрегування. Друге радикальне проектне рішення – побудова «цифрової копії об'єкта» - так називають імітаційну модель об'єкта управління.

Очевидно потрібен раціональний підхід до виробництва дійсно потрібних суспільству матеріальних та інформаційних продуктів. Чисто емпіричні паростки раціонального підходу виникли на мегазаводах Кореї, Японії і Німеччині. Тільки в останні роки на базі цього емпіричного досвіду

почали виникати нові підходи до раціонального управління сучасними виробництвами в сучасних умовах.

Тобто потрібно вирішити задачу оцінювання ефективності мультипроцесорної системи з розподіленою пам'яттю. Сьогодні комп'ютери стали невідемними складовими технічних та інформаційних систем. Тобто, при оцінюванні КС слід враховувати функції КС в складі об'єкта, що саме «обслуговується» даною КС.

В рамках аналітичних методів не існує задовільного рішення задачі», однак реальні успіхи реального моделювання породжують напрям «моделеорієнтоване проектування».

Мета і задачі дослідження Метою роботи є формування, обґрунтування і реалізація системи оцінок мультипроцесорних КС узгоджених з критеріями верхніх рівнів – технологічних, екологічних, соціально-освітніх.

Відповідно до поставленої мети в роботі вирішуються такі задачі:

- розробка моделі для тестування мультипроцесорної системи управління виробничою системою;
- перевірка ефективності та коректності програмного забезпечення і розробка методики його використання.

Об'єктом дослідження є процеси функціонування і розвитку виробничої системи і мультипроцесорної комп'ютерної систем управління виробництвом.

Предметом дослідження є комплексні показники ефективності системи : «виробництво + комп'ютерна система управління».

Результаты виконання роботи. Комплекс моделей и програмних модулів для оцінки ефективності мультипроцесорної при управління типовими об'єктами, призначений для прийняття рішень стосовно об'єкту і комп'ютерної системи.

Наукову новизну магістерської роботи становлять:

- покращено математичну модель і програму модуля для аналізу

чутливості показників ефективності виробничої системи, де на відміну від існуючих моделей аналізу чутливості використовується структура даних «бинарне дерево оптимального агрегування», що зводить обчислення функції впливу довільної підсистеми на систему в цілому до алгебраїчного рівняння, на відміну від аналогів, де потрібні обчислення нелінійних рівнянь, що підвищує ефективність в цілому.

– покращено математичну модель і програму модуля для аналізу чутливості до відмов і відмовостійкості виробничої системи, де на відміну від існуючих моделей аналізу чутливості до відмов використовується оптимальне агрегування виробничої системи з певною конфігурацією відмов, обчислення втрат зводиться до простої операції порівняння, а головне, обчислюється розподіл навантаження з мінімальними втратами, на відміну від аналогів, де оптимально реконфігурація неможлива.

Практичне значення одержаних результатів. Результати теоретичних досліджень програмно реалізовані програмними модулями для розрахунку і аналізу показників тестування.

Апробація результатів дисертації. Основні положення магістерської роботи доповідалися та обговорювалися на таких конференціях і семінарах: - Всеукраїнська Науково-практична конференція: Молодь в науці: дослідження, проблеми, перспективи 2019 року;

За тематикою проекту отримано авторське свідоцтво на програму «Оптимальне управління виробництвом на базі оптимального агрегування» авторів Боровська Т.М., Волосович О.Ю., Колесник І.С., Недоснований О.Ю» .

1. АНАЛІЗ МУЛЬТИПРОЦЕСОРНИХ СИСТЕМ ТА ПОКАЗНИКІВ ЇХ ЕФЕКТИВНОСТІ

Згідно стандартам науки наукове дослідження повинно починатись з аналізу аналогів і прототипів. Аналогічний стандарт існує в техніці для обґрунтування проектів, в патентному пошуку. Однак сучасна наука суттєво комп'ютеризована. Продуктивний науковий напрям сьогодні повинен мати певне програмне забезпечення для вирішення і дослідження теоретичних задач.

Програмне забезпечення для реального управління реальними виробничими системами повинно бути відмовостійким а не тільки ефективним а такожі безпечним. Крім того, у розробників і користувачів завжди є побажання: до програмної системи – забезпечувати оптимальне адаптивне управління об'єктом в цілому і всіма підсистемами без використання унікальних суперкомп'ютерів. При розробці системи управління для певного об'єкту виробництва, обслуговування, торгівлі пріоритет повинен віддаватись математичним моделям і програмам.

Недосконалість компоненту «моделі» не може бути скомпенсована ефективністю компоненту «комп'ютерна система». Компонент «моделі» КС – власне об'єкт: система виробництва, обслуговування логістики, розвитку. Сучасні великі системи можуть мати в своєму складі – підсистеми виробництва, обслуговування. Упішні системи розвиваються – створюють власні підсистеми обслуговування, розвитку, та інші функціональні підрозділи. Таким чином сучасна доступних комп'ютерних системах, для аналізу і синтезу управління і єдиним знайденим, це суттєво спрощує вибір. Далі подано предметний аналіз вибраних для вирішення поставлених задач і завдань методів.

1.1 Огляд математичних моделей і методів для мультипроцесорних систем з розподіленою пам'яттю

Детально розглядаємо цю затверджену стандартами технологію програмування таких систем, для того щоб запропонувати альтернативу: структура і функції МОС повинні визначатися умовами функціонування і розвитку сучасних об'єктів - виробничих систем та систем масового обслуговування. В монографії [23] подано стислий огляд архітектур мультипроцесорних обчислювальних систем (МОС) та інструментів програмування для них. Детально розглянуті технології, такі як OpenMP, яке стало стандартом для програмування систем зі спільною пам'яттю та комунікаційним інтерфейсом MPI спілкування, призначеним для МОС програмування розподіленої пам'яті. Поданий матеріал ілюструється прикладами. Крім того, розглядається операційне середовище, розробки і виконання паралельно програм на МОС і технології віддаленого доступу до режиму обчислювальних ресурсів центри для колективного використання. На закінчення, описано використання паралельних бібліотек підпрограм для розподілених пам'яті МОС ScaLAPACK та Aztec.

Системи, з розподіленою пам'яттю з самого початку конструювання таких систем пам'ять була побудована як система, призначена для подолання недоліків властивих в системах зі спільною пам'яттю. Було запропоновано побудувати мультипроцесорні обчислювальні системи як сукупність незалежних обчислювальних вузлів, що складається з локалізованої оперативної пам'яті, процесора і комунікаційного обладнання для спілкування між вузлами.

Така обчислювальна архітектура знімає проблеми, пов'язані з пропускнуою здатністю та обсягом пам'яті, потрібної для підтримки кеш когерентності усіх процесорів. Це значно розширяє можливості для збільшення кількості процесорів комп'ютера, яка зараз поки обмежена тільки потенціалом обладнання зв'язку.

Для систем MPP на передній план проблему ефективності комунікації середовища. Різні виробники використовують різні топології. Процесори Intel Paragon комп'ютерів утворюють двовимірну прямокутну сітку. Для цього, кожен вузол повинен мати чотири канали зв'язку. В компютерах Cray T3D/T3E використовувалась топологія 3D тора [26]. Для організації такої топології у вузлі має бути шість каналів зв'язку. При таких топологіях, якщо вимагається передача даних між не суміжними вузлами, які не є найближчими сусідами, необхідна трансляція повідомлення через проміжні вузли. Для великих обчислювальних систем обсяг посередницьких трансляцій стає непомірно високим, і це неминуче приводить до зменшення продуктивності комунікаційного середовища. У комп'ютерах nCUBE, для зменшення проміжних передач використовуються топології n-мірного куба (з максимальною технічно реалізованою розмірністю $n = 13$). Кожен вузол мав 13 каналів зв'язку, що дозволяє створити комп'ютерні системи, що містять до 8192 процесорів. Максимальна кількість трансляцій була тільки 13. [26].

В даний час функціонують комп'ютерні системи з розподіленою пам'яті, що складається з декількох сотень тисяч процесорів (IBM Blue Gene/L). Коли побудови таких систем за допомогою високошвидкісних перемикачів, пов'язаних в топології (товстого дерева). Спочатку системи з розподіленою пам'яттю з'явилися як дорогі, спеціалізовані обчислювальні системи. Проте, з появою недорогого устаткування високошвидкісного зв'язку, комп'ютерна система з розподіленою пам'яттю почала створюватись на основі стандартних системних блоків. Це значно скоротило витрати на багатопроцесорні системи і зробило їх загально доступними. Такі системи називаються обчислювальні кластери. Закінчені рішення, пропоновані сьогодні практично усіма виробниками комп'ютерного обладнання. Відсутність спільної пам'яті поліпшила масштабованість обчислювальних систем, але значно ускладнила взаємодії між процесорами і, отже, програмування для них. Як правило, кожен вузол діє на своїй копії цієї операційної системи, яка проходить окрему копію паралельних програми.

Зв'язок між вузлами робиться за допомогою механізму передача повідомлення. Цей механізм реалізований у вигляді бібліотеки додатків, (приладних програм – приложень) які повинні підключатися до паралельних програм. Кожен виробник багатопроцесорних систем з розподіленою пам'ятю надає такі бібліотеки в рамках системного програмного забезпечення на своїх комп'ютерах.

1. 2 Багатопроцесорні системи

Класифікація систем для паралельної обробки даних. Протягом всієї історії комп'ютерних розвитку спроби знайти якийсь загальний класифікації, під які підпадають всі можливі напрямки розвитку апаратних архітектур. Жодна з цих класифікації може не охоплюють різноманітні нові архітектурні рішення і не витримати випробування часом. Будь-якою комп'ютерна система (чи це супер комп'ютер чи персональний комп'ютер) досягає своєї максимальної продуктивності завдяки використанню високошвидкісних елементів і одночасне виконання великої кількості операцій. Можливістю паралельної роботи різноманітних системи пристроїв (перекриття) є основою для прискорення основних операцій.

Паралельні ЦОМ часто класифікуються відповідно до класифікації Флінн на машинах типу SIMD (одномісний інструкція кілька даних-з одного потоку команд з кількома потік даних) і MIMD (кілька інструкція кілька дані потік з кількох команд з кілька потік даних). Як і будь-яка інша, вищезгадана класифікація є специфічною, дещо особистнісною та залежить від досвіду, точки зору та мети класифікатора: є машини, які не отримує права перебувати в ній, є також важливі ознаки, які ця класифікація не враховує. Зокрема, на машинах , які називають SIMD на векторних процесорах, хоча їх висока продуктивність залежить від іншої форми - паралелізму конвеєру машиної організації. Класифікація Флінна не диференціює інші важливі особливості обчислювальних моделей, наприклад, функції на рівні

"зернистості" паралельних обчислень і методи синхронізації. Існують чотири основних типи паралельної обробки з такими системними архітектурами:

1) Конвеєрна і векторна обробка. Конвеєр є основою роздільного виконання деяких дій у декілька етапів (декількох кроків) під час кожного етапу виконується певна дія (мікрооперація) з передачею даних з однієї фази до наступної. Виконання пов'язане з тим, що в архітектурі ЕОМ використовуються одні і ті ж ресурси для вирішення різних завдань, причому якщо ці завдання тотожні, то це послідовний конвеєр, якщо завдання однакові - векторний конвеєр.

Застосування конвеєру підвищує продуктивність, оскільки одночасно дозволяє на різних етапах проходження конвеєра виконати декілька операцій паралельно. Конвеєрна обробка використовується лише коли завантаження конвеєра недалеко від повного і швидкість роботи нових операторів відповідає максимальній продуктивності конвеєра. Якщо є затримки на будь-якій фазі обробки, швидкість виконання всієї задачі зменшується, виникають простоя ресурсів обладнання. Векторні операції надають ідеальну можливість повністю завантажити обчислювальні потужності конвеєру.

При виконанні векторних команд одна і та ж операція відноситься до складових вектора (або частіше до відповідних елементів пари векторів). При налаштуванні конвеєра для виконання конкретної операції можуть бути виконані і деякі «налаштування», а потім операнди будуть проходити конвеєр з масимальною швидкістю, що допускається пам'ятю. Такі операції у вихідній програмі відповідають невеликим компактним циклом.

2) Машини типу SIMD. Машини SIMD типу складаються з великої кількості ідентичних процесорів обробки елементів, що мають свою власну пам'ять. Всі елементи процесорів в такій машині опрацьовують одну програму. Це очевидно, що машина, складається з великої кількості процесорів, які можуть забезпечити дуже високу продуктивність виконання лише тих завдань, в яких всі процесори можуть робити ту ж роботу. Модель обчислень SIMD машини дуже схожа на модель обчислень векторного

процесора: одна і та ж операція виконуються на великому блоці даних. Обробки елементів таких процесорів універсального програмованого комп'ютера підстроюватимуться паралельно, можуть бути досить складними і містити розгалуження. SIMD -системи зазвичай використовуються для конкретних завдань, що вимагають, як правило, не стільки гнучкості і універсальності обчислювальної машини, скільки самої обчислювальної потужності. Обробка медіа, наукові дослідження (ті ж симуляції і моделювання), або, наприклад, трансформації Фур'є гігантських матриць. Тому в графічних юнітах така велика кількість ядер: це SIMD-системи, де по-справжньому “розумний” процесор (такий, як у вашому комп'ютері) як правило один: він управляє багатьма простішими процесорами

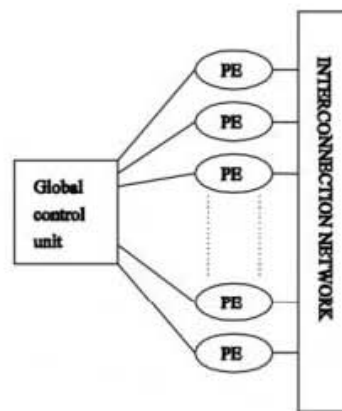


Рисунок 1.1 – Архітектура SIMD-системи

Так як “управляючий” процесор відправляє одні і ті ж інструкції всім “робочим” процесорам, програмування таких систем потребує досвіду та логічного мислення від програміста.

Приклади SIMD-машин - це старі ILLiac IV, MPP, DAP, Connection Machine CM-1/2, сучасні векторні юніти, специфічні співпроцесори і графічні юніти типу nVidia GPU.

3) Машини типу MIMD. MIMD-машини володіють більш широким функціоналом, тому в наших користувацьких комп'ютерах використовуються саме вони. Якщо (для прикладу) у вас хоча бы

двохядерний процесор в пристрої, вважайте, що ви користуєтесь MIMD-машиною із загальною пам'яттю. MIMD розподіленою пам'яттю володіють суперкомп'ютери типу IBM Blue Gene, чи кластери. Термін "мультипроцесор" охоплює більшість машин типу MIMD. В мультипроцесорній системі кожен процесорний елемент (PE) виконує свою програму досить незалежно від інших елементів. Процесорні елементи, звичайно, повинні якимось спілкуватися один з одним, що робить необхідні більш детальну класифікація машин типу MIMD. У багатопроцесорні Загальні оперативної пам'яті (silnosvjazannuh багатопроцесорні) доступна пам'яті даних і команд всі PE.

Базова модель обчислень на MIMD мультипроцесорі є сукупністю незалежних процесів, які епізодично звертаючись до розподілених даних. Існує цілий спектр варіантів цієї моделі. З одного краю спектра - модель розподілених обчислень, в якій програма ділиться велику кількість паралельних задач, складених з множин даних та множини підпрограм

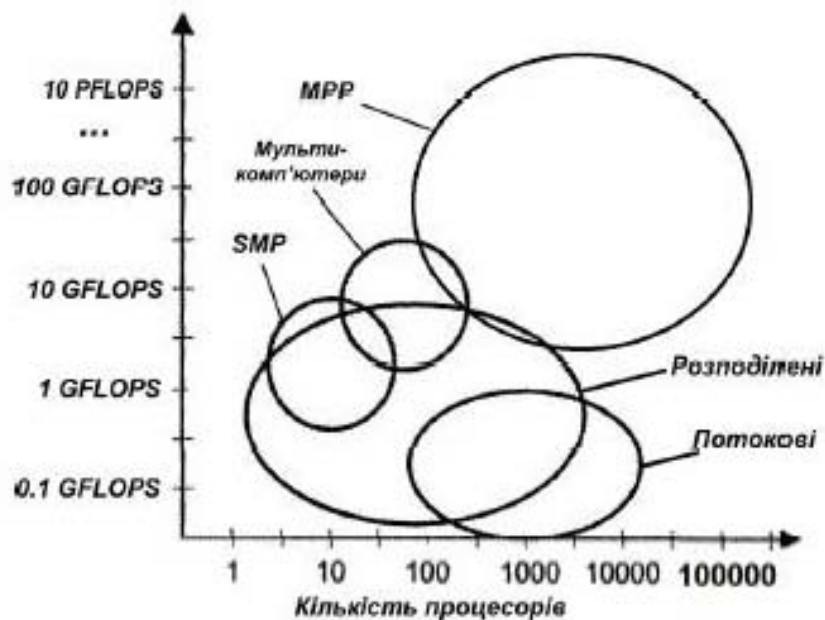


Рисунок 1.2 – Продуктивність MIMD-систем, як функція їх типу і кількості процесорів

На другому кінці спектру - модель поточкових обчислень, в яких кожна операція в програмі може розглядатись як окремий процес. При їх отриманні

операція виконується, і отримане значення передається тим процесам, яким воно потрібне. В потокових моделях обчислень з великим і середнім рівнями гранулярності, процеси мають велике число операцій і виконуються потоково.

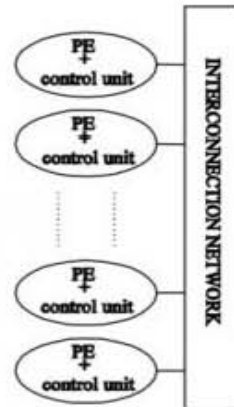


Рисунок 1.3 – Архітектура MIMD-системи

Симетрична мультипроцесорна система (SMP, Symmetric Multiprocessor) – це обчислювальна система, що володіє наступними характеристиками:

- є два чи більше процесорів, в яких однакові або майже однакові потужності;
- процесори сумісно використовують основну пам'ять та працюють в єдиному віртуальному і фізичному адресному просторі;
- всі процесори зв'язані між собою за допомогою шини чи за іншою схемою, так що час доступу до пам'яті для кожного з них однаковий;
- всі процесори розділяють доступ до пристроїв вивода або через одні і ті ж канали, або через різні канали, що забезпечують доступ до одного і того ж зовнішнього пристрою;
- всі процесори мають можливість виконувати однакові функції;
- будь-який з процесорів може обслуговувати зовнішні переривання;

Обчислювальна система керується інтегрованою операційною системою, яка організує і координує взаємодію між процесорами і програмами на рівні завдань, файлів і елементів даних. На відміну від слабо зв'язаних мультипроцесорних систем (кластерів), де як фізична одиниця обміну інформацією виступає повідомлення або повний файл, в SMP допустима взаємодія на рівні окремого елемента даних. Завдяки цьому досягається високий рівень зв'язності між процесорами. Хоча технічно SMP-системи симетричні, в їх роботі присутній невеликий чинник перекосу, який вносить програмне забезпечення. На час завантаження системи один з процесорів отримує статус ведучого (master). Це не означає, що пізніше, під час роботи якісь процесори будуть такими, які керуються – всі вони в SMP-системі рівноправні. Термін «керуючий» відноситься до того процесору, який за замовчуванням керуватиме первинним завантаженням ОС. Операційна система планує процеси або потоки процесів (threads) відразу по всіх процесорах, приховуючи при цьому від користувача характер SMP-архітектури. В порівнянні з однопроцесорними схемами SMP-системи мають перевагу за наступними показниками:

- продуктивність - якщо завдання, яке потрібно вирішити піддається розбиттю на декілька частин так, що окремі частини можуть виконуватися паралельно, то множина процесорів дає вигоду в продуктивності відносно одиночного процесора того ж типу
- готовність - у симетричному мультипроцесорі відмова одного з компонентів не призводить до відмови системи, оскільки будь-який з процесорів в стані виконувати ті ж функції, що і інші.

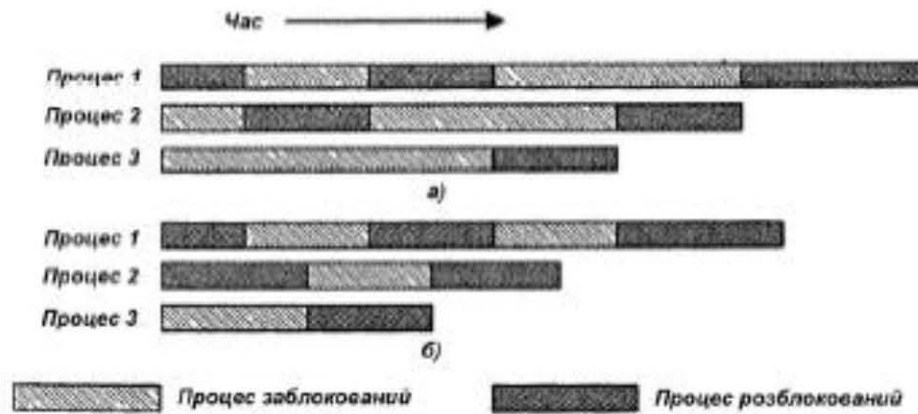


Рисунок 1.4 – Мультипрограмування і мультипроцесорна обробка

- розширюваність - продуктивність системи може бути збільшена додаванням додаткових процесорів.
- масштабованість - варіюючи число процесорів в системі, можна створити системи різної продуктивності і вартості.

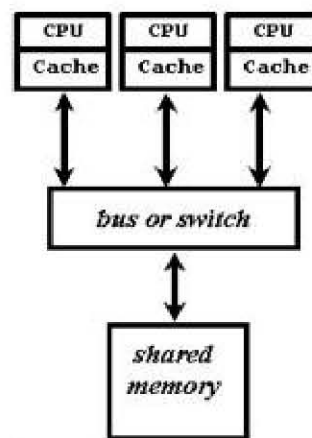


Рисунок 1.5 – Архітектура SMP – системи

4) Мультипроцесорні машини з SIMD процесорами. Багато сучасних супер-ЕОМ представляють собою багатопроцесорні системи, складені з векторних процесорів або процесори типу SIMD. Такі машини відносять до класу MSIMD. Мови програмування і їх відповідні компілятори для машин MSIMD «зазвичай» надають конструкції мови, які дозволяють програмісту описати "дрібнозернистий" паралелізм. В рамках кожного завдання

компілятор автоматично слідкує за відповідними циклами. Тип машини: векторні операції ("детальний" паралелізм) ті частини програми, прийнятні для цієї мети, гнучкі MIMD архітектури для інших частин програми.

Перспективність таких архітектур обумовлена двома факторами:.

1. MIMD архітектура забезпечує велику гнучкість: адекватної підтримки апаратного та програмного забезпечення може функціонувати як MIMD автономні системи забезпечення високої продуктивності обробки даних для одного застосування в задачі, як багатопроцесорна машина, яка виконує множину завдань паралельно, для якої можливі подібні комбінації.

2. Архітектура MIMD може використовувати всі переваги сучасних мікропроцесорів. Технологія заснована на суворому обліці співвідношення вартість/продуктивність. По суті, майже всі сучасні мультипроцесорні системи будуються на тому ж мікропроцесорах, які можуть бути знайдені в персональних комп'ютерах, робочих станціях і невеликих однопроцесорних серверах.

В мережах з комутацією каналів і в мережах з комутацією пакетів при зростанні вимог до обміну слід урахувувати можливості перенавантаження мережі. Тут міжпроцесорний обмін зв'язує мережеві ресурси: канали, процесори, буфери повідомлень. Обсяг інформації, яка передається може бути скорочений за рахунок ретельної функціональної декомпозиції задачі і ретельного диспетчерування функцій, виконується. Таким чином, існуючі MIMD машини поділяються на два основних класи, залежно від кількості об'єднаних процесорів, який визначає і спосіб організації пам'яті і методикку їх взаємозв'язків.

До першої групи належать машини з спільною оперативною пам'яттю, в результаті чого разом до декількох десятків (зазвичай менш ніж 32) процесори. Відносно невелике число процесорів такі машини дозволяють мати одну централізовану спільну пам'ять і об'єднувати процесори і пам'ять в одну шину.

Другу групу машин складають великомасштабні системи з розподіленою пам'ятю. Для того щоб підтримувати більшу кількість процесорів необхідно розподіляти основну пам'ять між ними, інакше пропускна здатність до пам'яті просто не може бути достатньою, щоб задовольнити вимоги, виходячи з дуже великої кількості процесорів. Підведемо підсумки аналізу розподіленої пам'яті:

- сумісність з зрозумілими як в однопроцесорних, так і маломасштабованих багатопроцесорних системах, механізмами, що використовують для обміну загальну пам'ять;
- простота програмування, коли моделі обміну між процесорами складні або динамічні при виконанні. Подібні переваги спрощують конструювання компілятора;
- менша затримка обміну і краще використання смуги пропускання при обміні малими порціями даних;
- можливість використання апаратно керованого кешування для зниження частоти віддаленого обміну, що допускає кешування усіх даних, як розділяємих, так і нерозділяємих.

Основні переваги обміну, передаючи повідомлення:

- обладнання може бути більш простим, особливо в порівнянні з моделлю зі спільною пам'ятю, яка підтримує масштабовану когерентність кеш-пам'яті;
- моделі обміну тут зрозумілі і цим заохочують програмістів (або компілятор) приділяти увагу процесам обміну, що звичайно потребує збільшення обчислювальних витрат.

Побудова механізмів для реалізації спільної пам'яті над повідомлення передавального механізму є набагато складнішою. Без очікуваної апаратної підтримки усіх звернень до спільної пам'яті вимагається залучення операційної системи для переобчислення адрес та захисту пам'яті в відправці і отриманні і повідомлень. Тому що завантаження і відображення операцій

звичайно працюють з невеликою кількістю даних, великі накладні витрати для підтримки таких обмінів роблять неможливим для них чисто програмну реалізацію. Тобто існуючі моделі і методи роботи з існуючою реалізацією розподіленої пам'яті суттєво зменшують її ефективність. Нарешті в літературі не знайдено постановки задач і методів забезпечення живучості: як система буде функціонувати при відмовах певного мікропроцесора і відповідної пам'яті.

1.3 Показники ефективності мультипроцесорних систем з розподіленою пам'яттю

При оцінці любого механізму обміну в розподіленій пам'яті критичними є три характеристики продуктивності:

1. Полоса пропускання: в ідеалі полоса пропускання механізму обміну будет обмежена полосами пропускання процесора, пам'яті і системі міжпоеднань, а не якимись там аспектами механізму обміну. Зв'язані з механізмом обміну накладні витрати (наприклад, довжина міжпроцесорного зв'язку) прямо діють на полосу пропускання.

2. Затримка: в ідеалі затримка повинна бути настільки мала, наскільки це можливо. Для її визначення критичні «накладні» витрати апаратури і програмного забезпечення, пов'язані з ініціюванням и завершенням обміну.

3. Приховування затримки: наскільки повно механізм приховує затримку шляхом перекриття обміну обчислюваннями та іншими обмінами.

Кожний з цих показників продуктивності діє на характеристики обміну в пам'яті. Зокрема, затримка і полоса пропускання можуть змінюватися в залежності від розміру елемента даних. В загальному випадку, механізм, який однаково добре працює з малими і великими обсягами даних буде більш гнучким і ефективним. Тому відмінності різних ЦОМ з розподіленою пам'яттю визначаються моделями пам'яті і механізмом обміну. Історично машини з розподіленою пам'яттю були спочатку побудовані, використовуючи механізм передачі повідомлень, оскільки вона була явно легше, і багато розробників і дослідників не вірили, що єдиний адресний простір, може бути

побудованим. Останнім часом модель обміну зі спільною пам'яттю дійсно застосовується в практично всіх розвинених машинах. В той час як централізована спільна пам'ять машин, на основі загальної шини ще домінує на комп'ютерному ринку, довгострокові технічні тенденції спрямовані на використання розподіленої пам'яті.

Основи реалізації. Ключовий момент реалізацій в багатопроцесорних системах з невеликим числом процесорів як скасування записи і записи з оновленням полягає у використанні механізму шини для виконання цих операцій. Виконуючи операції оновлення або відкликання процесор просто захоплює шини і передає його на адресу в яких слід оновлювати або скасування даних. Всі процесори постійно контролюють шину, контролюючи дані, які з'являються на її адресу. Процесори перевіряють, чи їхні кеш-адреси з'явилися на шині.

Якщо це так, то відповідні дані в кеші або анулюються, або оновлюються в залежності від протоколу що використовується. Шина забезпечує у будь-якому випадку строго послідовне виконання усіх записів.

Відомі також протоколи, в яких частина інформації розташовується в кеш-пам'яті. Позитив такого збереження інформації в єдиному довіднику є простота протоколу, обумовлена зберіганням необхідної інформації в одному місці.

Недолік таких довідників – розмір пропорційний загальному обсягу пам'яті, а не розміру кеш-пам'яті. Це не проблема для систем з сотень процесорів, а щодо великих систем слід «ефективно масштабувати структуру довідника» - це не проектне рішення, а побажання.

На рис. 1.6 подано приклади протоколів спостереження в системах розподіленої пам'яті для різних виробників в США. Подано типи протоколу, стратегії запису в пам'ять, а також відмінності виробника від інших.

В реальному проектуванні завжди існує радикальне проектне рішення, що вимагає більших витрат інтелекту – починати від змісту і

структури задач об'єкта управління. Крім того таке рішення не може бути знайдене «як краще» серед аналогів. Тому аналітики і філософи для такої ситуації знайшли термін «serendipity» - знаходження такого, що не було об'єктом пошуку. На рис. 1.7 подано характеристики міжпоєднань ряду мультипроцесорних систем з розподіленою пам'ятю

Назва	Тип протоколу	Стратегія запису в пам'ять	Унікальні властивості. Застосування
Одиничний запис	Запис з анулюванням	Зворотне копіювання при першому запису	Перший описаний в літературі протокол спостереження
Synapse N+1	Запис з анулюванням	Зворотне копіювання	Точний стан, де "власником є пам'ять" машин Synapse - перші машини з когерентною кеш-пам'ятю
Berkely	Запис з анулюванням	Зворотне копіювання	Стан "розподіляемий" - машина SPUR університета Berkely
Illinois	Запис з анулюванням	Зворотне копіювання	Стан "приватний"; може передавати дані з любого кеша - серії Power і Challenge компанії Silicon Graphics
"Firefly"	Запис з трансляцією	Зворотне копіювання для "приватних" блоків	Оновлення пам'яті під час трансляції SPARCcenter 2010

Рисунок 1.6 - Приклади протоколів спостереження в системах розподіленої пам'яті

Підведем підсумки огляду літератури стосовно мультипроцесорних систем з розподіленою пам'ятю. Якщо вести пошук на повній множині варіантів забезпечення обчислювальних потреб альтернативами: однопроцесорна ЦОМ, двопроцесорна, ...мультипроцесорна. На практиці це вирішується інакше.

Перший висновок – ця область аналізу і синтезу даного класу обчислювальних систем в комп'ютерній літературі розглядається в повному

відстороненні від використання комп'ютерів за конкретним призначенням: управління агрегатами, підприємствами, системами обслуговування «прозорих офісів» і управління важкими вантажними машинами.

В останньому застосуванні мультипроцесорність природна: кожне колесо має свій мікроконтролер, вся мультикомп'ютерна мережа діє скоординовано в заданих режимах – зарулювання на стостоянку, аварійне гальмування, та ін. В розглянутій літературі не знайдено матеріалів відносно відмовобезпечності і відмовостійкості мультипроцесорних систем.

Фірма	Назва	Кількість вузлів	Базова топологія	Розрядність зв'язку (біт) Частота синхронізації (МГц)
Thinking Machines	CM-2	1024-4096	12 - dimension cub	17110241987
nCube	nCube/ten	1-1024	10 - dimension cub	1101.26401987
Intel	iPSC/2	16-128	7 - dimension cub	11623451988
Maspar	MP-1216	32-512	2-вимірна мережа + ступінчаста Omega	125313001989
Intel	Delta	540	2-вимірна мережа	1640406401991
Thinking Machines	CM-5	32-2048	Багатоступінчасте товсте дерево	44020102401991
Meiko	CS-2	2-1024	Багатоступінчасте товсте дерево	87050500001992
Intel	Paragon	4-1024	2-вимірна мережа	1610020064001992
Cray Research	T3D	16-1024	3- вимірний тор	16150300192001993

Рисунок 1.7 - Характеристики міжпоєднань певних мультипроцесорних систем з розподіленою пам'яттю

Тобто розробка систем оцінювання ефективності мультипроцесорних систем актуальна, але потребує системного підходу, зокрема - розширення границь системи. Звернемось до аналогів: сьогодні газотурбінні двигуни (ГТД) застосовуються не тільки для літаків, ще й для вертольотів, кораблів,

мобільних електростанцій, очищення доріг. В них ідентичні і подібні модулі, але вимоги диктуються об'єктом «обслуговування», його цільовим призначенням і показниками ефективності. Теж саме для мультипроцесорних систем. Наприклад сучасні системи управління ГТД - цифрові, вони складаються з десятків підсистем, що мають свої спеціалізовані мікропроцесори. Ці мікропроцесорні системи об'єднуються в цілісну спеціалізовану мережу. Узгоджене їх функціонування забезпечує надійну і ефективну роботу ГТД.

1.4 Існуючі моделі і засоби оцінки ефективності комп'ютерних систем

В попередньому підрозділі ми розглянули типову особливість спеціалізованих мультипроцесорних систем управління складними об'єктами. Природний і надійний шлях оцінити їх ефективність – випробувати мультипроцесорну систему сумісно з об'єктом управління в заданих режимах.

Для оцінювання ефективності роботи комп'ютерних систем великого поширення набули тестові програми, узяті з різних наочних областей або модельні, або реальні, промислового застосування. Такі тести дозволяють оцінити продуктивність комп'ютера дійсно на реальних завданнях і отримати якнайповніше уявлення про ефективність роботи комп'ютера з конкретним застосуванням.

Для того, щоб оцінити ефективність роботи комп'ютерної системи на реальних завданнях, був розроблений фіксований набір тестів.. Проте LINPACK має істотний недолік: програма розпаралелюється, тому неможливо оцінити ефективність роботи комунікаційного компоненту суперкомп'ютера.

Проте при всій різноманітності тестові програми не можуть дати повного уявлення про роботу комп'ютера в різних режимах. Тому завдання визначення реальної продуктивності багатопроцесорних обчислювальних

систем залишається поки невирішеним. Головним інструментом прогнозування поведінки ЦУС в такій ситуації є моделювання. В дипломній роботі розробляється імітаційна модель для дослідження роботи цифрових управляючих систем. Модель повинна імітувати функціонування системи при різних навантаженнях та можливих відмовах певних елементів, обчислювати стандартні характеристики [2]. Цілі розробленої системи: можливість вибору і настроювання параметрів імітаційної моделі; розрахунок оптимальних рішень для протидії загрозам для об'єкту і комп'ютерної системи на базі моделювання; оцінка надійності функціонування цифрової управляючої системи; дослідження і навчання.

1.5 Висновки до розділу

Переглянуто аналоги і прототипи. Відібрано за показниками новизни і продуктивності – теоретичних основ та практичних особливостей, базові аналоги. Аналіз публікацій стосовно стану теорії і практики за темою роботи показав її актуальність. Сьогодні ефективність об'єктів комп'ютерного управління суттєво залежить від ефективності комп'ютерної системи. Урахування реалій у впровадженні іноваційних технологій в комп'ютерні системи сьогодні переважно емпірична задача, складна для формалізації та розробки моделей і методів управління. Задача даної роботи не має чітких границь і суттєво пов'язана з іншими аспектами обслуговування об'єктів та задоволення їх вимог. Виконано аналіз переваг і недоліків альтернативних типів розподіленої пам'яті мультіпроцесорних систем. Виявлено неповноту існуючих методів оцінки ефективності мультіпроцесорних систем, а саме

- не знайдено теоретично обґрунтованих і практично ефективних засобів забезпечення відмовостійкості і живучості як об'єкта обслуговування, так власне комп'ютерної системи;

- не знайдено методів оптимального узгодження структур комп'ютерної системи і об'єкта управління.

В прикладних роботах знайдено оцінки ефективності комп'ютерної системи як прирощення ефективності об'єкта обслуговування. Для рішення суперечливих проблем тестування і оцінювання ефективності вибрано метод оптимального агрегування, що вирішує значну частину розглянутих проблем через їх зняття. Зокрема, знято проблеми пошукових методів.

2 УДОСКОНАЛЕННЯ КОМПЛЕКСУ МОДЕЛЕЙ І ПРОГРАМ ДЛЯ ОЦІНКИ ЕФЕКТИВНОСТІ МУЛЬТИПРОЦЕСОРНОЇ СИСТЕМИ

В данному розділі виконується розв'язання визначених в розділі 1 проблем, пропонуються напрями їх вирішення, при цьому реалізується покращення відомих загальноприйнятих підходів щодо існуючих способів оцінки ефективності мультипроцесорних систем.

2.1. Аналіз структур і задач управління виробничих систем

Для досягнення поставленої в роботі мети, як один з елементів вирішено використовувати декомпозицію. Під декомпозицією звичайно розуміється розв'язання такої проблеми: задано глобальну задачу, після чого слід знайти такі задачі для елементів системи, паралельне розв'язання яких, веде до розв'язання глобальної задачі [4,9]. При використанні вищезгаданого методу виникають певна проблема. Проблема декомпозиції зводиться до трьох підпроблем, а саме до:

- проблеми синтезу елемента координації;
- проблеми модифікації;
- проблеми пошуку процедури координації.

Вважається, що метод координації окремих задач базується на відповідному методі декомпозиції [8]. Для великих організаційно-технічних систем Месарович розглядає багаторівневі ієрархічні декомпозиції. Стосовно технічних систем він виділяє такі три поняття (інтерпретації) рівнів декомпозиції:

- а) страта – рівень абстрагування опису системи;
- б) шар – рівень складності рішень;
- в) ешелон – організаційний рівень.

Тобто виконується три ієрархічних декомпозиції, причому так, щоб мінімізувати зв'язки між рівнями. Ціль вищого рівня – узгодження декомпозицій, так щоб їх рівні збігалися. В такому випадку на кожному рівні

будуть:

- свої об'єкти і поняття;
- свої задачі і методи прийняття рішень;
- свої організаційні структури, що приймають і реалізують ці рішення відносно об'єктів свого рівня.

Типовим прикладом чітко стратифікованої системи є ЦОМ, що має такі страти:

- рівень фізичних процесів в елементах мікросхем;
- рівень обробки імпульсів / рівень мікросхем;
- рівень обробки машинних слів / рівень функціональних модулів;
- рівень асемблера / рівень функціональних модулів;
- рівень програмних модулів/ рівень мікро контролера, ПЕОМ;
- рівень програмних систем / рівень ПЕОМ і локальних мереж.

Ще один приклад – мережева база даних, що має рівні файл–сервера, клієнт–сервера та ЦОМ клієнтів. Але це не ієрархічна система, файл–сервер не командує ЦОМ клієнтів, а обслуговує їх. Загальні принципи вибору страт:

1. Вибір страт, в термінах яких описується система, залежить від задач поставлених розробником чи дослідником;
2. Аспекти функціонування на різних стратах в загальному випадку не пов'язані між собою, тому принципи і закони певної страти не можуть бути виведені з принципів і законів інших страт;
3. Залежність між умовами функціонування на різних стратах є асиметричною;
4. Кожна страта має свій набір термінів, концепцій і принципів;
5. Розуміння системи зростає при переході по стратах: при підйомі до верхніх проявляються смисл і значення системи, при спуску зростає деталізація системи.

Розбиття на шари прийняття рішень у великих системах обумовлено

двома важливими особливостями процесів прийняття рішень:

- 1) коли приходить час прийняття рішення, то неприйняття рішення – теж рішення, іноді вдале;
- 2) невизначеність наслідків альтернатив, незнання певних альтернатив, дефіцит ресурсів, обмежують можливості раціонального вибору рішення.

Саме це обумовлює розбиття процесу прийняття рішень в певні шари (слої). Наприклад, задачу управління складною системою можна розбити на самоорганізацію, навчання і адаптацію, регулювання і вибір. На верхніх рівнях рішення приймаються рідко, на нижніх дуже часто. Звичайно нижні рівні прийняття рішень є автоматизованими. Концепції багаторівневої декомпозиції мають свої області застосування.

Взаємозв'язок страт, шарів та ешелонів повинен бути таким, щоб забезпечити взаємно однозначну відповідність цих рівнів. Проведення багаторівневих декомпозицій – неформалізований і нечіткий процес. На рис. 2.1 подана Структура управління вертикально інтегрованою системою

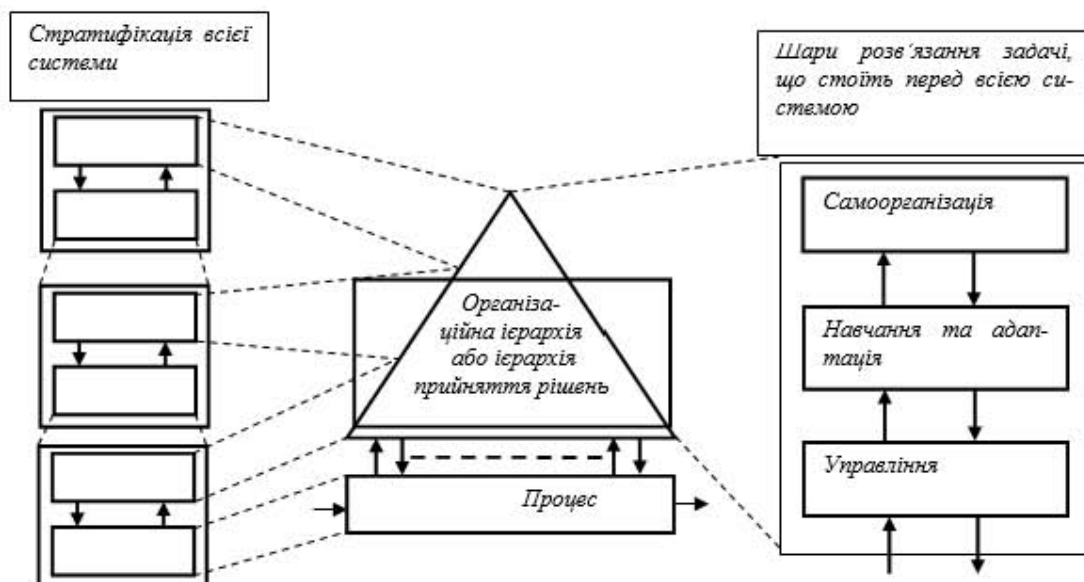


Рисунок 2.1 - Структура управління вертикально інтегрованою системою

В ієрархічній системі існують залежності двох типів: горизонтальні – між елементами одного рівня, і вертикальні – між елементами різних рівнів.

Елементи одного рівня пов'язані через спільні ресурси, спільний виробничий процес. Звичайно елементи мають власні критерії і цілі, і формують управління саме виходячи з своїх критеріїв.

Елементи різних рівнів пов'язані асиметрично – на верхній рівень передається певна інформація про стан і прийняті рішення (наміри) виробничих елементів, на нижній рівень йдуть з верхнього специфічні управління – вони задають однакові для всіх елементів нижнього рівня умови функціонування. Ці управління Месарович називає координаційними і пропонує два варіанти координаційного управління – „прогнозування взаємодії” та „оцінка взаємодії”. На рис. 2.2 подана багатопшара структура функціонування елементів управління. Це приклад науково обгрунтованого завантаження комп'ютерної системи інтелектуальними та інформаційними технологіями.

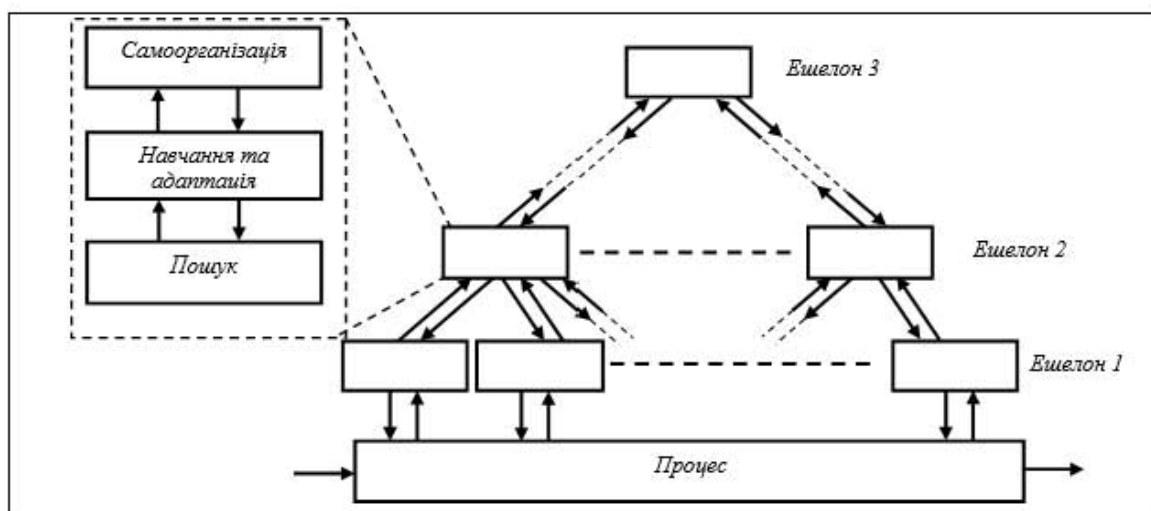


Рисунок 2.2 - Багатопшарове подання функціонування елементів управління

Уточнимо термін „багатопшаровий” – це скоріше „багато листковий”, а точніше – „многослойный”, або „multilayered”. На рис. 2.3 подано схему багаторівневої системи з управлінням по методу прогнозування взаємодій. На цій схемі $P1$ та $P2$ – узагальнені виробничі процеси, $C1$ та $C2$ – елементи управління нижнього рівня, $C0$ – елемент управління верхнього

рівня. Нижні елементи управляють своїми підпроцесами, майже незалежно від інших підпроцесів, але з урахуванням координаційного управління, що формує і передає елементам нижнього рівня елементам верхнього рівня.

Між підпроцесами є певні взаємозв'язки, але між елементами управління нижнього рівня безпосередні зв'язки відсутні. Інформацію про зв'язки між підпроцесами збирає верхній елемент, порівнює з прогнозами і коректує прогнози взаємодії, що видаються елементам управління нижнього рівня. Така схема декомпозиції управління раціональна і задовільна у всьому, крім інтерпретацій саме для великих систем.

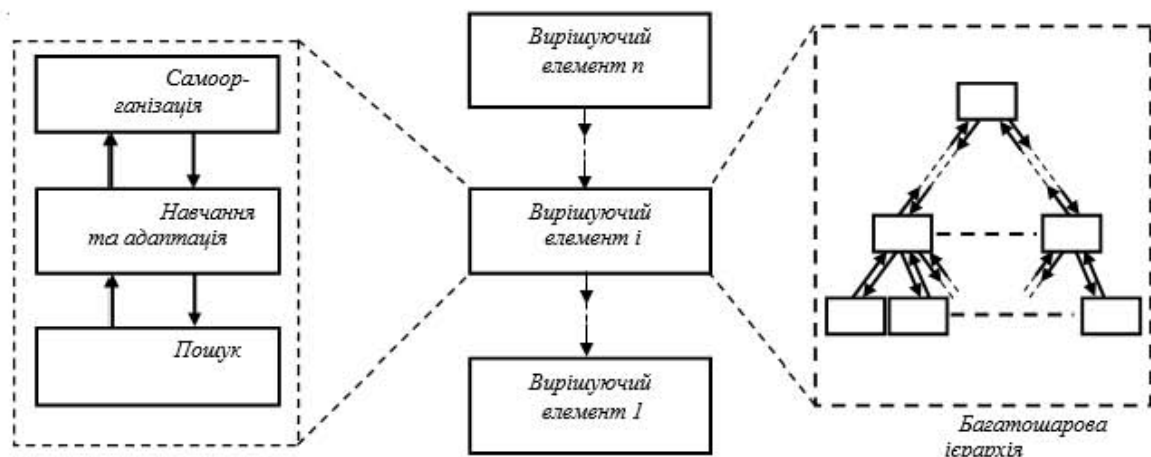


Рисунок 2.3 - Подання елементів управління, що утворюють багатошарову ієрархію, у вигляді багатошарових та багатоешелонних ієрархій

Бачимо, що для управління відсутнім на схемі об'єктом будується цілісна інтелектуальна структура управління: ієрархія управлінських структур, що взаємно управляються. На кожному рівні управлінської структури пропонується три рівні інтелектуального управління: «самоорганізація», «навчання та адаптація», «пошук». В пошуку за даними термінами видаються переважно описові роботи.

2.2 Моделі процесів утворення розподілених виробничих систем

Аналіз літератури з декомпозиційних методів дозволяє побачити пробіл відносно потреб сучасності – статичність декомпозицій. Характерна особливість реальних великих розподілених і мережових систем в тому, що вони є результатами певних еволюційних процесів розвитку і навчання. При аналізі і синтезі управління в таких системах слід враховувати ці процеси – ефективне управління повинно бути узгоджене з „природними” процесами функціонування і „навчання” розподілених систем. Реальні розподілені системи виникають в результаті двох типів процесів розвитку: К–системи, що виникають при об’єднанні незалежних систем в системи з певними зв’язками; Д–системи – результат декомпозиції „монолітної” системи певними способами в підсистеми. Ці процеси можуть бути як стихійними, так і результатом цілеспрямованого, свідомого проектування мережової системи. В даній роботі ставиться мета розробити комплекс математичних моделей і методів моделювання еволюції розподілених систем двох названих типів. Знайдено тільки один аналог запропонованого підходу до декомпозиції як стану процесу розвитку технічної системи – монографія «Управление конфигурацией технических систем» [3], де розглядаються організаційні аспекти процесів вилучення старих, неефективних підсистем і введення нових та ін.

Реальні розподілені системи звичайно навчаються, еволюціонують. Природно, що моделі повинні відображувати це „навчання”, розвиток. Реальні розподілені системи виникають в результаті двох типів процесів розвитку:

- об’єднання незалежних систем певними зв’язками, як, наприклад, інформаційно–обчислювальні мережі;
- розбиття „монолітної” системи певними способами в підсистеми для підвищення ефективності.

Номінально розподілені системи, що є результатом об’єднання

елементів (обчислювальні і телефонні мережі) можуть здаватися результатом стихійних процесів, а розподілені системи, що є результатом розбиття складної системи, складної задачі, складної моделі можуть здаватися результатом цілеспрямованого проектування розподіленої системи. Реально, навіть в рамках однієї розподіленої системи одночасно йдуть процеси композиції і декомпозиції, тому двох класів повинні бути частинами однієї узагальненої моделі.

Розглянемо спочатку математичні моделі мережевих структур, що мають зв'язки тільки через певний спільний ресурс. Ці моделі мають дуже багато інтерпретацій – від нейронних мереж і екологічних систем до трансп'ютерів і локальних мереж підприємства. Для досліджень в даній роботі виділимо такі класи задач моделювання систем, пов'язаних через спільний обчислювальний ресурс – система задач класу „ $N \times M$ ” (N обчислювальних елементів, M задач), що складається з трьох класів задач:

- задача розподілу „1 елемент для виконання M задач”;
- задача розподілу „ N елементів для виконання однієї задачі”;
- задача розподілу „ N елементів для виконання M задач”.

Відпрацювання робочих моделей для систем класу „ $N \times M$ ” дозволяє досить просто будувати робочі моделі і вести дослідження для різних мережевих систем

2.3 Постановка задачі моделювання розподілених систем класу «композиційна структура»

В цьому підрозділі подано базову математичну модель статичної і динаміки виробничих систем – як інформаційного так і матеріального виробництва. Тобто ми вибираємо єдину математичну модель і для комп'ютерної системи і для об'єкта управління - виробництва. На базі цієї математичної моделі в розділі 3 подано програму для тестування.

Моделі К-систем. Аналіз інформації відносно процесів розвитку певних класів виробів, виробництв, систем управління підприємствами

дозволяє виділити окремий клас розподілених систем різної природи – обчислювальних, техніко–економічних, інформаційних, які можна описати єдиною математичною моделлю „ N елементів M задач” [7, 8]. Поведінка елемента описується узагальненою імовірнісною моделлю зростання з обмеженням та імовірнісною моделлю локального управління розподілом ресурсу елемента між „виробництвом” окремих „продуктів”.

Розглядаємо систему з N елементів, що обробляють потік даних, який складається з задач і завдань M видів. Елементи мають різні характеристики виконання задач різних видів. Природно розподілити навантаження між елементами так, щоб сумарна продуктивність системи була максимальною, природно також розподілити інтенсивності виконання задач різних видів в кожному елементі так, щоб сумарна продуктивність елемента була максимальною. В умовах, коли реальні системи швидко еволюціонують, потрібен комплекс з модульної робочої моделі і раціональні технології розширення і модифікації моделі.

Вважаємо, що елементи системи мають власні (локальні) цілі, критерії, обмінюються певною інформацією, мають певні власні ресурси. Елементи розподіляють власні ресурси між напрямками своєї діяльності так, щоб оптимізувати власний критерій. Розглядаємо ситуацію, коли узагальнений ресурс (зокрема – задачі) ділиться в залежності від очікуваних ефективностей, детерміновано, або ймовірно. Імовірнісний розподіл може мати такі інтерпретації:

- метод "боротьби з невизначеністю" зовнішнього оточення – невизначеність, певна випадковість рішень. Керована невизначеність рішень – основа алгоритмів випадкового пошуку, так же як керована нечіткість – основа алгоритмів оптимізації на базі нечіткої логіки;

- відображення „людського фактору” в мережевих системах. Природно оцінити вплив такої випадковості на розвиток технічних систем. Робимо узагальнену модель з урахуванням імовірнісних факторів.

В [13] запропонована узагальнена модель функціонування мережевої

системи. За основу беремо відому детерміновану модель росту з обмеженням, але параметри її вважаємо імовірнісними. Логіка моделі функціонування така: обсяг узагальненого функціонування (опрацювання інформації, надання послуг (СМО)) зростає експоненційно при необмеженому обсягу потреб. Рано чи пізно сумарна співпраця усіх елементів заповнить потреби. Якщо сумарна продуктивність перевищує обсяг потреб, то „виробничі потужності” будуть скорочуватись, і для цього теж можуть бути потрібні затрати. Ці міркування можемо відобразити нелінійним різницеvim робочим рівнянням:

$$\Delta X_{i,j} = [ef_i[(X_t)_{i,j}] \cdot \left(\frac{Ryn_j - Sus_{j,t}}{Ryn_j} \right) \cdot Inv_{i,j}] \cdot krok;$$

$$(X_{t+1}) = (X_t)_{i,j} + \Delta X_{i,j}; i := 1 \dots N; j := 1 \dots M, \quad (2.1)$$

де ΔX_{ij} – поточне прирощення обробки i -тим елементом j -ї задачі,

ef_i – коефіцієнт ефективності,

X_t – матриця стану системи,

$(X_t)_{i,j}$ – елемент матриці стану обробки i -тим елементом j -ї задачі,

Ryn_j – потенційний попит,

$Sus_{j,t}$ – поточне сумарне виробництво,

$Inv_{i,j}$ – темп зовнішніх ресурсів для виконання j -ї задачі,

$krok$ – крок моделювання.

Будемо вважати параметр моделі (1.1) ef_i випадковою величиною з певним розподілом ймовірностей. Припускаємо, що цей розподіл може бути ідентифікованим в процесі функціонування з достатньою для управління точністю.

Моделі локального управління. Локальне управління в даному випадку:

– розподіл елементами узагальнених ресурсів між обробкою різних задач. Вибираємо таку детерміновану модель розподілу: для кожного

продукта елемент оцінює перспективність розширення виробництва. Мірою перспективності є певний критерій – „успіх” кожного продукту. Тут можуть бути такі альтернативи:

- згідно усередненому за певний період «успіху» кожного продукту;
- згідно сумі усереднених успіху та темпу успіху (темп – синонім похідної);
- згідно сумі усереднених успіху, темпу успіху і темпу від темпу успіху.

Параметризована детерміновано–імовірнісна модель розподілу ресурсів дозволяє записати ці три альтернативи виразом подібним рівнянню ПД–регулятора. Логіка розподілу ресурсу. „Інтуїтивна логіка” диктує правило розподілу поточного ресурсу: пропорційно перспективностям продуктів

$$(Roz_t)_{i,j} = efp_{i,j} \div \left(\sum_j efp_{i,j} \right). \quad (2.2)$$

Логіка „здорового глузду”: «враховувати ризики» диктує інше правило: поділити ресурс на дві частини: одну – $X_t \cdot \gamma$, розподілити пропорційно перспективностям, а другу – $X_t \cdot (1-\gamma)$, „розіграти в лотереї”, де шанси кожного продукту пропорційні його перспективності. Так отримуємо досить просту модель для широкого спектру алгоритмів локального управління розподілом ресурсів (як пристроїв, так і часу обслуговування) – від чисто імовірнісних ($\gamma=0$) до детермінованих ($\gamma=1$).

2.4 Задачі моделювання розподілених систем класу «декомпозиція»

Існуючі підходи до моделювання, аналізу і синтезу розподілених систем фактично створені для стабільних систем. В [9] запропоновано концепцію тривимірної декомпозиції, розраховану на роботу з відкритими еволюціонуючими системами. Помітною частиною процесу функціонування сучасних складних систем є реконфігурація та перерозподіл ресурсів для

адаптації до змін цілей, збурень різного виду, в тому числі, до відмов.

Для інформаційних систем, це може бути система з неідентичними паралельними каналами і можливістю реконфігурації на цих паралельних каналах. Такими паралельними каналами можуть бути завчасно синтезовані регулятори (фактично програми у мікроконтролерах системи управління), для мереж це можуть бути альтернативні маршрути передачі даних. Розглянемо підхід до побудови математичних моделей декомпозиційних структур технічних систем оптимальних в номінальних і неномінальних – при дії збурень.

Математичні моделі критеріальної оцінки розподілених систем. На сьогодні вимоги до технічних систем сформувалися, увійшли в стандарти проектування. Типові критерії – це надійність, живучість, ефективності і модифікуємість. Однак ці критерії є суперечливими і неконструктивними – вони ставлять вимоги до технічних систем та не визначають шляхи задоволення цих вимог.

Ситуація ускладнюється, коли розподілена система є інноваційною і не має статистики і відпрацьованих та перевірених рішень. Спеціалістам відомо, що сьогодні живучість інтелектуальних регуляторів в середньому нижчі ніж у звичайних. Для цього є певні підстави, в цьому певна логіка і закони розвитку, коли нове спочатку має дуже багато «дитячих хвороб». Зокрема, ефективне рішення проблеми живучості отримане на базі методології оптимального агрегування [12]

Згідно з цими тенденціями розвитку технічних систем домінуючою частиною процесу проектування повинен бути вибір деякої базової структури розбиття: системи на підсистеми; процесу проектування на етапи проектування; системи проектування в підсистеми, і розподіл узагальнених ресурсів між відповідними елементами: об'єктом проектування, процесом проектування та системою проектування.

Виходячи з визначень процесів функціонування та проектування систем, можна сформулювати сценарій для процедур оцінки ефективності і

живучості: „включимо (з урахуванням усіх інтерфейсних проблем) в склад наявної системи (моделі системи) нову підсистему і оцінимо, приріст ефективності”, а потім „виключимо цю підсистему (як мінімум, припустимо, що вона відключилася без наслідків для інших підсистем) і оцінимо зміну (втрату) ефективності”. Тобто модифікуємо базовий прототип і одразу ж оцінюємо одночасно позитивні і негативні наслідки включення нової підсистеми [11]. В таблиці 2.1. подано визначення пари взаємодоповнюючих задач проектування розподілених технологічних систем

Таблиця 2.1 - Критерії і задачі розробки розподіленої системи

Ефективність	Живучість
$Em(\bar{C}) = \max \{E(K(C));$ $Ki \in \{K\}\} C = Ci$	$Gm(\bar{C}) = \max \{G(K(C));$ $Ki \in \{K\}\} C = Ci$
$Q(\bar{C}) = Em(C) / C$	$F(\bar{C}) = Gm(C) / C$
$\Delta Q(C) = \Delta Em(C) / \Delta C$	$\Delta F(C) = \Delta G(C) / \Delta C$

Означення в таблиці 2.1 прийнято такі:

C, Ci – вартість, задана вартість для множини версій системи;

K, Ki – конфігурація, i -та конфігурація (версія) системи з вартістю Ci ;

$E(K(C))$ – ефективність деякої конфігурації вартістю C ;

$G(K(C))$ – втрати при відмові елементів сумарною вартістю C ;

$Q(C)$ – функція максимальної (на множині конфігурацій вартості Ci) ефективності (= структурна функція);

$F(C)$ – функція максимальних (на множині конфігурацій вартості Ci) втраг.

Ми одночасно знаходимо максимальний ефект для вводу підсистеми вартістю ΔC і максимальні втрати при відмовах вартістю ΔF .

По суті, приходимо до аналогу принципу мінімаксу відомого з теорії ігор – проектуємо систему, оцінюючи на кожному кроці проектування прирощення ефективності і втраг, обумовлених прогнозовними відмовами

системи [7, 8]. Шукаємо такий варіант модифікації, що дає мінімум максимальних витрат при можливих відмовах вартістю ΔC у модифікованій системі. Така методологія проектування систем залишиться неконструктивною, якщо не буде знайдена для об'єкта проектування деяка інваріантна базова структура. Такою структурою можуть бути методологія оптимального агрегування та 3D –декомпозиція, подана в наступному розділі.

Відомі так звані мілімодульні регулятори, матричні тренуємі регулятори, регулятори на базі зворотних моделей, інтелектуальні регулятори термінального управління. Особливість цих регуляторів – децентралізовані структури з взаємодіючих елементарних регуляторів, упорядкованих в деяку матрицю, куб, гіперкуб на основі певних декомпозицій.

2.5 Декомпозиційна структура для задач класу «декомпозиція»

Відомі філософсько-наукові концепції відношення «частина–ціле», методи розбиття систем на частини здаються самоочевидними. Однак потреби і досягнення практики стимулювали розвиток теорії. Для систем, де результати проектних помилок виявляються швидко, ефектно і коштують дуже дорого (космос, авіація, електростанції, банки), використовується розпаралелювання систем та підсистем у неідентичні канали, що виконують одну й ту ж функцію різними способами. Для програмних систем це має назву «багатоверсійне програмування», для технічних систем – «неідентичне резервування». В практиці проектування давно вже став стандартом принцип „плавної деградації”, коли ефективність системи зменшується пропорційно вартості підсистем, що вийшли з ладу. Усе це приклади практичного розв'язання проблеми декомпозиції для відкритих систем, що розвиваються. Вибираємо новий підхід до декомпозиції великих, цілеспрямованих систем – трирівневу декомпозицію [12].

Функціональна декомпозиція. Математичну модель (ММ) можна розбити

на субмоделі, що мають меншу кількість внутрішніх змінних при відносно невеликій кількості змінних, що подають зв'язки з іншими субмоделями. Така декомпозиція може бути ієрархічною – "... компоненти систем теж є системами". Кожна з субмоделей виконує (подає) якусь часткову функцію повної ММ. Назвемо таку декомпозицію функціональною та символічно подамо у виді:

$$\{fP_i\} = F(P), i = \underline{1 \dots K1},$$

де $\{fP_i\}$ – множина субсистем, $F(.)$ – оператор функціональної декомпозиції,

P – математична модель повної системи.

Редукційна декомпозиція. У прикладній математиці поширені задачі апроксимації деякої ММ рядом – ступеневим, тригонометричним. У відповідність до ММ повної системи поставимо множину моделей-апроксимацій, що належать до одного параметричного класу, тобто мають одну структуру і відрізняються тільки значеннями числових параметрів. Ці ММ виконують усі функції повної ММ, але з різними рівнями точності. Назвемо таку декомпозицію редукційною та символічно подамо у виді [12].

$$\{rP_j\} = R(P), j = \underline{1 \dots K2},$$

де $\{rP_j\}$ – множина субсистем, $R(.)$ – оператор редукційної декомпозиції.

Виникає питання – де шукати базові структури, що породжують редукційні декомпозиції? Чи не є це просто іншою назвою рядів Тейлора і Фур'є? Можна запропонувати більш загальну структуру для генерації сімейств редуктованих моделей, ніж апроксимація функції поліномом N -го порядку:

$$f_N(z) = c_0 + c_1 * z + c_2 * z^2 + \dots + c_n * z^N.$$

Це апроксимація Пале, що подає функцію у вигляді відношення поліномів:

$$f_{LM}(z) = (a_0 + a_1 * z + \dots + a_n * z^L) / (b_0 + b_1 * z + \dots + b_n * z^M).$$

Методи пониження порядку дифрівнянь, методи агрегування змінних

також дозволяють отримувати системи редукованих моделей.

Структурна декомпозиція. У реальному проектуванні типово ситуація, коли для деякого об'єкта чи процесу існує декілька, еквівалентних відносно критеріїв точності наближення, математичних моделей, що не належать до одного параметричного класу. Кожна з цих ММ виконує усі функції повної ММ. Назвемо таку декомпозицію структурною та подамо у виді:

$$\{sP_k\} = S(P), k = 1, \dots, K_3,$$

де $\{sP_k\}$ – множина субсистем, $S(\cdot)$ – оператор структурної декомпозиції.

В техніці структурна декомпозиція, – це реалізація системи з паралельно працюючими підсистемами, наприклад використання паралельно працюючих вимірювачів, чи приводів, побудованих за різними принципами, наприклад гідроприводів і електроприводів. Це робиться, зокрема для забезпечення відмовостійкості.

В загальному випадку декомпозиція системи є багаторівневою, ієрархічною. Введемо елементарну операцію декомпозиції, що складається з деякої послідовності виконання F-, S-, і R- декомпозицій, допустимо також, що кожна з трійки операцій може бути “одиночною”, тобто не розбиваючою об'єкт. Тоді можна описати послідовністю елементарних декомпозицій і звичну ієрархічну функціональну декомпозицію і менш звичну багаторівневу структурну. Різні послідовності операторів декомпозиції породжують різні елементарні декомпозиційні структури систем, наприклад [12] :

$$R\{S\{F\{P}\}\}, R\{F\{S\{P}\}\}, S\{R\{F\{P}\}\} \dots$$

Для деяких з цих декомпозицій можна знайти інтерпретації, наприклад, першій відповідає лінгвістична формула: система, де глобальна функція виконується як сукупність часткових функцій (F-декомпозиція), кожна часткова функція реалізується різними структурними способами (S-декомпозиція), кожен структурний спосіб реалізується з декількома рівнями точності (R-декомпозиція). Виявляється, що існуючі технічні, економічні, біологічні системи, їх можна віднести до однієї з цих декомпозиційних

структур. На базі цих структур можна будувати бази знань для експертних “надбудов” у САПР, мультипроцесорні САУ, можна побудувати сам процес проектування складної системи – “метод паралельних розробок” [6–8].

2.6. Синтез декомпозиційної структури для систем К-класу і Д-класу

В цій роботі ми розглядаємо розподілені комп’ютерні системи не як дещо фіксоване з визначеними характеристиками, а як процес розвитку. Ми розглянули два еволюційних процеси розвитку:

- об’єднання певних незалежних елементів у систему;
- розбиття „монолітної” системи на певні підсистеми.

Ці процеси можуть бути як цілеспрямованими так і спонтанними, стихійними.

Вводимо розподіл процесів еволюції декомпозиційної структури систем і відповідних декомпозиційних структур на два класи – К-системи і Д-системи (подані 2.3, 2.4). Як дескриптивну (описову) і прескриптивну (задану теоретично) модель для Д-систем вибираємо тривимірну декомпозицію для розбиття певної системи P на елементи – функціональні – $F(P)$, редуційні – $R(P)$ і структурні – $S(P)$. Ця модель, введена в [9] є дескриптивною – вона є формалізацією структур реальних технічних систем, нафтохімічних, енергетичних, металургійних підприємств. Ця модель корисна як прескриптивна для проектування нових виробничих систем та їх систем управління.

Ми вводимо аналогічну тривимірну композицію як модель для К-процесу збирання системи з елементів. Найближчий знайдений в літературі аналог – запропонований М.Пешелем підхід до побудови моделей складних систем з простих моделей по „анатомічному” і „фізіологічному” принципах [109]. Певний елемент p можна охарактеризувати „темою” – $T(p)$, „замінником” (альтернативним елементом) – $Z(p)$, „доповненням” – $D(p)$. Використана лінгвістична термінологія обумовлена подібністю створення коректно працюючої системи і складанням синтаксично і прагматично

коректного висловлювання. Розглянуті класи розподілених систем можна об'єднати в одну класифікаційну структуру. В цьому розділі поставлено за мету на основі аналізу літературних джерел не тільки проаналізувати існуючі декомпозиційні підходи і методи, але й узагальнити і доповнити їх з урахуванням властивостей сучасних розподілених технологічних систем – високих темпів кількісних і якісних змін.

Типовими показниками моделей вважаються адекватність, точність, збіжність. Для складних систем на перший план виходять інші показники. „Модель слід би оцінювати за її здатністю відтворювати або прогнозувати характеристики поведінки системи – стійкість, колювання, зростання, середній період колювань, загальні взаємозв'язки змінних, тенденції до підсилення або послаблення збурень”[16]. З багатьох робіт, де розглядаються проблеми ідентичності і точності моделей складних систем виділимо роботи Форестера [16] і відповідні висновки стосовно якості математичних моделей. Розподілені технічні системи є складними системами, для яких неможливо побудувати вичерпну «точну» модель [26].

Концепції управління розподіленими системами. Як було відзначено в розділі 1 в якості тестової задачі для мультипроцесорної системи використовують рішення задачі лінійного програмування. Ми вибираємо реалістичні нелінійні моделі виробництва. Розглянуті вище математичні моделі розподілених систем та оптимізаційні задачі для цих систем дають можливість підвести певні підсумки – описати два класи розподілених систем і два підходи до оптимізації розподілу ресурсу.

Централізоване управління. На рис. 2.4 подана схема централізованого управління розподілом ресурсів. Згідно цієї схеми центральний елемент на основі даних про характеристики елементів – узагальнені виробничі функції, розподіляє ресурс так, щоб отримати максимум сумарного виробництва. Елементи зручно розглядати як деякі технологічні перетворювачі отриманого ресурсу в кінцевий продукт. Для оптимального розподілу ресурсу

центральному елементу необхідно отримати розв'язання задачі оптимізації функції багатьох змінних при обмеженнях – розв'язати задачу нелінійного програмування високої розмірності з не випуклою цільовою функцією, або функціями обмежень. Додатковим ускладненням є те, що дані необхідні для визначення оптимального розподілу є нестационарними і невизначеними.

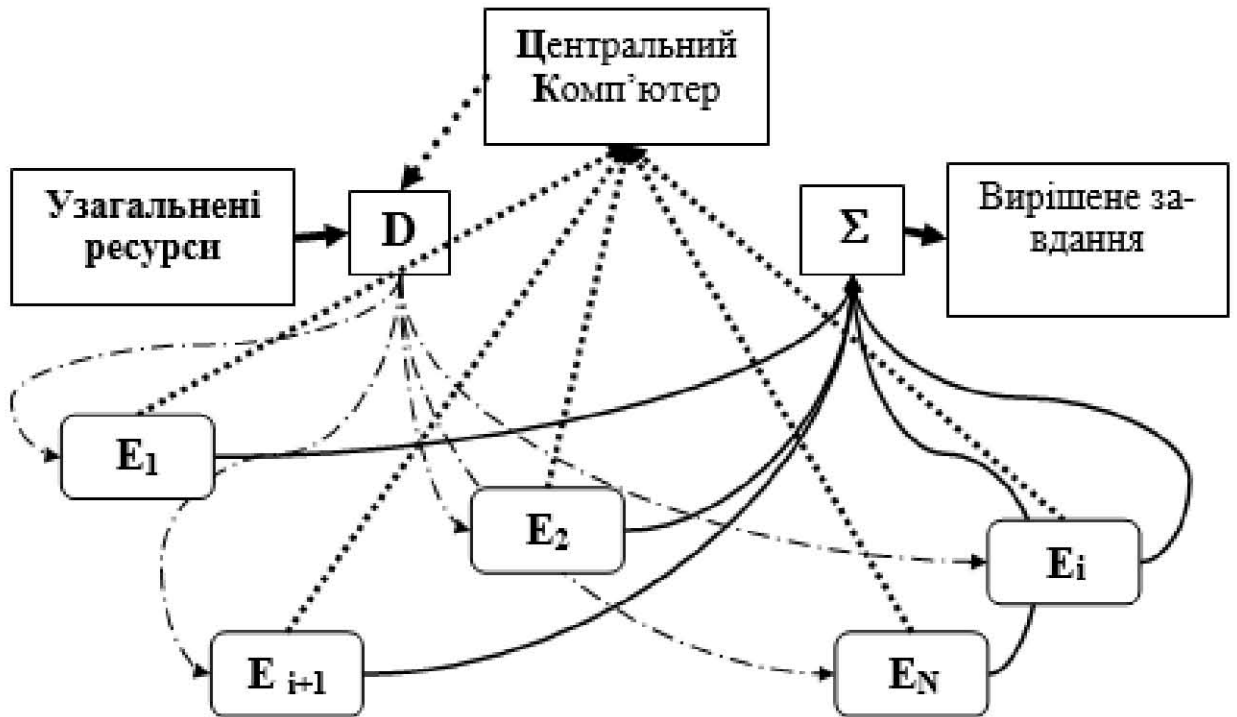


Рисунок 2.4 - Розподілена система з централізованим управлінням

На рис. 1.4 прийняті такі означення: E_1, E_2, \dots, E_N – елементи системи; Σ – вирішене завдання.

Децентралізоване управління на базі обміну ресурсами. На рис. 2.5 подана схема розподіленого управління розподілом ресурсів. Усі схеми, результати теорії розподілених систем з обміном ресурсами завжди можуть мати такі дві інтерпретації: а) схема системи управління реальною розподіленою системою з обміном ресурсами; б) схема обчислювального алгоритму оптимізації розподілу узагальнених ресурсів в розподіленій системі. Особливість обчислювального алгоритму в тому, що він фактично є

моделлю функціонування розподіленої системи – нелінійної, з багатьма зворотними зв'язками. При коректному виборі параметрів зворотних зв'язків система буде стійкою і оптимальною в усталеному стані буде близьким до оптимального.

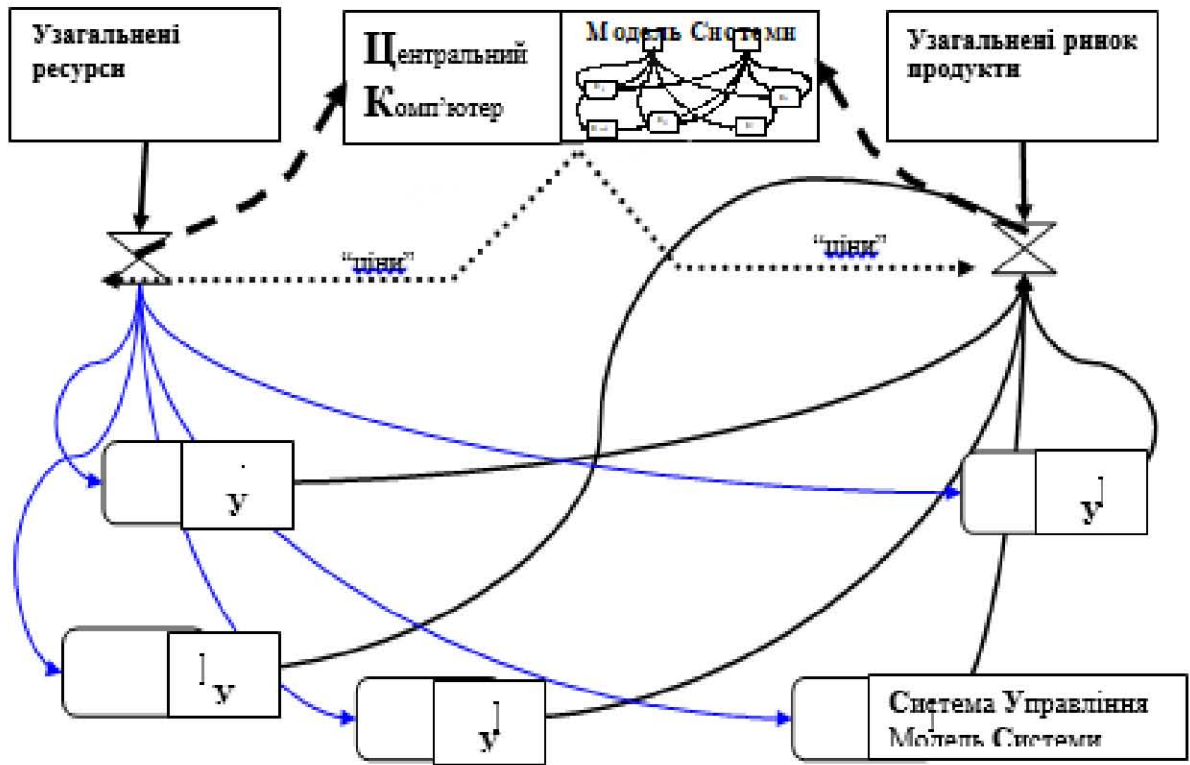


Рисунок 2.5 - Розподілена система з розподілом ресурсу на базі розпаралелювання оптимізаційної задачі

Цю ідею вперше чітко обґрунтував і розвинув Я.З.Ципкін [18]. Він назвав такий підхід „автоматною оптимізацією”. Концепція автоматної оптимізації фактично є узагальненням для усіх пошукових алгоритмів оптимізації із зворотними зв'язками, в тому числі і таких, що мають екзотичні назви: перцептрони, нейронні мережі, нейронечіткі мережі, мурашині алгоритми, штучні нейронні мережі та ін. Тепер опишемо функціонування системи з обміном ресурсами, поданої на рис. 2.6. Система складається з елементів, що виробляють певні узагальнені продукти – матеріальні, інформаційні. Для цього узагальненого виробництва елементи

отримують і витрачають узагальнені ресурси.

В системі існує центральний елемент (ЦО, це «головний комп'ютер» - комп'ютер входами якого є інформація про „дефіцити”, а виходами – вектори „затрат” ресурсів і продуктів [4]. В даному випадку „ціни” – параметри обчислювальних алгоритмів. На рис. 2.6 показано, що центральний елемент системи містить в собі модель розподіленої системи в цілому. Ця модель призначена для прогнозування майбутніх станів натуральної розподіленої системи. Елементи системи „подають заявки” на потрібні їм обсяги ресурсів і обслуговують певну узагальнену кількість заявок. Структуру виробництва і структуру „заявок” на ресурси елементи вибирають на базі оптимізації власних критеріїв оптимальності. Критерії оптимальності є функціями обсягів продукції, витрачених ресурсів. Елементи мають у своєму складі систему управління (СУ на рис. 1.6), що формує „заявки” на ресурси і „плани” виробництва та модель системи (МС на рис. 1.6), що дає можливість елементу прогнозувати „ціни” ресурсів і продуктів. Кожен елемент може мати інформацію про виходи (обсяги виробництва) і входи (витрати ресурсів) інших елементів. В такій децентралізованій системі глобальна оптимізаційна задача в явному виді не розв'язується. Центральний елемент тільки змінює „ціни” так, щоб зводити дефіцити до нуля, тобто є простим регулятором. Саме так визначається принцип координаційного управління, висунутий М.Месаровичем [9]. В даній роботі розглядається дворівнева ієрархічна система управління. Наявність інформаційних зв'язків, певна інформаційна симетрія в системі приводять до ще одної поширеної і продуктивної концепції – відкритого управління [5]. Суть відкритого управління в тому, що елементи знають, за якими правилами і алгоритмами змінюються „запити” і виходячи з цього планують свої дії. Саме це забезпечує оптимальність стану рівноваги.

2.7 Висновки до розділу

Згідно меті і задачам дослідження виконано аналіз відомих альтернативних теоретичних концепцій аналізу і синтезу сучасних комп'ютерних систем. Зокрема, продуктивні структури і методи запропоновані М. Месаровичем - багаторівневі ієрархічні системи з декількома класами рівнів (стратами, шарами, ешелонами). Недолік моделей Месаровича – структурування управлінь при відсутності аналізу структур і звязків. Конструктивні методи області управління великими розподіленими системами запропоновані в В.Опойцевим –«метаігровий синтез», «мінімально розумне управління» повністю доведені математичних моделей і алгоритмів. Головним недоліком цих методів були пошукові алгоритми без гарантованої збіжності, а також непрацездатність у випадку неопуклих функцій виробництва.

Ці недоліки існуючих методів підтвердили коректність вибору методу оптимального агрегування, який не має таких недоліків і ряд великих переваг стосовно оптимізації - відсутність пошукових процедур

З розглянутих альтернатив моделей і методів вибрано, для конкретної реалізації, такі сучасні і продуктивні для реалізації теоретичні положення:

- оптимальне агрегування – заміна багатовимірної задачі нелінійного програмування еквівалентною системою одновимірних задач;
- 3D – декомпозиція технічної системи в функціональні, структурні і редуційні підсистеми;
- об'єднання концепцій ефективності і живучості – це положення базується на двох перших, а зміст його - введення ресурсних функцій живучості.

Так створена робастна основа для побудови систем показників ефективності мультипроцесорної системи. Запропонована комп'ютерна система і підсистема оцінки ефективності, що дозволяє підвищити ефективність, надійність і живучість об'єкту і комп'ютерної системи.

3 ПРАКТИЧНА РЕАЛІЗАЦІЯ ЗАСОБІВ АНАЛІЗУ ЕФЕКТИВНОСТІ МУЛЬТИПРОЦЕСОРНИХ СИСТЕМ

В цьому розділі практично реалізуються відібрані в розділах 1 і 2 вимоги показників ефективності, теоретичні моделі використовуються як для побудови комплексу програмних модулів оцінки ефективності, так і розробки тестових програм-імітаторів об'єкта управління і зовнішнього середовища. Актуальність аналізу ефективності сьогодні швидко зростає з ускладнення функцій комп'ютерних систем управління виробництвами продуктів і послуг. З множини проблем відібрано таку, що вирішується в магістерській роботі – підсистему тестування

3.1 Структура і функції засобів для тестування і оцінки ефективності мультипроцесорних систем з розподіленою пам'яттю

Сучасні системи і продукти виробництва різноманітні до унікальності. Але для всіх виробництв спільні структури і логіка ресурсних та інформаційних зв'язків. На рис. 3.1 подана структура сучасної виробничої системи з комп'ютерним управлінням: виробнича система з «деталей» і «блоків» виробляє певний продукт для ринку. Функції виробництва розподілені між підсистемами. Сучасні підсистеми мають власні комп'ютерні системи, які ініціюють і контролюють технологічні процеси. Структури поєднання підсистем розглянута в розділі 2.

Техпроцеси в підсистемах можуть виконуватися в паралельних розгалужених структурах, послідовних, в тому числі конвеєрах, бути циклічними. У свій час новацією були конвеєри виробництва телевізорів – вони починались в підвалах на складах і закінчувались на складах готової продукції. Між цими пунктами конвеєр проходив 3-5 поверхів, мав окремі ділянки, де відновлювався виявлений брак окремих, і великі спец конвеєри, де цілу добу телевізори працювали включеними, потрапляли в зони вібрацій, високих температур, вологості.

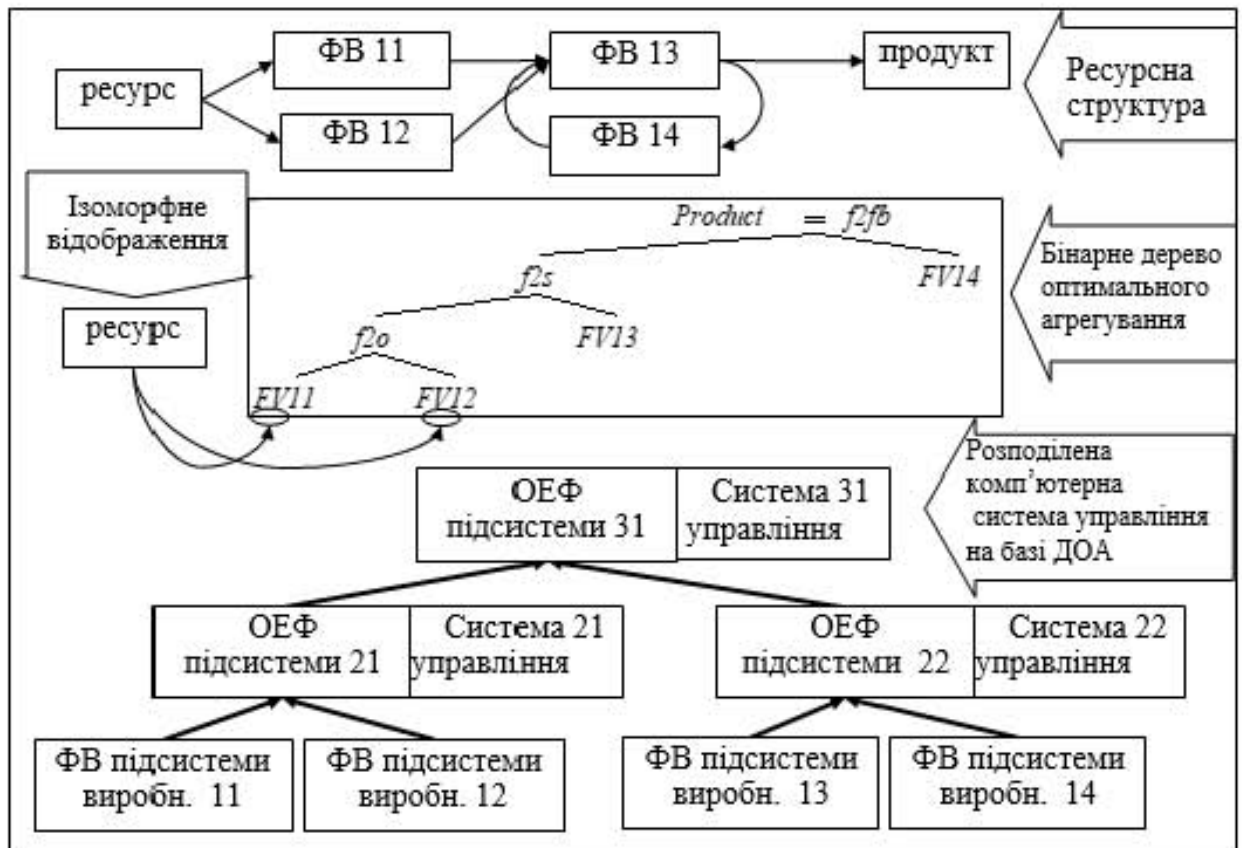


Рисунок 3.1- Оптимальне агрегування довільних ресурсних структур

Деякі з них вибухали, загорались на багатовитковому конвеєрі тестування - автоматика знімала їх і сортувала - в ремонт, в деталі, у відходи. Вже в сучасну епоху створені такого типу конвеєри для автомобілів. Тепер уявимо як повинна оцінюватись ефективність «мультипроцесорних систем з розподіленою пам'яттю», що обслуговують подібні виробництва системи. Майже очевидно, що в сучасних виробничих системах існують десятки і сотні технологічних підсистем і комп'ютерів, що обслуговують процес виробництва. Оскільки вагомі в середовищі виробничої системи показники комп'ютера показані на тесті для арифмометрів - рішення задач лінійного програмування, за умови, що вартість одиничної операції зменшилась в сотні разів.

Очевидно, що без комп'ютерних систем управління (КСУ) сучасні виробництва неможливі, КСУ без об'єкта і програмного забезпечення - просто апаратура.

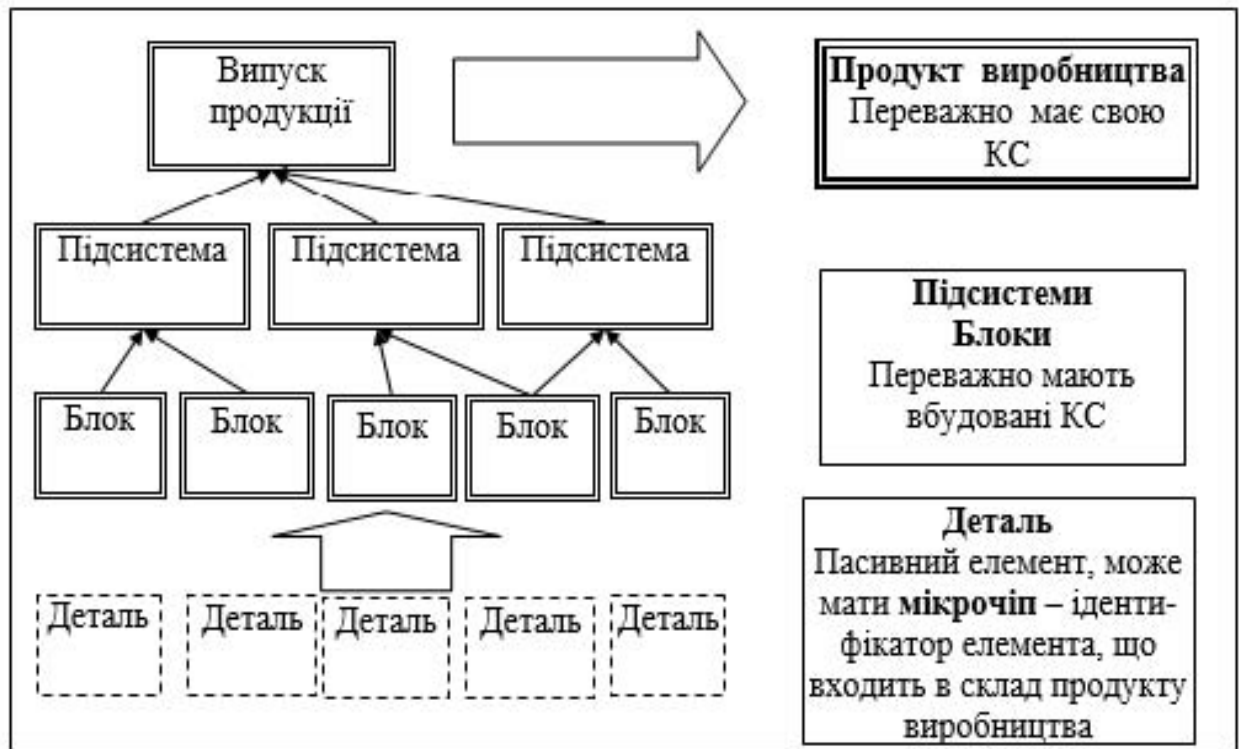


Рисунок 3.2 - Структура сучасної виробничої системи з комп'ютерним управлінням

На рис. 3.2 подана логіка системи оцінювання ефективності певного сучасного виробництва з сучасним управлінням і певної комп'ютерної системи задовільної для поставлених задач. На рис. 3.2 подано схему відображення ефективності КСУ в ефективність об'єкта управління (СМО, виробничої системи).

В якості оцінки ефективності беремо не число (скаляр), а функцію. Вводимо два види функцій класу «витрати», випуску»:

- функцію виробництва виробничої системи і підсистем $f_p(x)$;
- функцію продуктивності комп'ютерної системи, що обслуговує певну підсистему виробництва, систему верхнього рівня (випуск продукції) цілому підсистеми $Fk(xk)$.

Ці функції можуть бути методами прикладного системного аналізу (виявлення породжуючих елементів) і експериментами (на реальній та, або віртуальній реальності)

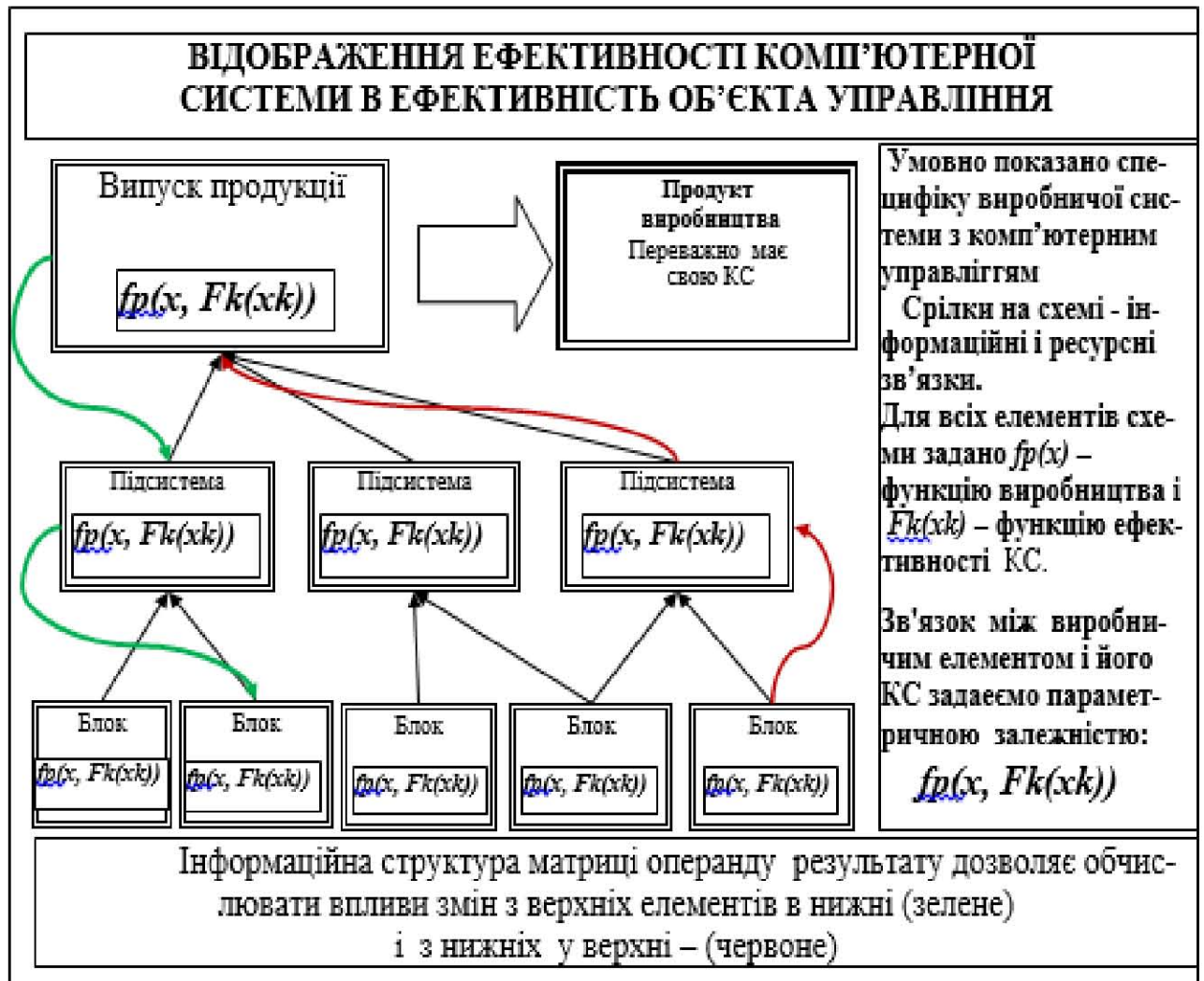


Рисунок 3.3 - Відображення ефективності комп'ютерної систем в ефективність обекта управління

Виробнику потрібна інформація про впливи ефективності певної КС на ефективність своєї підсистеми, але також інформація про вплив на кінцевий результат довільної підсистеми (елемента). Це задача теорії чутливості, яка може бути вирішена тільки на базі методології оптимального агрегування. Це розглядається в наступному підрозділі.

3.2 Оптимальне агрегування виробничої системи з урахуванням впливу комп'ютерних систем

Важливі переваги методів оптимального агрегування [7] порівняно з аналогами :

- відсутність математичних обмежень на узагальнені функції виробництва (здаємо назви: лінійне, квадратичне програмування);
- відсутність пошукових процедур і алгебраїчність;
- тільки лінійна залежність обсягу обчислень від розмірності задачі.

Остання специфічна, але реальна особливість: метод оптимального агрегування дозволяє ставити і вирішувати задачі, які неможливо навіть сформулювати в рамках класичних методів. Дослідимо приклади оптимального агрегування на базі яких побудовано комплекс засобів оцінювання ефективності МПСРП.

Методи оптимального агрегування базуються на ієрархічних структурах виробничих систем. Розв'язання оптимізаційної задачі для системи з 32 елементів розбивається в систему двовимірних задач оптимізації. Запишемо формулу оптимального агрегування для системи з трьох паралельно працюючих елементів в звичайній формі: $Ops3 := f2o(f1, f2o(f2, f3))$ і в структурній для чотирьох і п'яти елементів: Можна в це дерево додавати функції розвитку - скільки потрібно. Подаємо як інструкцію користувачу оптимальне агрегування. Коментуємо: - формула для $Ops3$ в звичайній формі - важко контролюється та занадто довга; формула для $Ops4$ легко контролюється; - формула $Ops5$ економна і зрозуміла: якщо треба перемножити 5 чисел, а 4 вже перемножені, так домножимо цей результат на $f5$ і отримаємо те що треба. Для контролю отримуємо $Ops5$ з використанням структурної форми.

На рис. 3.3 подано дві формули оптимального агрегування – для систем з 4-ох і з 5-ти елементів, результати агрегування і формат для вхідних операндів $f5$ і $f2$. Структура операндів: $f5$ і $f2$ - елементи, внутрішня структура відсутня, одиниці в другому стовпці це «увесь ресурс єдиному». Операнди результатів $Ops4, Ops5$: стовпці: 0 – квантоване значення ресурсу (вхід), 1 – оптимальний вихід для відповідного значення ресурсу; стовпці 2-5 і 2-6 - оптимальний розподіл даного ресурсу нормований (сума значень = 1).

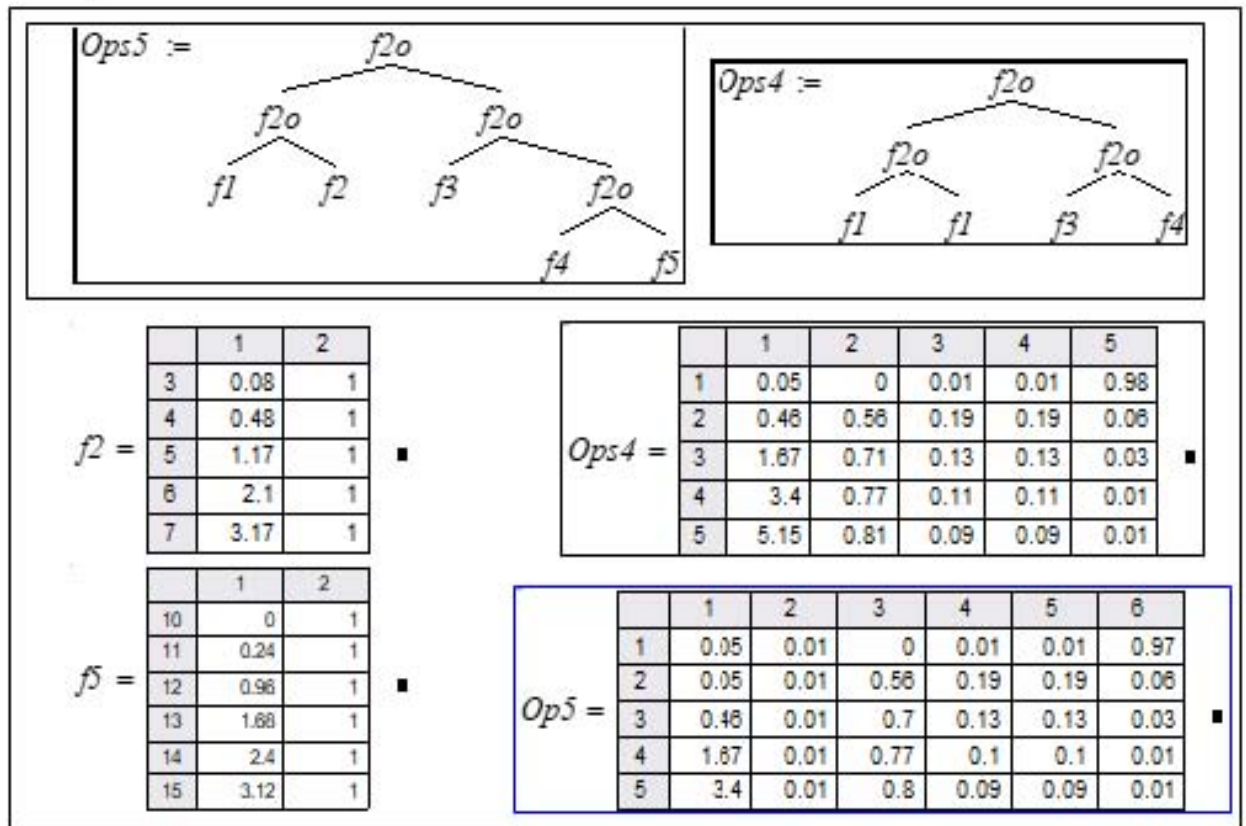


Рисунок 3.4 - Оптимальне агрегування

Дивимось на структурні формули на рис. 3.4. Бачимо, що оператор оптимального агрегування $f2o$ - бінарний, він бере пару ФВ і повертає ФВ. Матриці на рис. 3.4 дуже зручні для комп'ютерів, але малоінформативні для людини. На рис.3.5 подані параметризації з використанням оптимального агрегування.

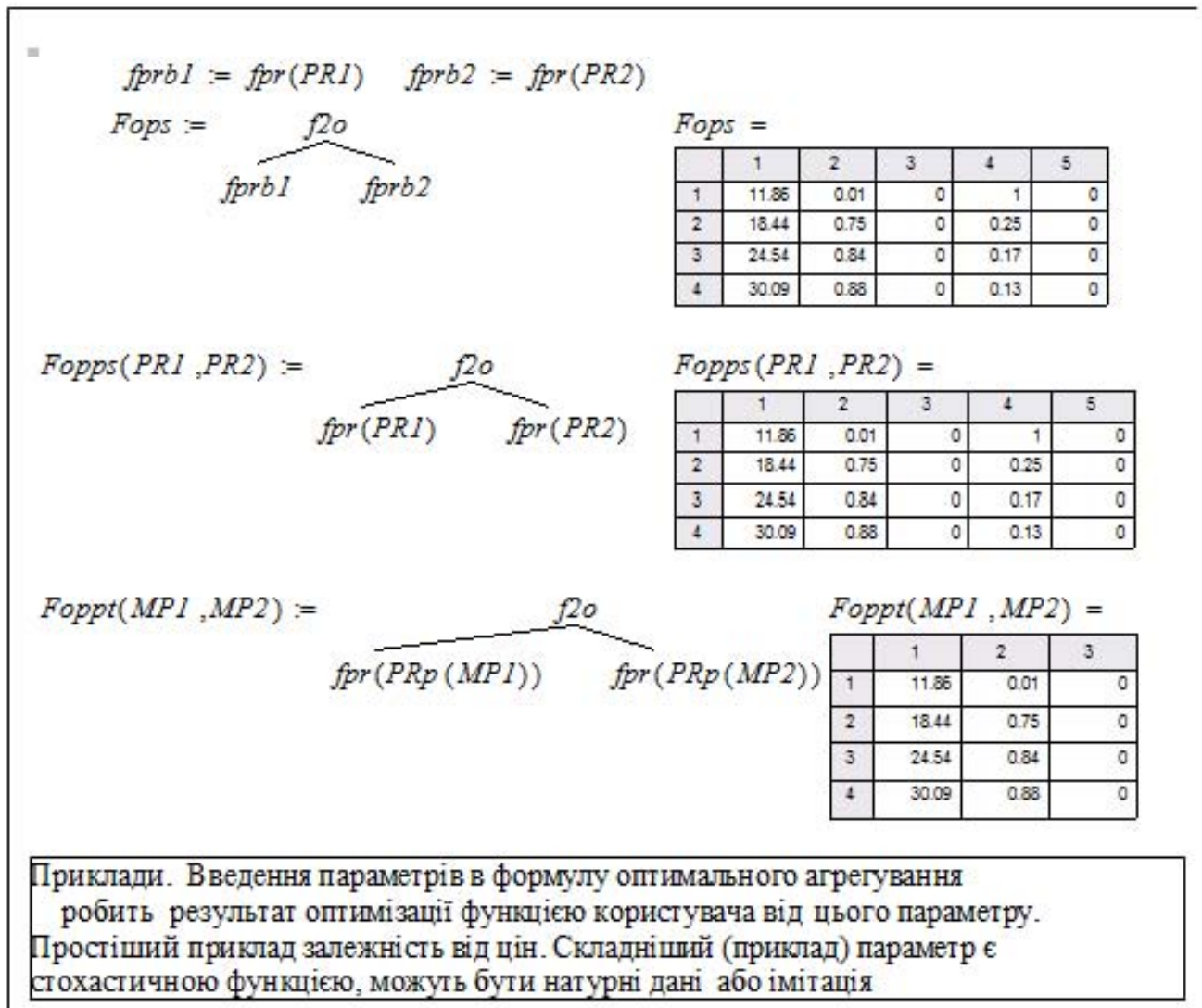


Рисунок 3.5 - Параметризація оператора оптимального агрегування.

На рис. 3.6 подано текст версії *vr9* підпрограми, параметри якої: Mt – матриця виробники, продукти, lox – рівень ризикового управління [13], rv - заданий довільний розподіл ймовірностей.

Бачимо, тепер залежність розподілу ресурсу від його обсягу (в нормованих одиницях – від 0 до 100). Загальна тенденція: при малих обсягах ресурсу він дається переважно (рис. 3.4) підсистемам з високою ефективністю при малих навантаженнях, при збільшенні обсягу ресурсу відповідно – зменшуються частки «малих» підсистем і збільшуються частки «великих». Підсистема № 4 включається при обсязі ресурсу 40 одиниць і забирає більшу частину.

Однак цей перерозподіл «неперервно ступінчастий» - з розривами.

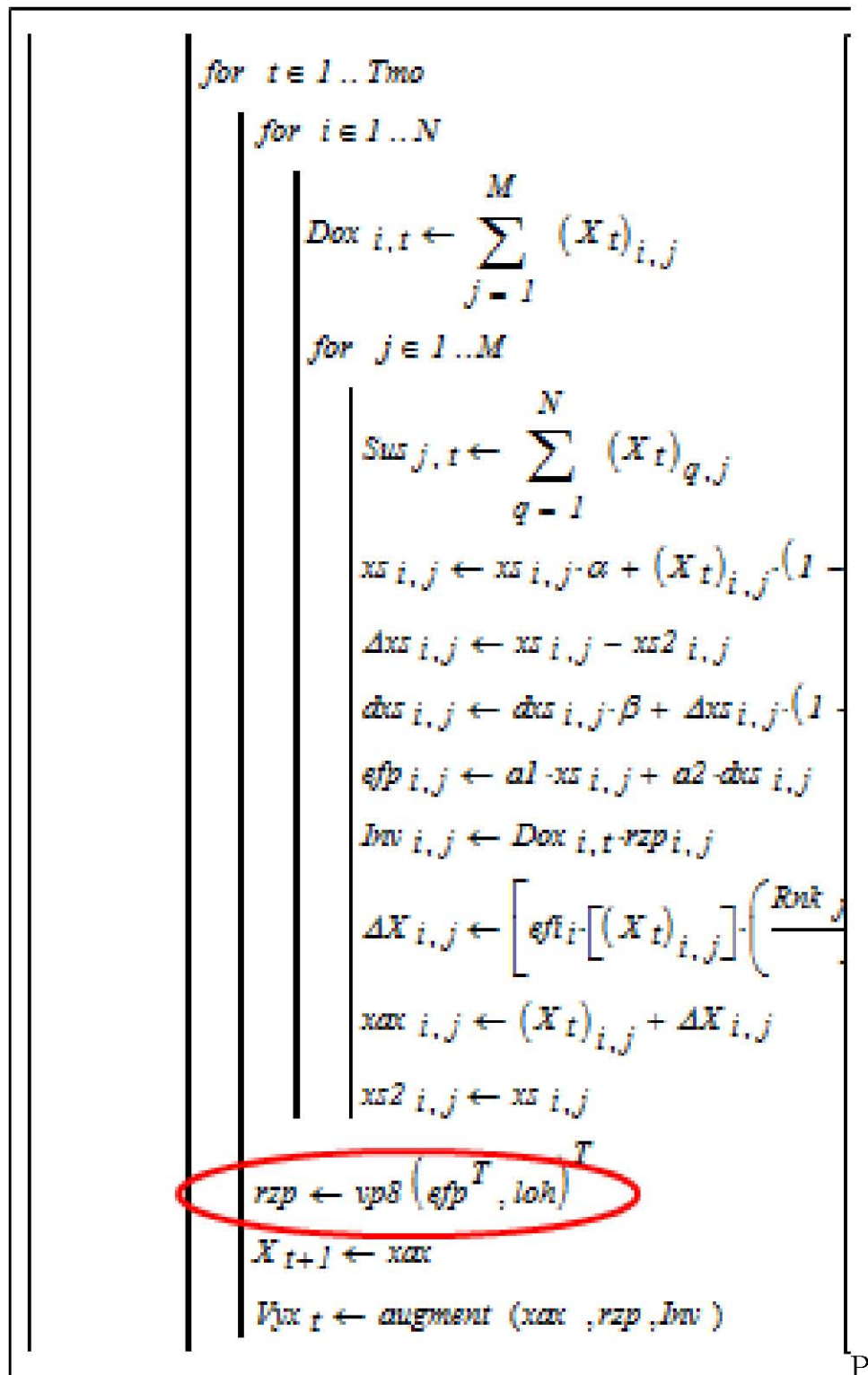


Рисунок 3.6 - Модуль «імітація випадкових подій»

Класичні методи такі задачі майже не вирішують.

В даній роботі для побудови комплексу засобів вибрано методи оптимального агрегування або для оцінювання ефективності. На рис. 3.3 - 3.5 подано незначну частку можливих результатів рішення задач оптимального агрегування.

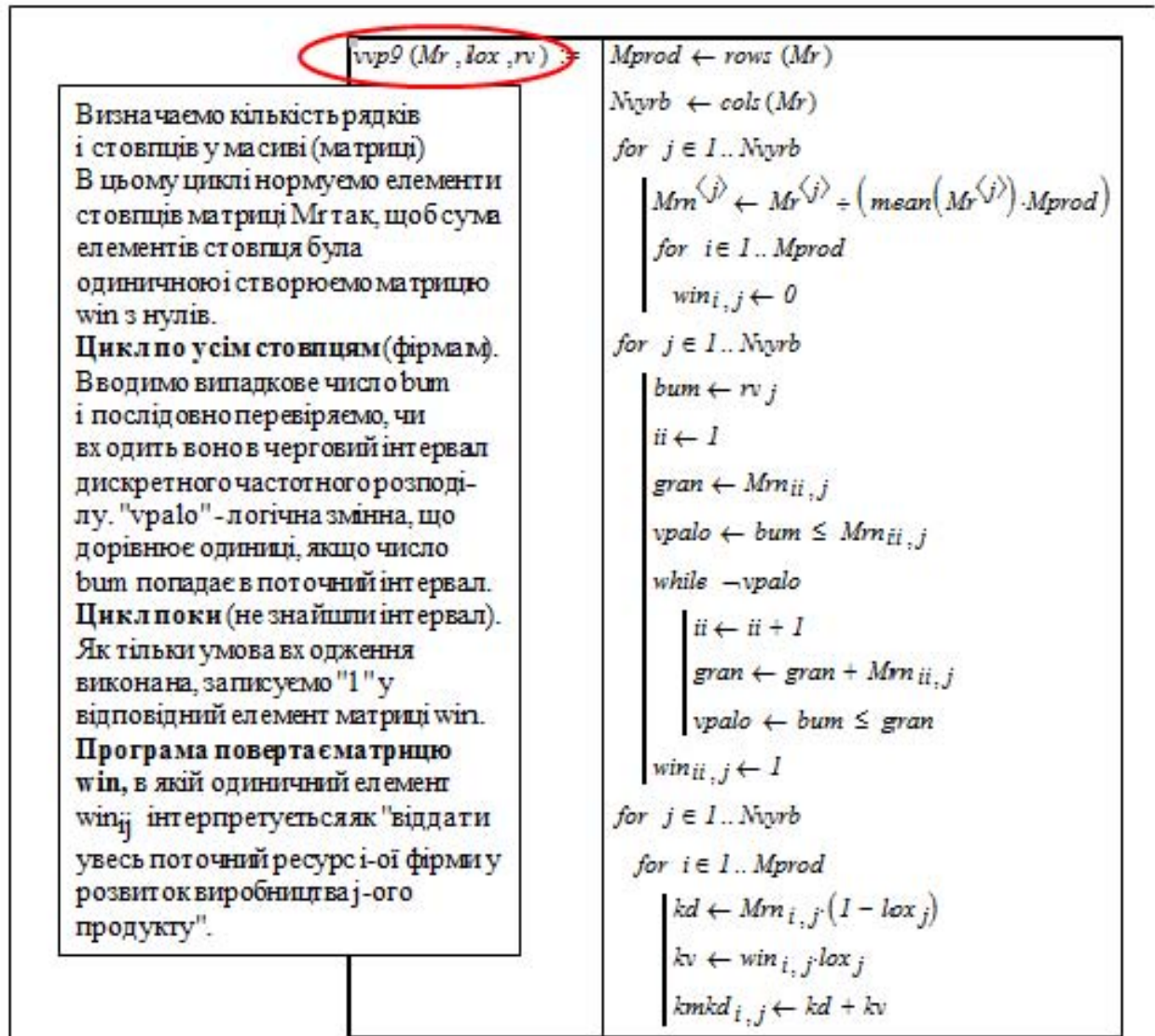


Рисунок 3.7 - Підпрограма імітації прийняття рішень в КС

Однак на прикладах рис. 3.5- 3.7, вирішується задача вищого рівня: маємо не точечне рішення, а функцію – ОЄФВ. Для кожного значення ресурсу отримується його оптимальний розподіл. Незавжно перевірити, що обчислювальні витрати (кількість запусків модуля «бінарний оператор оптимального агрегування» дорівнює розмірності системи мінус одиницю.

Збільшення точності теж проста задача: для збільшення точності вдвічі треба вдвічі збільшити вдвічі кількість точок дискретизації. Елементи виробничих систем можуть мати інші структури, якітосуються ресурсних зв'язків крім паралельної. Отримані оператори оптимального агрегування для таких структур "послідовні", "ресурсний зворотний зв'язок" (РОС) - англійською - "рециклінг" - перерозподіл вільних ресурсів та тихякі мають звільнитися – вони обслуговують інтегровану виробничу систему, де інтегроване власне виробництво і розвиток цього виробництва, ресурси витрачаються на виробництво "зараз" і прирощення виробничих потужностей, яке дасть прирощення вирибництва на наступному кроці процесу функціонування і розвитку.

В даній роботі ми модифікуємо програмнв модулі для імітації «механізмів» потужностей і чергових завдань для утворення наближеної до реальності тестової програми. Червоним колом виділено точку виклику підпрограми $vr5$, .. $vr9$. Цифра в імені номера версії . Сумісне використання детермінованих і ймовірнісних механізмів дозволяє відтворювати реальні системи і ситуації.

Подаємо приклади тестування. Розглядаємо систему «3 задачі, 4 обслуговувачих елемента». Задаємо початкові значення змінних:

елементи: пропорції розподілу: темпи швидкості: перспективність обслуговування:

$$X_{oo} \equiv \begin{pmatrix} 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 \end{pmatrix}; rzo \equiv \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0.5 \\ 0 & 0 & 0.5 \end{pmatrix} \quad tmpy \equiv \begin{pmatrix} 0. & 0. & 0 \\ 0 & 0. & 0 \\ 0. & 0. & 0 \\ 0 & 0 & 0 \end{pmatrix}; \quad nrmefe \equiv \begin{pmatrix} 10 \\ 10 \\ 10 \end{pmatrix}$$

Параметри задачі. Період моделювання кроків $Tmo \equiv 150$; дискретний час $tt := 1..Tmo$; крок моделювання $dT \equiv 0.1$; параметри усереднення: темпу обслуговування $\alpha \equiv 0.8$; прирощення обслуговування: $\beta \equiv 0.7$; пріоритету: темпу $a1 \equiv 1$; прирощення обслуговування $a2 \equiv 30$;

Потужності системи	Ефективність залучення інших елементів
$Rnk \equiv \begin{pmatrix} 3 \\ 5 \\ 8 \end{pmatrix} \begin{pmatrix} \text{продукт1} \\ \text{продукт2} \\ \text{продукт3} \end{pmatrix}$	$eff \equiv \begin{pmatrix} 1.2 \\ 1.1 \\ 1.0 \\ 0.9 \end{pmatrix} \begin{pmatrix} \text{виробник1} \\ \text{виробник2} \\ \text{виробник3} \\ \text{виробник4} \end{pmatrix}$

Розпаковуємо вихід: $By1^{(tt)} := (RNM_{tt})^{(1)}$; $By2^{(tt)} := (RNM_{tt})^{(2)}$;
 $By3^{(tt)} := (RNM_{tt})^{(3)}$

і обчислюємо показники розподілу запитів: сумарна продуктивність
 $sdox_i := \sum_j (RNM_{Tmo})_{i,j}$ та частки: $dr_{i,j} := \frac{(RNM_{Tmo})_{i,j}}{\sum_j (RNM_{Tmo})_{i,j}}$ для

кожного елемента. Обчислюємо швидкість обслуговування для кожного :

$$SD1_{tt} := By1_{1,tt} + By2_{1,tt} + By3_{1,tt} ; \quad SD2_{tt} := By1_{2,tt} + By2_{2,tt} + By3_{2,tt} ;$$

$$SD3_{tt} := By1_{3,tt} + By2_{3,tt} + By3_{3,tt}$$

$$SD4_{tt} := By1_{4,tt} + By2_{4,tt} + By3_{4,tt} .$$

На рис. 3.8 подано тестування для поданої програми.

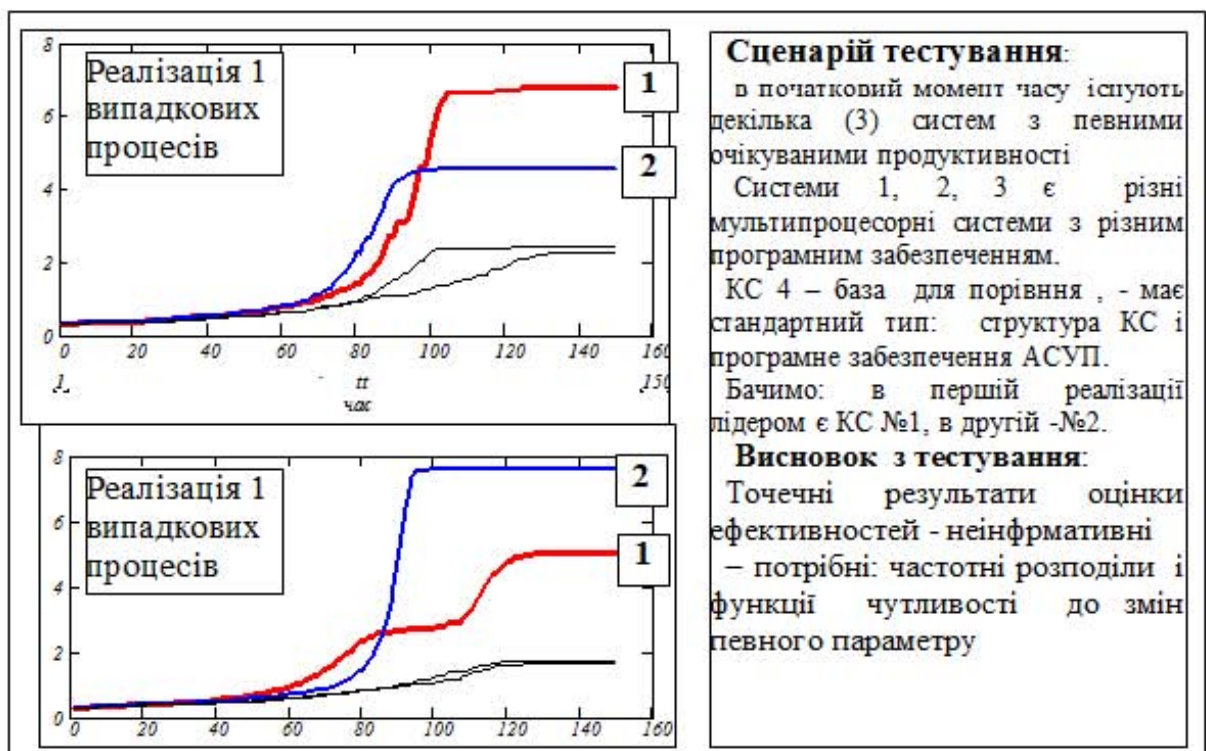


Рисунок 3.8 - Результати тестування різних програм управління

В даній роботі вибрано в якості тестової програми вибрано не рішення задачі лінійного програмування для мультипроцесорної КС (відома задача планування виробництва, транспортна задача та ін.), а рішення задачі нелінійного програмування – оптимального управління. Для наближення до реальності в імітаційну модель додані модулі імітації випадкових подій і збурень

На рис. 3.9 подано структурну схему розробки – певний підсумок попередніх розділів. Бачимо мультипроцесорну систему і об'єкт для яких розроблені показники ефективності. Також подано конкретні показники і компютеризований об'єкт.

Це все відображено в показниках



Рисунок 3.9 - Структурна схема розробки: комплекс засобів аналізу ефективності мультипроцесорних систем

3.3 Моделі оцінювання ефективності комп'ютерних систем з урахуванням розвитку та інновацій

В другому розділі розроблено теоретичні основи, виконано розробку і дослідження методів і моделей сучасних досить об'єктів з підсистемами виробництва і масового обслуговування. Інтенсивна глобальна конкуренція просто змушує такі системи створювати у себе підрозділи розвитку та інновацій. Слід обов'язково розглянути ці тренди. Отримана система моделей дозволяє виконувати оцінювання показників ефективності

саме для розширених систем «виробництво, розвиток інновацій оцінювати ризики та визначати впливи варіацій технічних параметрів в апаратних і програмних засобах комп'ютерних систем. Однак, в певних галузях головна особливість, що виділяє їх серед інших, є швидкий інноваційний розвиток, а саме (для телефонії):

- вдосконалення радіотехнічних характеристик каналів, характеристик стійкості до збурень і захисту від несанкціонованого доступу;
- мікроелектроніка на рівні елементів і типових модулів і функціональних модулів і підсистем, зокрема – управління мобільним зв'язком, системи GPS та ін.
- розширення спектру і якості послуг користувачам, зокрема, швидке розширення послуг, далеких від традиційної телефонії, а саме – створення малих форм обслуговування.

Ще один приклад - безпілотні системи управління на суші, морі, в космосі, зокрема, системи паркування авто в містах. Всі ці інноваційні напрямки вимагають побудови адекватних моделей, що могли б системно враховувати фактори і технологічного і ринкового розвитку.

Постановка задачі розробки моделей оптимального розвитку. Подаємо побудову і дослідження низки ускладнюючихся моделей розвитку та інновацій. Математичні моделі розвитку відносяться до класу варіаційних задач де змінні управління не числа, а функції - стратегії управління. Для варіаційних задач розвитку існує невелика кількість аналітичних розв'язань, а для числових розв'язань маємо проблему розмірності, досліджену Р. Беллманом. Метод динамічного програмування Беллмана давав певні можливості зменшити вплив розмірності задачі.

В даній роботі за основу покращення методів динамічного програмування та методу принципу максимуму вибрано методологію оптимального агрегування. Однак для агрегування структури «виробництво, розвиток, інновації» невідомі ефективні рішення. На рис. 3.10 подано

схему системи моделей СМЗ, що є об'єктом розробки в даній роботі і виділено блоки – об'єкти розробки і досліджень в даному розділі

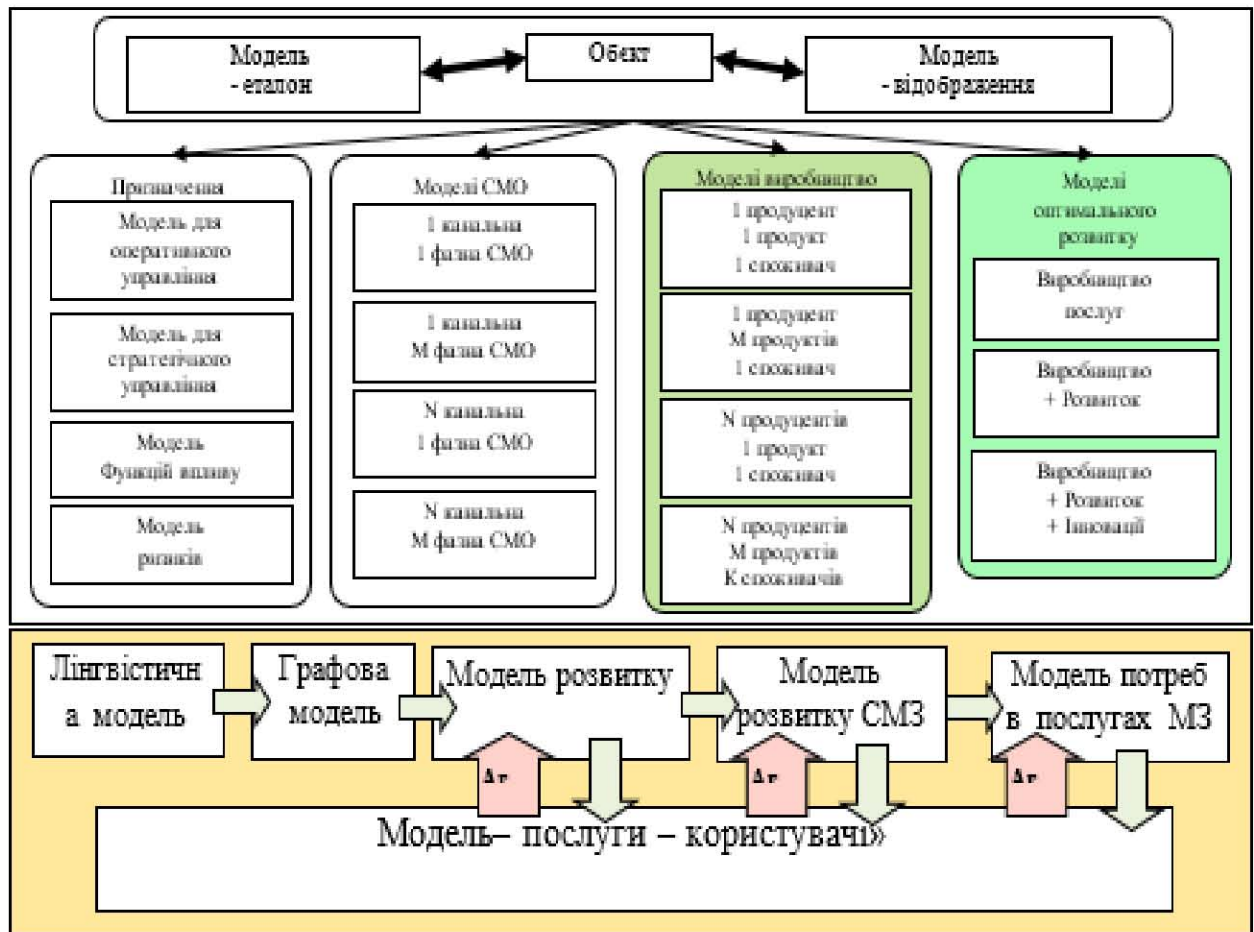


Рисунок 3.10 - Схема системи моделей для СМЗ і технології їх розробки.

Розвиток

Постановка проблеми. Очевидно, що не тільки неможливо, але і нерационально намагатися створити єдину модель, проте можливо створити систему узагальнених робочих моделей для окремих аспектів систем мобільного зв'язку. Ці моделі природно упорядкувати (стратифікувати) по рівнях опису – від схемотехніки до динаміки глобального ринку послуг мобільного зв'язку. Задовільне планування і прогнозування для великих комп'ютерних систем потребує багаторівневу систему моделей об'єкту. Сьогодні найменш розроблені моделі верхніх рівнів. Головна причина не стільки складність цих моделей, скільки їх міждисциплінарність. В системній моделі необхідно відобразити моделі взаємодії всіх аспектів функціонування

попиту, ринку видами послуг та ін.. В цьому повнні приймати участь саме розробники технічних засобів із залученням спеціалістів з психології, маркетингу тільки як консультантів.

Постановка варіаційної задачі оптимального розвитку. Першоджерелом для побудови моделі оптимального розвитку є дослідження Ричардом Беллманом "задач розвитку" в загальному виді та для окремих випадків.

Побудуємо базову одновимірну модель оптимального розвитку, а потім побудуємо систему моделей оптимального розвитку через розширення і уточнення базової моделі. На рис. 3.11 подана узагальнена схема функціонування і розвитку з урахуванням впливу зовнішнього середовища. В цій моделі подано:

- представлення елементів системи технологічними перетворювачі інформаційних і матеріальних ресурсів в інформаційні продукти, – моделями «витрати - випуск»;
- представлення системи як замкнутої динамічної системи
- використання методу оптимального агрегування, що дозволяє зняти звичайні обмеження на розмірність системи і вид функцій «витрати - випуск». На рис. 3.11 представлена схема агрегованої системи як об'єкту, що розвивається.

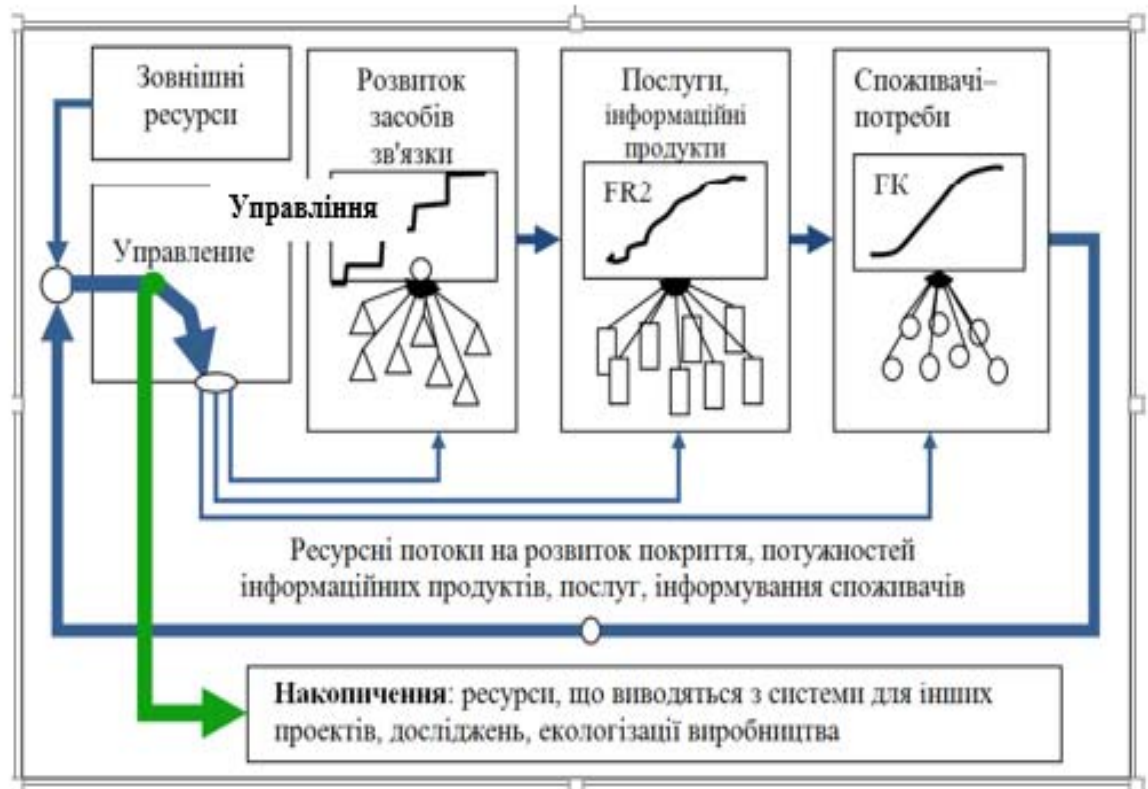


Рисунок 3.11 - Схема агрегованої системи мобільного зв'язку як об'єкту розвитку

На схемі умовно показано агрегування паралельно працюючих технічних засобів, використовуваних видів послуг і користувачів. Конкретизуємо використані в роботі терміни щодо об'єктів даної роботи.

Ресурси - матеріальні, інформаційні та інші ресурси згідно Дж. Форрестеру, що задіяні в процесах функціонування і розвитку. Власні і зовнішні ресурси - системне аналітичне поняття, залежне від того, як визначені межі системи.

Продукт - результат матеріального, інформаційного та інтелектуального виробництва, який знайшов користувачів, або внутрішнє використання, конкретно це: апаратні засоби, програмне забезпечення, послуги мобільного зв'язку та ін.

Виробничий елемент - в даній роботі - радіотехнічні модулі, що забезпечують покриття певної території. Для виробничого елемента визначаємо:

- функцію виробництва (ΦB) - залежність між кількістю "продукту" і необхідними для цього ресурсами, що може бути неперервною, дискретною, детермінованою, імовірнісною і нечіткою; - функцію розвитку (ΦP) - залежність між прирощенням виходу "продукту" і необхідними для цього ресурсами. На рис. 3.12 подана система моделей розвитку. На базі цієї схеми деталізуємо схему системи моделей, подану на рис. 3.11 і отримуємо систему моделей, що складається з багаторівневої системи ускладнюючихся моделей (редукційна декомпозиція) та систем функціональних моделей на кожному рівні.

Запишемо відповідно до схеми на рис. 3.12 математичні моделі трьох рівнів. Типова система мобільного зв'язку є розподіленою територіально і «багатопродуктовою», тобто кожна територіальна підсистема $i = 1..N$ надає певний спектр послуг $j = 1..M$. Формально задача визначення оптимальної стратегії розвитку зводиться до визначення $N \cdot M$ оптимальних стратегій розвитку для кожного виду послуг в кожній підсистемі, тобто маємо варіаційну задачу великої розмірності.

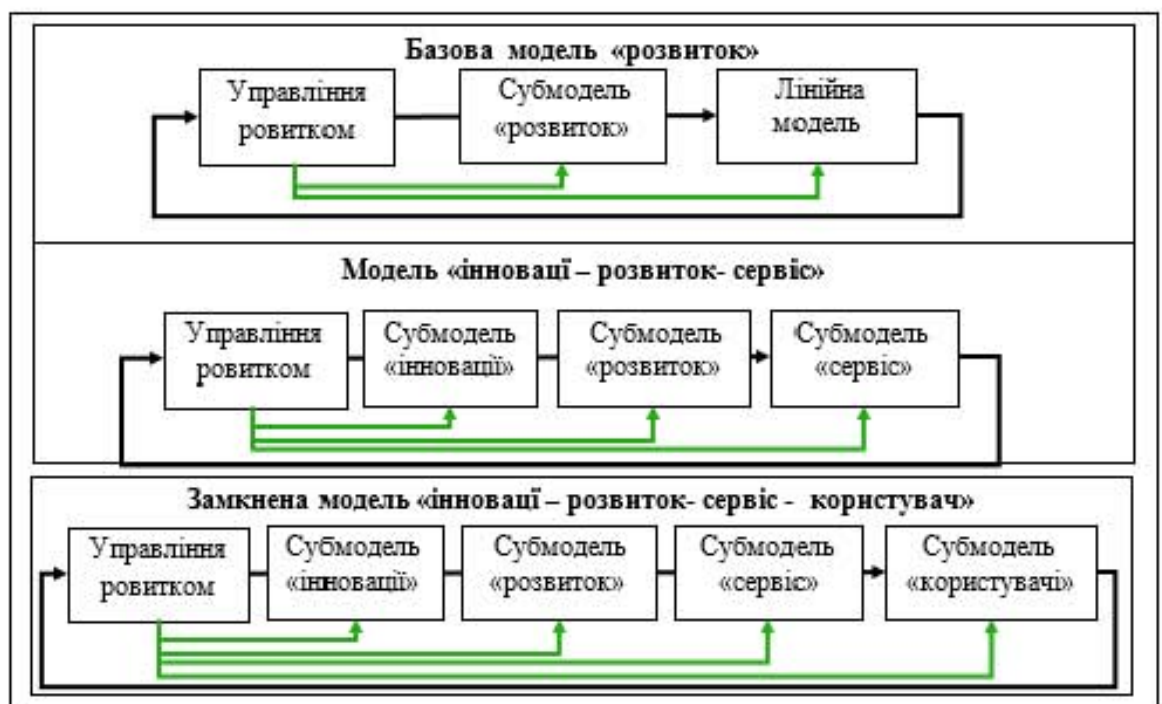


Рисунок 3.12 - Система моделей розвитку системи мобільного зв'язку

Під час пошуку і аналізу аналогів виявлена особливість більшості робіт, яку можна назвати «рання формалізація» задачі оптимізації. Суть її в тому, що згідно деталізованому опису складної системи записується система рівнянь великої розмірності, а потім для цієї системи шукаються оптимальні розв'язання, потужними і витонченими пошуковими методами, що не знімають проблему розмірності і не працюють з невивуклими і розривними функціями.

Альтернативний підхід - фундаментальний аналіз властивостей складної системи, виявлення діючих в системі «поборжуючих механізмів» - законів механіки, електродинаміки, теорії імовірностей та ін. На цій основі виділяються задачі «субоптимізації» - оптимізації певних змінних, що не залежать від системи в цілому, оптимізації частини процесу чи системи з вільною змінною для задоволення системним обмеженням при додаванні чергової частини процесу або чергового елемента системи. Перш, ніж шукати оптимальне рішення варіаційної задачі розвитку проаналізуємо об'єкт оптимізації - розподілену багатопродуктову систему.

Метод динамічного програмування Беллмана базується на принципі оптимальності: яким би шляхом не прийшла систему в певну точку фазового простору, подальша оптимальна стратегія повинна бути оптимальною тільки відносно даної точки. Подібне сформулюємо для наших задач оптимізації продукційних систем «витрати-випуск»: скільки б ресурсу не було виділено для певної підсистеми даної системи, його розподіл повинен бути оптимальним за критерієм «витрати-випуск».

Використаємо метод оптимального агрегування, що є незвичним – він повністю алгебраїзований, безпошуковий, малочутливий до розмірності оптимізаційної задачі, і має тільки одне обмеження для функцій «витрати – випуск»: нестрога позитивність і нестрога монотонність.

Методику застосування методу оптимального агрегування подамо на конкретному прикладі з тестовими числовими значеннями параметрів

розподіленої багатопродуктової системи. Причина використання прикладу - матеріальні та інформаційні технології, операції над структурами даних - невідемна і важлива частина методу:

1. Визначення ФР усіх елементів дворівневої системи
2. Дискретизація і подання всі ФР в стандартній формі елементів алгебри
3. Виконання однорівневого і дворівневого оптимального агрегування
4. Отримання нового об'єкта алгебри функцій «витрати – випуск» - оптимальної у еквівалентній ФР СМЗ і відповідної вектор-функції розподілу ресурсу.

Аналіз точного і наближених рішень задачі розвитку системи. Розглянемо стандартну процедуру постановки і отримання рішення варіаційної задачі розвитку - для виявлення і усунення проблемних моментів методу. - роботи Беллмана - для лінійних функцій розвитку і монографія, де рішення отримано для випадку довільної ФР і лінійної ФВ. Подаємо постановку базової задачі.

Підставляємо знайдені рішення для спряжених функцій [8] отримуємо вираз для функції Гамільтона. Згідно з методом принципу максимуму оптимальне управління процесом розвитку для кожного моменту часу процесу визначається як таке, що дає максимум функції Гамільтона.

Запишемо в дискретному вигляді остаточний вираз для функції Гамільтона і порівняємо його з апроксимацією в просторі стратегій [9].

$$H_k(x_k, u_k) = x_k \cdot (1 - u_k) + \text{fin}(x_k \cdot u_k) \cdot \psi_n(u_k, x_k, d\text{fin}, t);$$

$$H_0(x_k, u_k) = x_k \cdot (1 - u_k) + \text{fin}(x_k \cdot u_k) \cdot (T - t).$$

В загальному випадку вирішення варіаційної задачі розвитку при умові існування неперервних похідних зводиться до розв'язання системи диференціальних рівнянь для змінних стану, спряжених функцій та змінних управління У матрично-векторній формі маємо:

$$\frac{d}{dt}x = f(x, u); \quad \frac{d}{dt}\psi = fs(\psi, u); \quad \frac{\partial}{\partial u}H(x, u) = 0, \text{ де}$$

x - вектор стану, u - вектор управління, ψ - вектор спряжених змінних.

$f(x, u)$ - праві частини дифрівнянь стану, $fs(\psi, u)$ - праві частини спряжених дифрівнянь; $H(x, u)$ - функція Гамільтона. При наявності обмежень типу нерівностей шукається замість нульового значення максимум функції Гамільтона. В методі принципу максимуму Л. Понтрягіна береться неповний гамільтоніан - без складових для управлінь. і шукається його максимум. Початковою областю застосування методу принципу максимуму були задачі з критерієм - час перехідного процесу при обмеженому рівня управління. Для такого класу задач оптимальні управління приймають фіксовані значення $(-u_{max}, 0, u_{max})$. Наявність інтервалів знакопостійності для управлінь суттєво спрощує задачі з критерієм "час перехідного процесу". В цьому напрямку напрацьована велика кількість робіт.

Для задач розвитку з інтегральним критерієм класу "накопичений ефект" майже єдині узагальнені моделі і методи були розроблені Р. Беллманом, який дослідив так звану "задачу розподілу" і довів, що за умови неперервних похідних у функцій розвитку і підінтегральних функцій критерію рішення варіаційної задачі розподілу - оптимальна стратегія, має три інтервали: "все в розвиток", ейлерева ділянка та "все в накопичення".

На першому і третьому інтервалах оптимальне управління дорівнює $u(t) = 1, u(t) = 0$, відповідно. На другому інтервалі оптимальне управління змінюється $0 \leq u_k \leq 1$. В дослідженнях Беллмана головна увага приділена визначенню умов існування цих інтервалів та визначення точок переходу між інтервалами. Проблема розмірності у Беллмана вирішувалась так: отримання рішення для задачі з однією змінною, потім - з двома змінними стану, і, нарешті - вирішення задачі довільної розмірності.

Аналогічний підхід застосовувався для при урахування нелінійностей цільових функцій і обмежень: вирішення задачі для лінійної функції, потім - для ступеневої чи логарифмічної, потім узагальнення для монотонних функцій. [Беллман]. Для задач інноваційного розвитку конкретні методики Беллмана просто не можуть бути застосовані з причин невизначеностей функцій обмежень, необхідності враховувати ступінчастий характер узагальнених функцій розвитку (зміна потужностей дискретно - стандартними модулями. Це спричиняє необхідність застосування числових методів, слабке місце яких - проблема розмірності. Використовуємо ресурсний підхід, коли елементи СМЗ розглядаємо як технологічні перетворювачі узагальнених ресурсів в узагальнений продукт. Постулювання монотонності цих функцій дало можливість обґрунтувати і застосувати метод оптимального агрегування, що дозволяє замінити складну багатовимірну СМЗ оптимальним еквівалентним одновимірним елементом. За рахунок того, що оператор оптимального агрегування базується на знаходженні послідовності пошуку екстремумів функції однієї змінної, метод оптимального агрегування не має проблем розмірності – обчислювальні витрати зростають не більше ніж лінійно при зростанні розмірності задачі. «Не більше ніж лінійно» означає розпаралелювання виконання одновимірних задач оптимізації.

Це дозволяє для пошуку екстремуму використати метод прямого перебору на фіксованій сітці, що знімає вимоги лінійності, неперервності, випуклості для узагальнених функцій класу «ресурс, продукт».

В методі принципу максимуму варіаційна задача визначення екстремуму функціонала замінюється послідовністю одновимірних задач знаходження функції Гамільтона (для еквівалентної оптимальної одновимірної системи). Таким чином, ми розбили задачу оптимального розвитку СМЗ на дві задачі - оптимального агрегування підсистем СМЗ в оптимальний еквівалентний елемент та варіаційну задачу оптимального

розвитку об'єкта «оптимальний еквівалентний елемент». Ці задачі ефективно виконуються на сучасних процесорах і мультипроцесорних системах

3.4 Аналіз результатів моделювання процесів розвитку системи обслуговування

Модель оптимального розвитку реалізована в середовищах ефективних пакетів для моделювання, відлагоджена і відтестована на цій моделі проведені ряд досліджень, що підтвердили адекватність моделі і дозволили розробити більш складну точну, але і більш точну модель інноваційного розвитку.

При використанні числових методів для довільної варіаційної задачі для розподіленого об'єкта фактично треба знайти екстремум функції $N \times K$ змінних (число підсистем \times число кроків процесу). Використання методу оптимального агрегування дозволяє зняти «половину» проблем моделювання і оптимізації систем великої розмірності - знаходження екстремуму функції тільки K змінних.

З позицій класичного підходу до математичного моделювання, модель інноваційного розвитку побудувати неможливо. Загальну методологію моделювання і оптимізації за наявності невизначеностей запропонував Р. Беллман. Елементи цієї методології: побудова «сходів» моделей для складної системи, коли попередня модель служить першим наближенням для наступної; заміна завдання вибору крапки в багатовимірному фазовому просторі послідовністю завдань вибору в просторах меншої розмірності. Методологія Беллмана не вирішує всі проблеми, але створює можливості для поетапного створення моделі ще не існуючого реального об'єкта чи проєкта.

У вузькій формальній постановці модель є відтворенням суттєвих для користувача властивостей реального об'єкта. Побудова моделей ще не існуючих - майбутніх об'єктів і процесів набула широкого розповсюдження завдяки можливостям комп'ютерної техніки. Для формалізації таких

моделей треба розширити визначення моделі: від моделі одиничного об'єкта перейти до параметричного класу класу моделей. Для технічних систем існує стабільний показник – коефіцієнт новизни - частка існуючих елементів в новій системі. Типові значення коефіцієнта – 0.80 – 0.95, тобто власне нового в нових системах 5-20%. Наприклад, перші космічні апарати не мали прямих прототипів, але вони склалися з існуючих, взятих від інших технічних систем елементів.

Подібні закономірності існують для інформаційних продуктів і систематичних моделей. Тому можливо створити потенційно адекватну модель інноваційного об'єкта базуючись на аналогах та достовірно виявлених «породжуючих механізмів - законів фізики, електродинаміки, термодинаміки ..

Шляхи вирішення проблеми. Для вирішення проблеми потрібно побудувати модель, що відображає істотні особливості процесу інноваційного розвитку виробництва матеріальної, інформаційної продукції та послуг, вивчити властивості моделі і з урахуванням цих властивостей розробити метод оптимізації, що не має обмежень по лінійності, випуклості, неперервності. Дотримуємось методології оптимального агрегування заміни складної системи еквівалентним оптимальним елементом, для якого вирішуємо задачу оптимального розвитку, і нарешті отримуємо оптимальні управління розвитком для кожного елемента за допомогою операції дезагрегування.

Базова модель повинна бути робочою, тобто такою, що виконується в середовищі певної програмної платформи і відкритої для уточнення, розширення, створення нових моделей.

На початку розділу система мобільного зв'язку була подана як послідовне поєднання підсистем «інновації», «розвиток» і «надання послуг» . На рис. 3.13 представлена конкретизована схема сСМЗ з урахуванням функцій розвитку та створення інновацій.

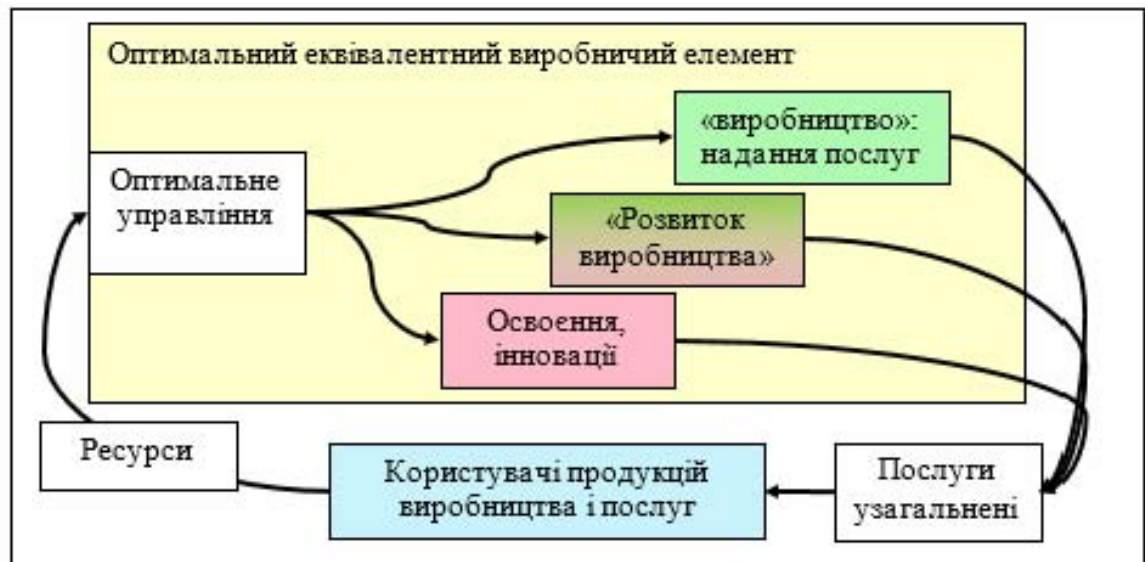


Рисунок 3.13 - Схема СМЗ як об'єкту інноваційного розвитку

Згідно схемі на рис.3.13 конкретизуємо постановку задачі:

- в якості змінних управління вибираємо розподіл ресурсів між підсистемами;
- зв'язки між підсистемами – підсистема «інновації» змінює параметри функції «вхід-вихід» підсистеми «розвиток», підсистема «розвиток» змінює параметри функції «вхід-вихід» підсистеми «виробництво послуг»;
- враховуємо зворотний зв'язок через підсистему «користувачі послуг»;
- мета розробки - заміна виробничої системи оптимальним еквівалентним елементом та постановка і розв'язання задачі оптимального розвитку для еквівалентного оптимального елемента.

Прийняте нами розділення системи на три рівні – певна ідеалізація тому, що в реальності підсистеми «інновації», «розвиток» можуть бути незалежними бізнес – одиницями, і власне виробництво може бути розподіленим і децентралізованим. Проте, наша модель - засіб для алгебраїзації задач аналізу і синтезу СМЗ з урахуванням функцій інновацій і розвитку.

У цій статті робимо модуль оптимізації на базі цієї альтернативи.

Сценарій алгоритму оптимізації:

- задаємо величину приросту ресурсу (тестовий квант) і розподіл ресурсу між трьома підсистемами - інновації, розвиток, темп виробництва кінцевого продукту, задаємо критерій ефективності системи в цілому (простий - випуск продукції);
- ділимо тестовий квант в пропорціях початкового розподілу, обчислюємо критерій, потім послідовно «віддаємо» однаковий приріст ресурсу кожній підсистемі, обчислюємо прирости критерію;
- коректуємо початковий розподіл ресурсу: задаємо квант перерозподілу ресурсу – у «відстаючих» підсистем забираємо цей квант і віддаємо «лідерові».
- Повторюємо цей набір операцій необхідне число разів.
- Якщо цей цикл повнорить для безлічі значень ресурсу, отримаємо
- оптимальну еквівалентну функцію виробничої системи;
- вектор-функцію оптимального розподілу ресурсів.

Звернемо увагу на червоне слово «послідовно» - ми робимо «обчислювальні методи» оптимізації так, щоб не шукати а гарантовано знаходити екстремум, і в сучасних обчислювальних системах це «послідовно» може бути виконано паралельно. Доведення збіжності і оптимальності базується на монотонності і обмеженості виробничих функцій. Ми робимо базову модель, альтернативи оптимальному агрегуванню, що є альтернативою оптимальному агрегуванню структури на базі оператора «виробництво розвиток». Ця альтернатива поки не має узагальнено рішення. Подаємо програму як послідовність модулів, які можна змінювати і замінювати незалежно від інших (рис. 3.14).

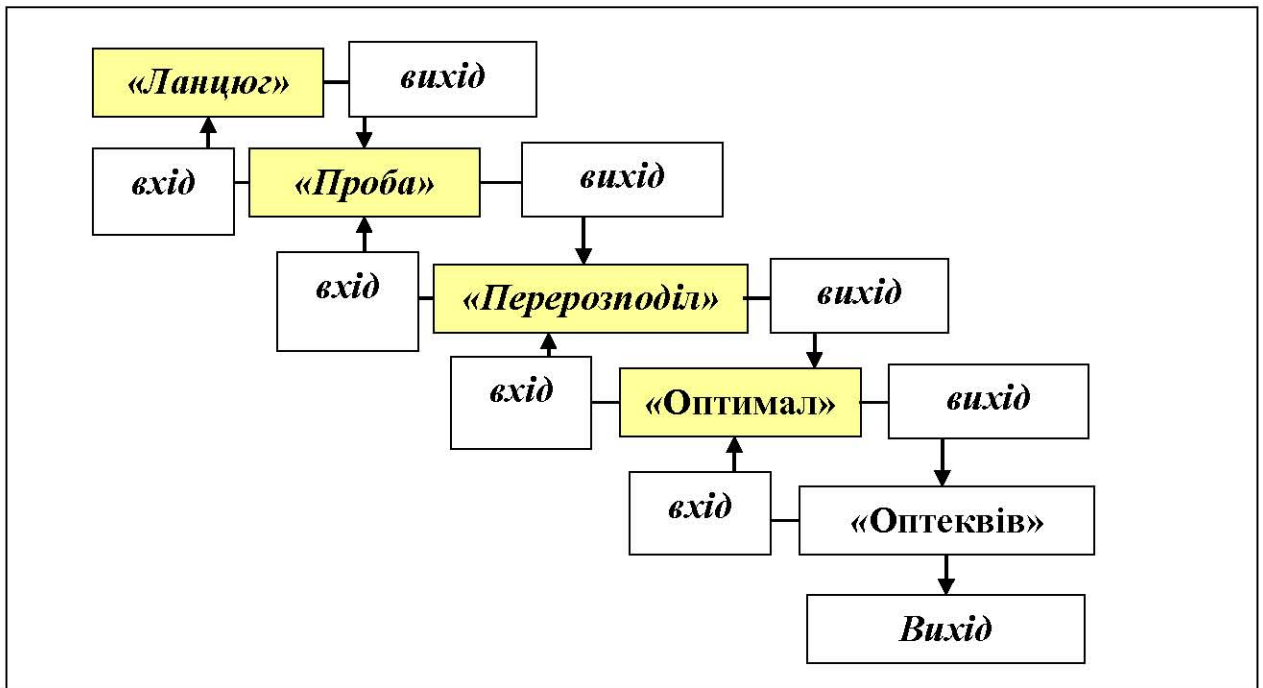


Рисунок 3.14 - Структура системи для оптимізації системи «інновації, розвиток, виробництво»

Модуль "ланцюг" $OsRzPr(vRs, Mpse)$ бере структуру "об'єм і розподіл ресурсу" vRs і структуру "параметри системи" $Mpse$, а повертає нові значення параметрів функцій освоєння (інновацій), розвитку і виробництва. Модуль – просто послідовність операцій алгебри. Всі проблеми в складових $kz1$ і zrP . Суть модуля: реакція трирівневої системи із заданими параметрами на вхід величина м розподіл ресурсу.

Модуль "проба" $Prob(vRs, Mpse, dlt)$ бере

Нижче представлена версія модуля "проба". Особливість модуля - використання технологічного масиву $plek$ Сценарій роботи модуля: задана трирівнева система і вхідний розподіл ресурсів. Потрібно визначити вихід системи тестових розподілів ресурсаеме - інновацій, розвитку, виробництва.

Вихід формований з векторів нормованих і ненормованих значень виходу виробничої системи при всіх варіантах розподілу приросту. Програма використовує підпрограму "ланцюжок" - $OsRzPr(vRs, Mpse)$.

Бачимо, що в даному випадку більше продукції отримуємо, якщо допресуре “кинемо» в розвиток, на другому місці - не міняти пропорцію. . При інших параметрах виробничої системи результат може бути іншим.

Модуль "переділ" бере

Сценарій роботи модуля: задана трирівнева система і вхідний розподіл ресурсів. Задаємо приріст ресурсу для системи, і визначуваний вихід системи для таких розподілів: 1) розподілити в колишніх пропорціях, 2, 3,4) по черзі віддати кожній підсистемі - інновацій, розвитку, виробництва.

Мінятимемо розподіл ресурсів так: лідерові - даємо квант ресурсу, забираємо квант у двох інших пропорційно величинам попереднього розподілу. Величина кванта визначається величиною прирости у лідера. Дивимося, бачимо у лідера - на 25% більше номінала і майже удвічі проти інших альтернатив

Запишемо три рівняння перерозподілу ресурсів

$$vRs2_{bols1,1} := (vRs1)_{bols1,1} \cdot vRs2 - bols1,3 \cdot kra$$

$$vRs2_{bols2,1} := (vRs1)_{bols2,1} \cdot vRs2 - bols2,3 \cdot kra$$

$$vRs2_{bols3,1} := (vRs1)_{bols3,1} \cdot vRs2 + 1 \cdot kva$$

Враховуємо розподіли з нульовими складовими, збираємо розглянуті елементи в модуль (це рутинна на 95%).

Приводимо структури входу і виходу модуля для цих даних. Кінцева мета використання складних структур даних зведення програми обробки даних до псевдоодновимірної форми:

"вихідна структура = оператор від вхідної структури"

з великою швидкістю і можливістю алгебраїзації.

Модуль "оптимал" бере

Сценарій роботи модуля: задана трирівнева система і вхідне розподілене ресурсів. Задаємо приріст ресурсу для системи, і визначуваний вихід системи для таких розподілів: розподілити в колишніх пропорціях, по черзі віддати кожній підсистемі - інновацій, розвитку,

виробництва. Задаємо число кроків $kto := 46$ $j := 1..kto$ $dlt := 0.05$ нове випадкове $kkn \equiv 1$ Тестуємо програму, записуємо в масиви вихід модуля $Dym := OpTo(vRs, Mpse, dlt)$ $Mym^{(j)} := Dym_{1,j}$

Аналіз роботи модуля оптимізації. Значення інтерфейсу - досліджувати програму на предмет синтаксис-ческой і семантичній коректності. Це необхідний компонент розробки нової моделі і програми, тому, що не має прототипів.

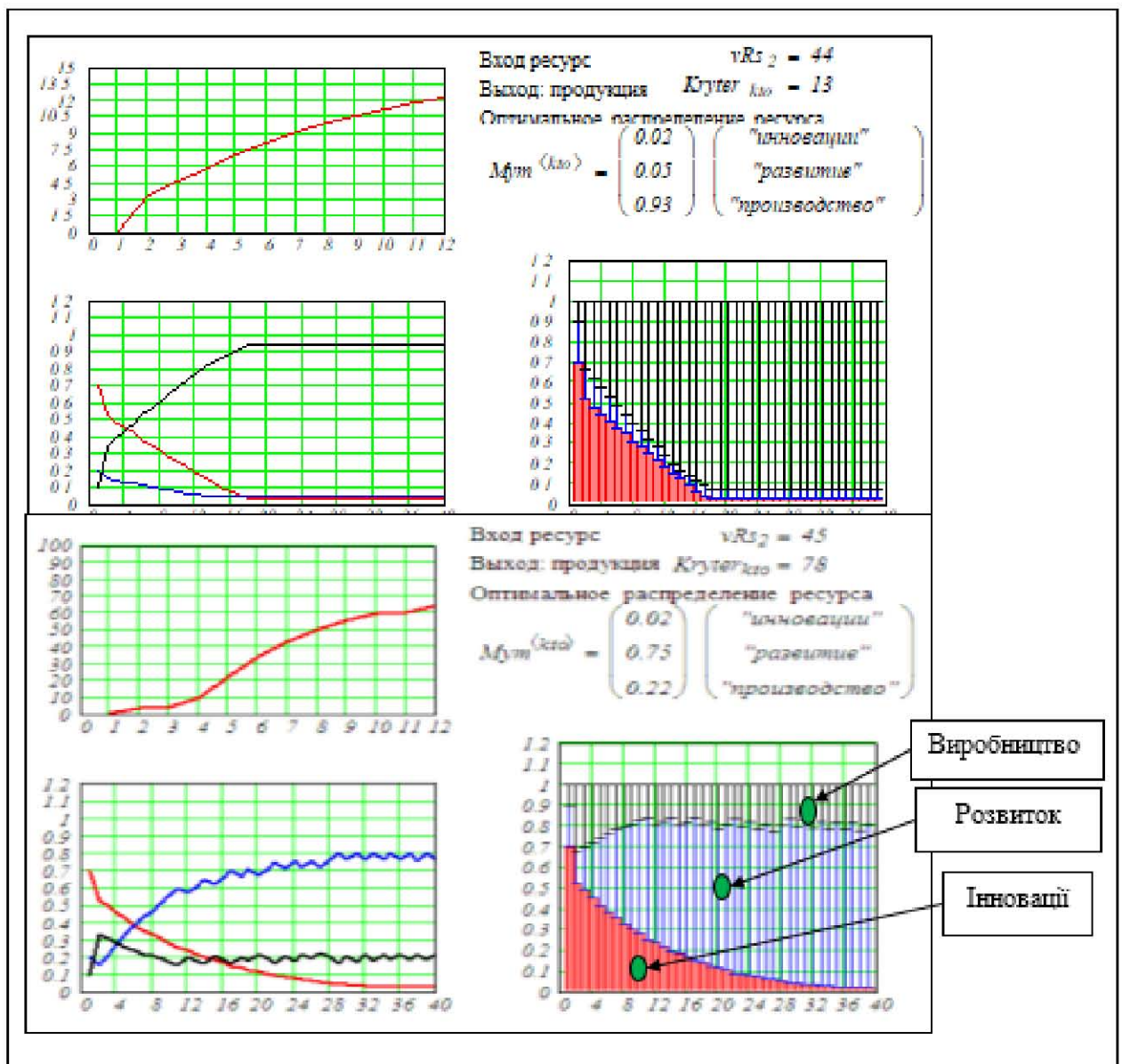


Рисунок 3.15 - Аналіз процесів оптимізації розподілу ресурсів.

На рис. 3.16 подано приклад оптимального агрегування для тестової системи. З багатьох результатів відібрано незвичний, але такий, що не суперечить реальним породжуючим механізмам

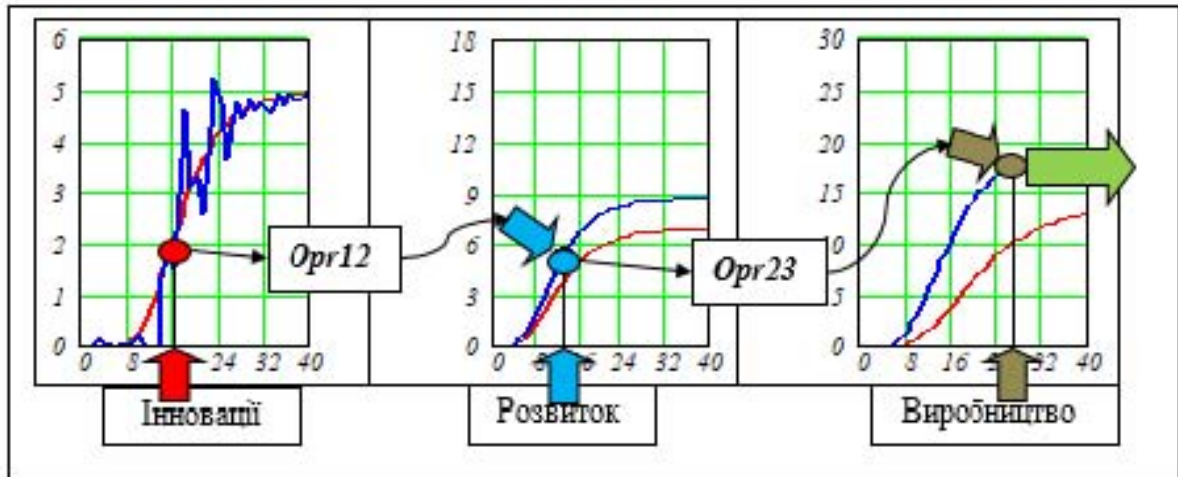


Рисунок 3.16 - Приклад розрахунку оптимальної еквівалентної функції виробництва для системи «інновації, розвиток, виробництво послуг»

На цих графіках - приклад розрахунку еквівалентної оптимальної виробничої функції. На верхніх графіках - "причина" - виробничі функції підсистем. Наступний графік - оптимальна еквівалентна виробнича функція. Нижня пара графіків - вектор-функція оптимального розподілу ресурсів в двох формах «простий» і «в приростах». Червоне – інновації, синє - розвиток, чорний, – виробництво.

Щодо заданої математичної моделі і її параметрів результат коректний. Але чи може бути таке по життю (у реальності)?

Так це: дивимося ще вище - на графіки функції виробництва "до того" і "після" (вливання в систему певної кількості оптимально розподіленого ресурсу. Бачимо: при тих же витратах можна випустити більше продукції. Зрозуміло, що системи з процесами розвитку, освоєння, інновацій і, навпаки, старіння, спрацювання ресурсу потрібні нові, адекватні реаліям, моделі розподілених систем.

Саме в підрозділі 3.4 подано приклади розробки і тестування систем «виробництво, розвиток, інновації».

4 ЕКОНОМІЧНА ЧАСТИНА

4.1 Оцінювання комерційного потенціалу розробки

Метою проведення технологічного аудиту є оцінювання комерційного потенціалу розробки, створеної в результаті науково-технічної діяльності.

Результатом магістерської кваліфікаційної роботи «Засоби аналізу ефективності моделювання мультипроцесорних систем з розподіленою пам'яттю» є розробка програмного забезпечення. Для проведення технологічного аудиту залучено трьох незалежних експертів. У нашому випадку такими експертами є: Колесник Ірина Сергіївна (к.т.н., доцент каф. обчислювальної техніки ВНТУ), Ткаченко Олександр Миколайович (к.т.н., доцент каф. обчислювальної техніки ВНТУ) та Черняк Олександр Іванович (к.т.н., доцент каф. обчислювальної техніки ВНТУ).

Оцінювання комерційного потенціалу буде здійснене за критеріями, що наведені в таблиці 4.1.

Таблиця 4.1 – Критерії оцінювання комерційного потенціалу розробки бальна оцінка

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
ри-тер.	0	1	2	3	4
	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів
	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					

Продовження таблиці 4.1

	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
Практична здійсненність					
	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витратити значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
0	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві
1	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років

Продовження таблиці 4.1

2	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту
---	---	--	---	--	---

Результати оцінювання комерційного потенціалу експертами розробки зведено в таблицю 4.2.

Таблиця 4.2 - Результати оцінювання комерційного потенціалу розробки

Критерії	Прізвище, ініціали, посада експерта		
	1 – Колесник	2 – Ткаченко	3 – Черняк
	Бали, виставлені експертами:		
1	4	3	4
Ринкові переваги (недоліки):			
2	3	3	3
3	4	3	4
4	4	4	4
5	4	4	3
Ринкові перспективи			
6	4	3	3
7	4	3	3
Практична здійсненність			
8	4	3	3
9	3	2	3
10	4	3	4
11	4	3	4
12	3	4	5
Сума балів	СБ ₁ =45	СБ ₂ =38	СБ ₃ =43
Середньоарифметична сума балів $\overline{СБ}$	42		

За даними таблиці 4.2 можна зробити висновок, щодо рівня комерційного потенціалу розробки. Зважимо на результат й порівняємо його з рівнями комерційного потенціалу розробки, що представлено в таблиці 4.3.

Таблиця 4.3 – Рівні комерційного потенціалу розробки

Середньоарифметична сума балів $\overline{СБ}$, розрахована на основі висновків експертів	Рівень комерційного потенціалу розробки
0 – 10	Низький
11 – 20	Нижче середнього
21 – 30	Середній
31 – 40	Вище середнього
41 – 48	Високий

Рівень комерційного потенціалу розробки, становить 42 бали, що відповідає рівню «високий».

Галузь розробки ПЗ одна з найважливіших сфер застосування сучасних інформаційних технологій. Зручне програмне забезпечення дозволяє аналізувати перспективи створення спеціалізованих мультипроцесорних систем з розподіленою пам'яттю, визначати перспективні об'єкти, які потребують для обслуговування саме таких систем .

В якості аналога для розробки було обрано LINPACK, це програма, призначена для вирішення системи лінійних рівнянь алгебри з щільною матрицею з вибором головного елемента по рядку використовується для формування списку Top500 – п'ятисот наймогутніших комп'ютерів світу.

Основним недоліком є те, що програма розпаралелюється, тому неможливо оцінити ефективність роботи комунікаційного компоненту суперкомп'ютера.

Також до недоліків можна віднести те, що програмний засіб є платним.

У розробці дана проблема вирішується, адже за допомогою створеного програмного забезпечення є можливість ефективно та швидко підналаштувати параметри безпосередньо під вимоги замовника та можливості розробника. Також програмний продукт випереджає аналог за таким параметром як доступність.

У таблиці 4.4 наведені основні технічні показники аналога і нового програмного продукту

Таблиця 4.4 - Основні технічні показники аналога і нового програмного продукту

Показники	Аналог	Нова розробка	Відношення параметрів нової розробки до параметрів аналога
Функціональність	набір з 24 Ліверморських циклів і пакет NAS тестів, що відображають різні сторони реальних програм обчислювальної гідродинаміки.	функціонування системи при різних навантаженнях та можливих відмовах певних елементів	7/10
Надійність	Забезпечення безперебійної роботи при заданих умовах	Забезпечення безперебійної роботи при заданих умовах	10/10
Сумісність	Велика сумісність	Максимальна сумісність	8/10
Супровід	Уже знайдено методів оптимального узгодження структур комп'ютерної системи і об'єкта управління	Структура майбутньої комп'ютерної системи підлаштовується з використанням математичного моделювання до об'єкта управління	5/8
Економія ресурсів і часу	Програмне забезпечення виконує необхідні функції, використовуючи допустиму кількість ресурсів у визначений відрізок часу	Програмне забезпечення виконує необхідні функції, використовуючи допустиму кількість ресурсів у визначений відрізок часу	10/10
Простота використання	Зручний інтерфейс	Більш зручний інтерфейс	8/9

Нова розробка має доволі простий інтерфейс, на відміну від аналогу, адже для недосвідченого користувача інтерфейс аналогу є ускладненим.

Програмний продукт швидко і ефективно виконує функції прийняття рішень з вибору можливих, або з модифікації комп'ютерної системи для певного об'єкту. Нова розробка може бути використана, як комплекс моделей і програмних модулів для оцінки ефективності мультипроцесорної комп'ютерної системи при управління типовими об'єктами. В подальшому

програмний продукт може бути використаний на об'єктах, які використовують в своїй діяльності мультипроцесорні системи з розподіленою пам'яттю. Засіб дозволяє розробнику витратити набагато менше ресурсів пристроїв та ресурсів часу на проектування мультипроцесорної системи з розподіленою пам'яттю. У результаті роботи отримано дослідний зразок – прототип у стадії оптимізації.

На підставі вищевикладеного можна стверджувати, що нове технічне рішення, що пропонується для розробки, буде мати кращі показники ніж у аналога та більшою мірою задовольнить потреби споживачів. Тому його розробка та впровадження є актуальним та доцільним.

Комерціалізація розробки знаходиться на початковому етапі. Ведуться пошуки інвесторів та партнерів. Наявні зацікавлені особи, що готові першими випробувати програмний засіб в обмін на акт впровадження та подальшу рекламу від їх імені. Обсяги можливого продажу 250 шт. протягом першого року реалізації продукту.

Просування на ринок планується шляхом реалізації власного сайту та продажу через спеціалізовані магазини ПЗ. Організація сервісного післяпродажного обслуговування передбачає звернення до розробника безкоштовно. Під час встановлення ціни та попиту на новий програмний продукт основна увага повинна акцентуватися на унікальності об'єкта купівлі-продажу, цінах продуктів конкурентів, перевагах порівняно з аналогами, витратах, які зазнає покупець у разі заміни старого продукту новим, ступені терміновості та гостроті потреби. Орієнтовна ціна нової розробки складатиме 500 грн.

4.2 Прогнозування витрат на виконання науково-дослідної, дослідно-конструкторської та конструкторсько-технологічної роботи

Прогнозування витрат на виконання науково-дослідної роботи складається з таких етапів:

1. Розрахунок витрат, які безпосередньо стосуються виконавців даного розділу роботи;
2. Розрахунок загальних витрат на виконання даної роботи;
3. Прогнозування загальних витрат на виконання та впровадження результатів даної роботи.

Виконаємо розрахунок витрат, які безпосередньо стосуються виконавця даного розділу роботи, приймаючи до уваги те, що розробкою займався один розробник.

При цьому основна заробітна плата для розробника розраховується за формулою:

$$Z_o = \frac{M}{T_p} \cdot t, [\text{грн}], \quad (4.1)$$

де M - місячний посадовий оклад розробника, $M = 12500$ грн.;

T_p - кількість робочих днів у місяці, $T_p = 21$ день;

t - число днів роботи розробника, $t = 66$ дні.

$$Z_o = \frac{12300,00}{21} \cdot 66 = 39285,71 \text{ (грн)}.$$

Додаткова заробітна плата розробника, що брав участь у виконанні даного етапу роботи, розраховується як 10 % від суми основної заробітної плати розробника за формулою:

$$Z_d = 0,10 \cdot Z_o [\text{грн}]. \quad (4.2)$$

$$З_д = 0,10 \cdot 39285,71 = 3928,57 \text{ (грн)}.$$

Нарахування (ЕСВ) на заробітну плату ($H_{зп}$) розробника, що брав участь у виконанні даного етапу роботи становлять 22%. Нарахування на заробітну плату розробника розраховуємо за формулою:

$$H_{zn} = (З_о + З_д) \cdot \frac{\beta}{100} \text{ [грн]}, \quad (4.3)$$

де $З_о$ – основна заробітна плата розробників, грн;
 $З_д$ – додаткова заробітна плата всіх розробників та робітників, грн;
 β – ставка єдиного соціального внеску на загальнообов'язкове державне соціальне страхування, %.

$$H_{зп} = (39285,71 + 3928,57) \cdot 0,22 = 95071,40 \text{ (грн)}.$$

Амортизація обладнання та приміщення, яке використовувалось для проведення розробки, розраховується за формулою:

$$A = \frac{Ц \cdot H_a}{100} \cdot \frac{T}{12} \text{ [грн]}, \quad (4.4)$$

де $Ц$ – балансова вартість обладнання, грн.;
 H_a – річна норма амортизаційних відрахувань;
 T – термін використання під час розробки, місяців.

Норма амортизації розраховується за формулою:

$$H_a = \frac{B_n - B_n}{B_n \cdot T_{кс}} \cdot 100 \text{ [грн]}, \quad (4.5)$$

де $V_{п}$ і $V_{л}$ – відповідно первісна та ліквідаційна вартість основних фондів;

$T_{кв}$ – строк корисного використання, 5 роки.

$$H_a = \frac{20000 - 2000}{20000 \cdot 5} \cdot 100 = 18\% \text{ (грн).}$$

Розрахуємо амортизаційні витрати на ноутбук, балансова вартість якого становить 20000 грн, а термін використання – 3 місяці:

$$A = \frac{20000 \cdot 18}{100} \cdot \frac{3}{12} = 900 \text{ (грн).}$$

Зроблені розрахунки наведено у таблиці 5.5.

Таблиця 4.5 – Амортизаційні відрахування

Найменування	Балансова вартість, грн	Термін використання, р	Фактична трив. використання, міс.	Величина амортизаційних відрахувань, грн
Приміщення	380000,00	20	3	4275,00
Ноутбук	20000,00	5	3	900,00
Всього:				5175,00

Всі види витрачених матеріалів їх вартість і кількість зображені в таблиці 4.6.

Таблиця 4.6 – Витрати на матеріали, що були використані для розробки.

Найменування матеріалу	Одиниці виміру	Ціна за одиницю, грн	Витрачено, шт	Вартість витрачених матеріалів, грн
Папір	уп.	90	1	82,00
Ручка	шт.	10	2	16,00
Олівець	шт.	13	1	8,00
Загальна сума витрат за статтею				106,00

Наступним кроком буде розрахунок витрат на силову електроенергію

(V_e) для виконання даного етапу роботи, розрахунок здійснюється за формулою:

$$V_e = V \cdot П \cdot \Phi \cdot K_{\Pi}, \text{ [грн]}, \quad (4.6)$$

де V – вартість 1 кВт-год. електроенергії, становить 1,9 грн./кВт.

$П$ – установлена потужність комп'ютера, кВт (у нашому випадку становить 0,3 кВт);

Φ – фактична кількість годин роботи обладнання.

K_{Π} – коефіцієнт використання потужності, становить 0,3.

$$V_e = 1,9 \cdot 0,3 \cdot 528 \cdot 0,3 = 90,29 \text{ (грн)}.$$

Інші витрати $V_{\text{ін}}$ охоплюють: витрати на управління організацією, оплату службових відряджень, витрати на утримання, ремонт та експлуатацію основних засобів, витрати на опалення, освітлення, водопостачання, Інтернет послуги. Інші витрати I_B можна прийняти як 100% від суми основної заробітної плати розробника:

$$V_{\text{ін}} = 50\% \cdot (Z_p) \text{ [грн]}. \quad (4.7)$$

$$V_{\text{ін}} = 1 \cdot 39285,71 = 19642,86 \text{ (грн)}.$$

Сума всіх попередніх статей витрат дає витрати на виконання даної частини (розділу, етапу) роботи – V :

$$V = 39285,71 + 3928,57 + 95071,40 \\ + 5175 + 106,00 + 90,29 + 19142,86 = 77235,83 \text{ (грн)}.$$

Наступний етап роботи полягає у розрахунку загальних витрат на виконання даної роботи $V_{\text{заг}}$ та визначається за формулою:

$$B_{\text{заг}} = \frac{B}{\alpha} \text{ [грн]},$$

(4.8)

де α – частка витрат, які безпосередньо здійснює виконавець даного етапу роботи, у відн. одиницях ($\alpha = 1$.)

$$B_{\text{заг}} = \frac{77235,83}{1} = 77235,83 \text{ (грн).}$$

Далі визначаємо загальні витрати на виконання та впровадження результатів виконаної наукової роботи (ЗВ) за формулою:

$$ЗВ = \frac{B_{\text{заг}}}{\beta} \text{ [грн]}, \quad (4.9)$$

де β – коефіцієнт, який характеризує етап (стадію) виконання даної роботи. Так, якщо розробка знаходиться:

- на стадії науково-дослідних робіт, то $\beta \approx 0,1$;
- на стадії технічного проектування, то $\beta \approx 0,2$;
- на стадії розробки конструкторської документації, то $\beta \approx 0,3$;
- на стадії розробки технологій, то $\beta \approx 0,4$;
- на стадії розробки дослідного зразка, то $\beta \approx 0,5$;
- на стадії розробки промислового зразка, $\beta \approx 0,7$;
- на стадії впровадження, то $\beta \approx 0,9$.

$B_{\text{заг}}$ - загальна вартість всієї наукової роботи.

$$ЗВ = \frac{77235,83}{0,5} = 154471,66 \text{ (грн).}$$

Витрати на виконання наукової роботи та впровадження її результатів становитиме 154471,66 грн.

4.3 Прогнозування комерційних ефектів від реалізації результатів розробки

У даному підрозділі проведемо кількісне прогнозування, яку вигоду, зиск можна отримати у майбутньому від впровадження результатів виконаної наукової роботи. В умовах ринку узагальнюючим позитивним результатом, що його отримує підприємство від впровадження результатів тієї чи іншої розробки, є збільшення чистого прибутку підприємства. Зростання чистого прибутку можна оцінити у теперішній вартості грошей. Зростання чистого прибутку забезпечить підприємству надходження додаткових коштів, які дозволять покращити фінансові результати діяльності.

Виконання даної наукової роботи та впровадження її результатів складає приблизно 1 рік. Позитивні результати від впровадження розробки очікуються вже в перші місяці після впровадження.

Проведемо детальніше прогнозування позитивних результатів та кількісне їх оцінювання по роках.

Обчислимо збільшення чистого прибутку підприємства $\Delta\Pi_i$ для кожного із років, протягом яких очікується отримання позитивних результатів від впровадження розробки, розраховується за формулою:

$$\Delta\Pi_i = \sum_1^n (\Delta\Pi_{\text{я}} \cdot N + \Pi_{\text{я}} \cdot \Delta N)_i [\text{грн}], \quad (4.10)$$

де $\Delta\Pi_{\text{я}}$ – покращення основного якісного показника від впровадження результатів розробки у даному році;

N – основний кількісний показник, який визначає діяльність підприємства у даному році до впровадження результатів наукової розробки;

ΔN – покращення основного кількісного показника діяльності підприємства від впровадження результатів розробки;

$\Pi_{\text{я}}$ – основний якісний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки;

n – кількість років, протягом яких очікується отримання позитивних результатів від впровадження розробки.

Припустимо, що внаслідок впровадження результатів наукової розробки чистий прибуток підприємства збільшиться на 125,00 грн, а кількість одиниць реалізованої послуги збільшиться: протягом першого року – на 250 од., протягом другого року – ще на 1500 од., протягом третього року – ще на 1100 од.

Орієнтовно: реалізація продукції до впровадження результатів наукової розробки складала 1 шт., а прибуток, що його отримувало підприємство на одиницю продукції до впровадження результатів наукової розробки – 50,00 грн.

Потрібно спрогнозувати збільшення чистого прибутку підприємства від впровадження результатів наукової розробки у кожному році відносно базового.

Збільшення чистого прибутку підприємства $\Delta\Pi_1$ протягом першого року складе:

$$\Delta\Pi_1 = 50 \cdot 1 + (50 + 125) \cdot 250 = 43800,00 \text{ (грн)}.$$

Обчислимо збільшення чистого прибутку підприємства $\Delta\Pi_2$ протягом другого року:

$$\Delta\Pi_2 = 50 \cdot 1 + (50 + 125) \cdot (250 + 1500) = 306300,00 \text{ (грн)}.$$

Збільшення чистого прибутку підприємства $\Delta\Pi_3$ протягом третього року становитиме:

$$\Delta\Pi_3 = 50 \cdot 1 + (50 + 125) \cdot (250 + 1500 + 1100) = 498800,00 \text{ (грн)}.$$

Отже, розрахунки показують, що комерційний ефект від впровадження розробки виражається у значному збільшенні чистого прибутку підприємства.

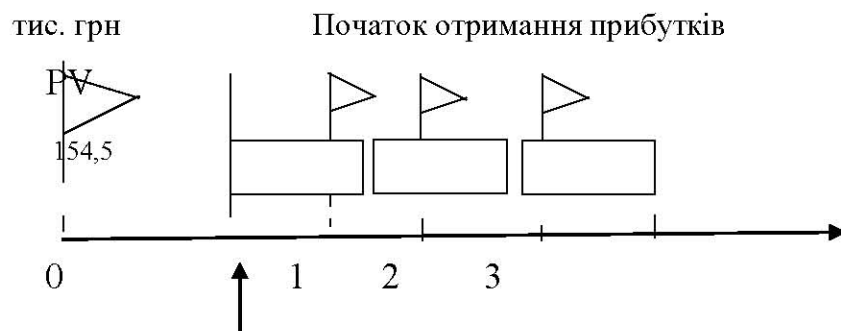
4.4 Розрахунок ефективності вкладених інвестицій та період їх окупності

Для того щоб впровадити розробку необхідно залучити інвестора, а також підтвердити, що дана розробка буде ефективною. Розрахунок ефективності вкладених інвестицій передбачає проведення таких робіт:

1. Розраховуємо теперішню вартість інвестицій PV, що вкладаються в наукову розробку. Будемо вважати, загальні витрати на виконання та впровадження результатів НДДКР (або теперішня вартість інвестицій PV) дорівнює 154471,66 грн.

2. Очікуване збільшення прибутку $\Delta\Pi$, що його отримає підприємство від впровадження результатів наукової розробки, для кожного із років, починаючи з першого року впровадження розраховане раніше. У першому році підприємство отримає збільшення чистого прибутку відносно базового на 43800,00 грн, у другому році – на 306300,00 грн, у третьому році – на 498800,00 грн.

3. На третьому етапі, будемо вісь часу, на яку наносимо всі платежі (інвестиції та прибутки), що мають місце під час виконання НДДКР та впровадження її результатів. Рисунок, що характеризує рух платежів (інвестицій та додаткових прибутків) буде мати вигляд, наведений на рис. 4.1.



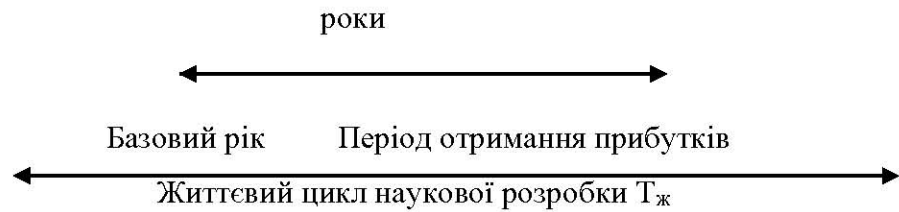


Рисунок 4.1 – Вісь часу з фіксацією платежів, що мають місце під час розробки та впровадження результатів НДДКР

Наступним кроком є розрахунок абсолютної ефективності вкладених інвестицій $E_{абс}$:

$$E_{абс} = (ПП - PV), \text{ [грн]}, \quad (4.11)$$

де ПП – приведена вартість всіх чистих прибутків, що їх отримає підприємство (організація) від реалізації результатів наукової розробки, грн.;

PV – теперішня вартість інвестицій $PV = 3B$, грн.

У свою чергу, приведена вартість всіх чистих прибутків ПП розраховується за формулою:

$$ПП = \sum_1^{\tau} \frac{\Delta\Pi_i}{(1+\tau)^t}, \text{ [грн]}, \quad (4.12)$$

де $\Delta\Pi_i$ – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої НДДКР, грн;

τ – період часу, протягом якого виявляються результати впровадженої НДДКР, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні; для України цей показник знаходиться на рівні 0,1;

t – період часу (в роках) від моменту отримання чистого прибутку до точки 0.

$$\text{ПП} = \frac{43800,00}{(1 + 0,1)^1} + \frac{306300,00}{(1 + 0,1)^2} + \frac{498800,00}{(1 + 0,1)^3} = 667962,43(\text{грн}).$$

Після цього розраховуємо абсолютної ефективності вкладених інвестицій $E_{\text{абс}}$:

$$E_{\text{абс}} = 667962,43 - 154471,66 = 513490,77 (\text{грн.})$$

Оскільки $E_{\text{абс}} > 0$, то результат від проведення наукових досліджень та їх впровадження принесе прибуток, а вкладання коштів у даний проект є доцільним.

Розраховуємо відносну (щорічну) ефективність вкладених в наукову розробку інвестицій $E_{\text{в}}$ за формулою:

$$E_{\text{в}} = \sqrt[T_{\text{ж}}]{1 + \frac{E_{\text{абс}}}{PV}} - 1 \quad (4.13)$$

де $E_{\text{абс}}$ – абсолютна ефективність вкладених інвестицій, грн;

PV – теперішня вартість інвестицій $PV = 3B$, грн;

$T_{\text{ж}}$ – життєвий цикл наукової розробки, роки.

Отже, відносна (щорічна) ефективність вкладених в наукову розробку інвестицій $E_{\text{в}}$ складе:

$$E_{\text{в}} = \sqrt[3]{1 + \frac{513490,77}{154471,66}} - 1 = \sqrt[3]{4,32} - 1 = (1,63 - 1) = 0,63 \text{ або } 63\%$$

Далі, розраховану величину $E_{\text{в}}$ порівнюємо з мінімальною (бар'єрною) ставкою дисконтування $\tau_{\text{мін}}$, яка визначає ту мінімальну дохідність, нижче за яку інвестиції вкладатися не будуть. У загальному вигляді мінімальна (бар'єрна) ставка дисконтування $\tau_{\text{мін}}$ визначається за формулою:

$$\tau = d + f, \quad (4.14)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; $d = 0,18$;

f – показник, що характеризує ризикованість вкладень; зазвичай, величина $f = (0,05 \dots 0,1)$, але може бути і значно більше, у нашому випадку $f = 0,5$.

$$\tau = 0,18 + 0,05 = 0,23$$

Оскільки $E_B = 63\% > \tau_{\text{мін}} = 23\%$, то інвестор буде зацікавлений вкладати гроші в дану наукову розробку, адже він отримає більший прибуток, ніж просто покладе свої гроші на депозит у комерційному банку.

Наступним кроком є визначення терміну окупності вкладених у реалізацію наукового проекту інвестицій. Термін окупності вкладених у реалізацію наукового проекту інвестицій $T_{\text{ок}}$ розраховується за формулою:

$$T_{\text{ок}} = \frac{1}{E_e} [\text{року}] \quad (4.15)$$

Отже, термін окупності складе:

$$T_{\text{ок}} = \frac{1}{0,63} = 1,59 \text{ (року)},$$

Термін окупності інвестицій $T_{\text{ок}} 1,59 < 3 \dots 5$ років і свідчить, що фінансування даної наукової розробки є доцільним.

Отже, розрахунок ефективності вкладених інвестицій та періоду їх окупності показав, що фінансування розробки є доцільним, оскільки інвестиції буде повернуто в термін до двох років.

4.5 Висновок

В даному розділі було виконано оцінювання комерційного потенціалу нової розробки.

Проведено технологічний аудит з залученням трьох незалежних експертів. Визначено, що рівень комерційного потенціалу розробки є високим.

Аналіз комерційного потенціалу розробки показав, що програмний продукт за своїми характеристиками випереджає аналогічні програмні продукти, що підтверджує її перспективність. Він має кращі функціональні показники, а тому є конкурентоспроможним товаром на ринку. Існуючі переваги нової розробки сприятимуть швидкому її поширенню на ринку.

Згідно із розрахунками всіх статей витрат на виконання науково-дослідної, дослідно-конструкторської та конструкторсько-технологічної роботи загальні витрати на розробку складають 154471,66 грн.

Розрахована абсолютна ефективність вкладених інвестицій в сумі 513490,77 грн свідчить про отримання прибутку інвестором від комерціалізації програмного продукту.

Щорічна ефективність вкладених в наукову розробку інвестицій складає 63 %, що вище за мінімальну бар'єрну ставку дисконтування, яка складає 23%. Це означає потенційну зацікавленість інвесторів у фінансуванні розробки.

Термін окупності вкладених у реалізацію проекту інвестицій становить 1,59 року, що також свідчить про доцільність фінансування нової розробки.

ВИСНОВКИ

В роботі досягнуто поставленої мети – проаналізовано і вибрано структуру і математичні моделі для аналізу ефективності інтегрованих комплексів «об’єкт управління, комп’ютерна система» і побудовано комплекс засобів для аналізу ефективності мультипроцесорних систем. В аналогах ці проблеми розглядаються на рівні статистики, емпіричних формул і просто словесних міркувань.

Виконано такі напрямки досліджень:

- проаналізовано структуру і функції засобів для тестування і оцінки ефективності мультипроцесорних систем з розподіленою пам’яттю;
- розроблено моделі оцінювання ефективності комп’ютерних систем з урахуванням розвитку та інновацій;
- розроблено моделі для тестування мультипроцесорної системи управління виробничою системою;
- перевірено ефективність та коректність програмного забезпечення.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Авторське свідоцтво на програму «Оптимальне управління виробництвом на базі оптимального агрегування» авторів Боровська Т.М., Волосович О.Ю., Колесник І.С., Недоснований О.Ю.»
2. Колесник І.С., Закордонець А.В. Функції імітаційного моделювання в роботі сучасних комп'ютерних систем / Колесник І.С., Закордонець А.В. // Тези доповідей НТКП ВНТУ. Факультет інформаційних технологій та комп'ютерної інженерії (2018), Україна, Вінниця, 22-23 березня 2018 р.: збірник наукових праць.- Вінниця: ВНТУ, 2018.
3. Отримання ймовірностей розподілу ринку між виробниками певного сегменту виробництва / Боровська Т. М., Колесник І.С., Закордонець А.В., Григоришен О.О. / ЗАЯВА №4 від 04.11.2019
4. Методичні вказівки до виконання студентами-магістрантами економічної частини магістерських кваліфікаційних робіт / Уклад. В. О. Козловський – Вінниця: ВНТУ, 2012. – 22 с.
5. Колесник І. С. Моделювання процесів розподілу ресурсів у децентралізованих системах: дис. 01.05.02 канд. техн. наук, зах. 17.03.2006, затв. 08.06.2006 / Колесник І. С. – Вінниця.: ВНТУ, 2016. – 208 с.
6. Колесник І.С. Розробка імітаційних моделей для оцінки ризиків ринку./ Колесник І.С., Северілов П.В. ,Северілов В.А. // Матеріали V Міжнародної НПК „Економічна безпека сучасного підприємства”. – Вінниця: УНІВЕРСУМ-Вінниця, 2014 – С. 66-71 – ISBN 978-966-641-246-4.
7. мережі як напрям розвитку розподілених супер-ЕОМ//Сб. "Системи управління і силова електроніка"/ Під заг. ред. В.Д.Ковальова.-М.: ВЕІ.-2001, с.62-67
8. Дзегеленок І.І., Ільїн П.Е. і ін. Декомпозиційний підхід до здійснення GRID-технологій// Науково-технічний журнал «Інформаційна математика»– М.: Вид-во «АСТ-фізико-математична література» № 1(5), 2005, с.110–119
Таненбаум Е. Комп'ютерні мережі. – СПб.: Пітер, 2004 р. 848с.

9. Щеглов А. Ю. Принципи обслуговування заявок в обчислювальних системах з динамічними відносними пріоритетами // Інформаційні технології. 1997. № 8. С. 17—21.
10. Щеглов А. Ю. Принципи уніфікації методів кодового управління множинним доступом до ресурсів обчислювальних систем і ЛОМ // Інформаційні технології. 1998. № 2. С. 20-25.
11. Щеглов А. Ю. Метод синтезу розкладів обслуговування заявок для розподілених обчислювальних систем і ЛОМ реального часу // Інформаційні технології. 1997. № 10. С. 22-27.
12. Шеннон Р. Имитационное моделирование систем – искусство и наука. / Шеннон Р. – М.: Мир, 1978. – 418 с.
13. Шрейдер Ю.А. Системы и модели. / Шрейдер Ю.А., Шаров А.А. — М.: Радио и связь, 1982. – 216 с.
14. Ясинский Ф.Н. Математическое моделирование с помощью компьютерных сетей. / Ясинский Ф.Н., Чернышева Л.П., Пекунов В.В. – Иваново, 2000. – 204 с.
15. Bellman R.E. Decision-making in a Fuzzy Environment. / Bellman R.E., Zadeh L.A. — MS, vol. 17, 1970.
16. David L. Drake, Katherine L. Morse. The Security-Specific Eight-Stage Risk Assessments Methodology. - Proceed. Of the 17 th National Computer Security Conference, 1946.
17. Nykolaychuk Y., Pitukh I., Vozna N., Nykolaychuk L.. Information technologies of models formalization and designing for data movement in computer networks of automatic control system // Proc. of the third IEEE workshop on Intelligent data acquisition and advanced computing systems: Technology and applications (IDAACS'2005). - Sofia, (Bulgaria).-2005. - P.253-259.