

С. С. Бульба, В. В. Давидов, Г. А. Кучук

Національний технічний університет "Харківський політехнічний інститут", Харків, Україна

## МЕТОД РОЗПОДІЛУ РЕСУРСІВ МІЖ КОМПОЗИТНИМИ ЗАСТОСУНКАМИ

**Предмет** розгляду – методи оптимального розподілу ресурсів. **Метою статті** є розробка методу розподілу ресурсів між композитними застосунками, орієнтованому на динамічний вибір алгоритму оптимізації. методи, що використовуються, – жадібні та мурашині алгоритми, кластеризаційний підхід. **Результати роботи.** Розглянуто узагальнення існуючих евристик динамічного планування, які відносяться до класу жадібних алгоритмів та знаходять на кожному кроці локально оптимальне рішення. Проведено укрупнену класифікацію методів розподілу пакету композитних застосунків. Проаналізовано два типи кластеризації - вертикальну та горизонтальну. Розглянута можливість планування наборів композитних застосунків на базі кластеризаційного підходу. Наведено спосіб розподілу наборів композитних застосунків як з використанням мурашиного алгоритму, так і на базі кластеризаційного підходу з використанням мурашиного алгоритму. Проаналізовано переваги та недоліки кожного із розглянутих підходів, що дало змогу визначити межі застосування кожного із підходів. **Висновки.** Запропоновано чотири різних підходи до розподілу ресурсів між композитними застосунками з використанням таких методів: жадібні алгоритми, мурашині алгоритми, вертикальна та горизонтальна кластеризація, сумісне використання мурашиного алгоритму та кластеризації. Розроблений метод дозволяє динамічно провести найкращий вибір. Подальші дослідження будуть направлені на розробку відповідного алгоритму.

**Ключові слова:** композитний застосунок, жадібний алгоритм, мурашиний алгоритм, кластеризація.

### Вступ

Безперервно зростаючі вимоги в області складних обчислювальних систем призводять до необхідності розвитку нових методів реалізації обчислювальних ресурсів і сервісів, функціонування яких можуть задовольнити дані вимоги. Один з напрямків вирішення даної проблеми засноване на подальшому розвитку технологій розробки платформ розподілених обчислень для виконання композитних застосунків (КЗ). Під композитними застосунками в розподілених середовищах маються на увазі програми, які дозволяють користувачеві мати доступ до сервісів, розташованих на різних пристроях, не пов'язаних між собою. Даний підхід дає можливість в деякій мірі забезпечити процес обробки інформації завдяки тому, що вихід з робочого стану одного з сервісів не призводить до зриву процесу.

У процесі роботи в даному обчислювальному середовищі користувач надсилає запит на виконання необхідних обчислень, після чого дана система розподіляє отримане навантаження між наявними сервісами. Далі сервіси повертають оброблену інформацію назад, компонується відповідь і надається користувачеві. При цьому при паралельному виконанні декількох КЗ виникає задача розподілу ресурсів системи між КЗ.

**Аналіз літератури.** В [1 – 7] розглядаються різні підходи для розподілу ресурсів у інфокомунікаційних системах. При цьому застосовується цілий ряд оптимізаційних алгоритмів, зокрема, кожен із підходів, що пропонується, зазвичай є орієнтованим тільки на один із методів. Однак варто приділити увагу питанням, пов'язаним з динамічним вибором оптимізаційного алгоритму, при цьому вибір повинен бути орієнтовним на якість розподілу. Це і визначило завдання досліджень, тобто метою даної статті є розробка методу розподілу ресурсів між композитними застосунками, орієнтованому на динамічний вибір алгоритму оптимізації.

### 1. Жадібні алгоритми

Жадібний алгоритм — евристичний алгоритм, який приймає найкраще рішення, виходячи з наявних на поточному етапі даних [8]. Сутність жадібного алгоритму полягає локально оптимальному виборі на кожному кроці. Для реалізації алгоритму необхідно визначити такі елементи задачі:

- набір можливих варіантів, з яких робиться вибір;
- функція вибору, за допомогою якої знаходиться найкращий варіант;
- функція придатності, яка визначає придатність отриманого набору;
- функція цілі, яка оцінює цінність рішення та зазвичай не виражена явно;
- функція розв'язку, яка вказує на те, що знайдене кінцеве рішення.

Основний недолік жадібних алгоритмів – велика ймовірність того, що знайдені рішення не буде оптимальним. Основна перевага – швидкість знаходження рішення. Якщо постановка задачі потребує знаходження оптимального, а не раціонального рішення, то необхідно проведення аналізу можливостей його застосування.

До оптимізаційної задачі можна застосувати принцип жадібного вибору, якщо послідовність локально оптимальних виборів дає глобальний оптимальний розв'язок. Доведення оптимальності можна проводити за такою схемою: довести, що жадібний вибір на першому етапі не унеможливило шляху до оптимального розв'язку: для всякого розв'язку є інше, узгоджене із жадібним і не гірше першого, подалі довести, що підзадача, яка виникла після жадібного вибору на першому етапі, аналогічна початковій. Доведення проводиться методом математичної індукції [6].

Ідею, що лежить у використанні жадібних алгоритмів до розподілу ресурсів КЗ, можна описати таким чином:

- на кожному етапі розподілу ресурсів для КЗ розглядається деякий набір обчислювальних задач, які можуть належати різним КЗ;
- у існуючому наборі можуть міститися тільки задачі, виконання яких можливо на даний момент виходячи з відносин черговості графів КЗ;
- відповідно до значення деякого критерію вибору задач КВЗ вибирається одна ОЗ з набору КЗ, після чого вибрана ОЗ отримує ОБ відповідно до другого критерію призначення ресурсу КПП, приклад представлено на рис. 1;

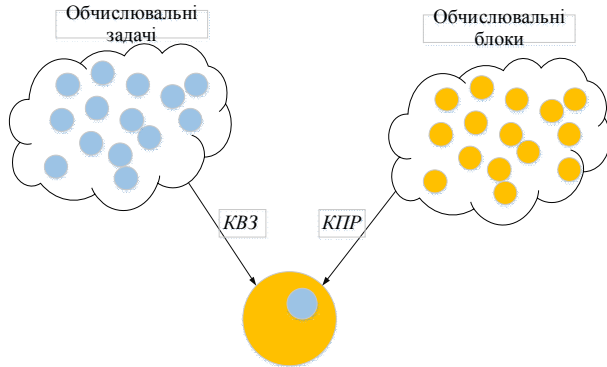


Рис. 1. Планування виконання КЗ з урахуванням критеріїв КВЗ та КПП

- поповнення множини КЗ новими ОЗ здійснюється за існуючими моделями.

Підхід, що розглядається, є узагальненням існуючих евристик динамічного планування, які відносяться до класу жадібних алгоритмів та знаходять на кожному кроці локально оптимальне рішення. У даному випадку прийняття рішення про призначення для всіх ОЗ здійснюється до їх запуску, тобто формується статичний план.

Для коректного розподілу ресурсів між ОЗ в представленому методі виникає необхідність в розподілі на базі критеріїв вибору завдання *MinEnd*, або *MaxEnd*. (табл. 1). Такий підхід дасть можливість враховувати витрати на передачу даних і можливість наявності попередньо зарезервованих часових вікон і вікон, зайнятих вже спланованими завданнями. Критерій вибору ресурсу дозволяє визначити ОБ з множини  $CP_{mn}$  для ОЗ що буде на нього назначена. Час завершення завдання на ресурсах обраного типу розраховується з урахуванням поточного завантаження ресурсів їх утилізації та часу комунікації.

Задамо множину ОЗ *StPar* призначених для розподілу. Множина *StPar* формуються з ОЗ які не мають батьківських обчислювальних задач або вони вже виконані.

Таблиця 1 – Опис критеріїв

Позначення	Вираз для розрахунку	Опис критерію
<i>КВЗ (критерій вибору завдань з набору)</i>		
MinTime,	$\min_{(m,n)} \tau_{mn}$	Планування за мінімальним часом виконання ОЗ
MaxTime	$\max_{(m,n)} \tau_{mn}$	Планування за максимальним часом виконання ОЗ
MinAvgTime	$\min_{(m,n)} \frac{\tau_{mn}}{card(CP_{smn})}$	Планування за мінімальним середнім часом виконання ОЗ
MaxAvgTime	$\max_{(m,n)} \frac{\tau_{mn}}{card(CP_{smn})}$	Планування за максимальним середнім часом виконання ОЗ
MinEnd,	$\min_{(m,n)} t_{mn}^{(1)}$	Планування за мінімальним часом завершення ОЗ
MaxEnd	$\max_{(m,n)} t_{mn}^{(1)}$	Планування за максимальним часом завершення ОЗ
MinAvgEnd,	$\min_{(m,n)} \sum_1^{card(CP_{smn})} t_{mn}^{(end)} / card(CP_{smn})$	Планування за мінімальним середнім часом завершення ОЗ
MaxAvgEnd	$\max_{(m,n)} \sum_1^{card(CP_{smn})} t_{mn}^{(end)} / card(CP_{smn})$	Планування за максимальним середнім часом завершення ОЗ
Suffrage	$\max_{(m,n)} (weig_m \min t_{mn}^{(1)} - \min t_{mn}^{(1)})$	Планування за максимальною різницею значення часу завершення ОЗ
<i>КПП (критерій призначення ресурсу)</i>		
MinTime	ОЗ призначаються на ОБ який за мінімальний час виконає її розрахунок	
MinEnd	ОЗ призначаються на ОБ який за мінімальний час завершить її розрахунок	
Rand	ОБ призначається випадково	
MinLoad	Завдання призначається на ОБ з мінімальним навантаженням	
MinUtil	Завдання призначається на ОБ з мінімальною утилізацією	

Для поповнення множини *StPar* існують такі методи:

- *PlanT*. Нова ОЗ додається до множини *StPar* лише у тому випадку коли виконуються умови: а) ОЗ не розподілена та не міститься в наборі; б) для неї розподілені всі батьківські ОЗ, які завершуються перед її виконанням .

- *ExecutT*. Проводиться розподіл ОЗ з множини *StPar*, далі множина *StPar* формується шляхом додавання ОЗ, у яких всі батьківські ОЗ були виконані.

Представлені критерії визначають групу методів розподілу *GreedPlan* яка складеться з *КВЗ*, *КПП*, *PlanT* *ExecutT*, отже розроблені методи розподілу

мають наступний вигляд *GreedPlan(KB3, KPP, PlanTExecutT)*. Представлений план може мати застосункові критерії при розподілі обчислювальних задач КЗ.

Отже різні методики розподілу в залежності від конфігурації ОР, числа КЗ і віддаленості крайніх термінів завершення будуть давати результати які відрізняються як з точки зору дотримання термінів виконання, так і з точки зору ефективності утилізації, а отже доцільно вибирати в якості еталонного план з найкращим значенням утилізації ресурсів.

## 2. Кластеризація завдань наборів композитних застосунків

Виконання розподілу пакету композитних застосунків можливо за допомогою таких методів [10]:

- розподіл по обчислювальним задачам, тобто КЗ представлені до розподілу об'єднуються, потім за допомогою обраного алгоритму розподілу формується метод розподілу для одного КЗ;
  - розподіл на основі пріоритетизації КЗ - композитні застосунки з пакету  $DS$  розподіляються відповідно до деякого порядку в залежності від їх пріоритету за обраним параметром;
  - розподіл на основі кластеризації ОЗ - множина обчислювальних задач  $NDS_{mn}$  розбивається на підмножини, які потім розподіляються на ОР в залежності від вимог розподілу.
- Існують такі типи кластеризації - вертикальна та горизонтальна.

При вертикальній кластеризації, кожний КЗ для розподілу на ОБ розбивається на групи ОЗ з врахуванням закінчення попередніх задач, рис. 2.

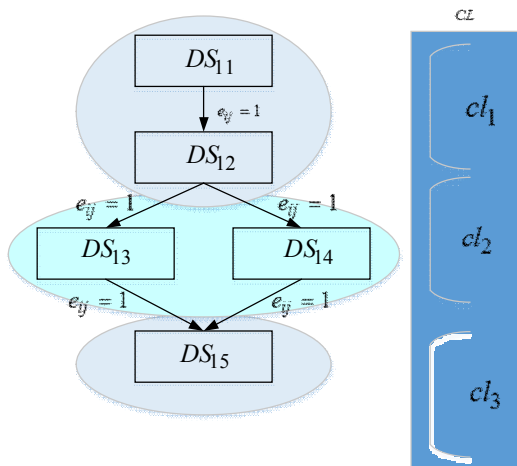


Рис. 2. Вертикальна кластеризація

При горизонтальній кластеризації виникає можливість входження завдань різних КЗ в один і той же кластер, приклад наведено на рис. 3. При горизонтальній кластеризації також є необхідність врахування закінчення попередніх задач. Процес розподілу при використанні представлених методів кластеризації ділиться на етапи, причому кожному етапу назначається набір завдань що підлягають розподілу. Додавання ОЗ до набору здійснюється тільки із застосуванням методу *ExecutT*

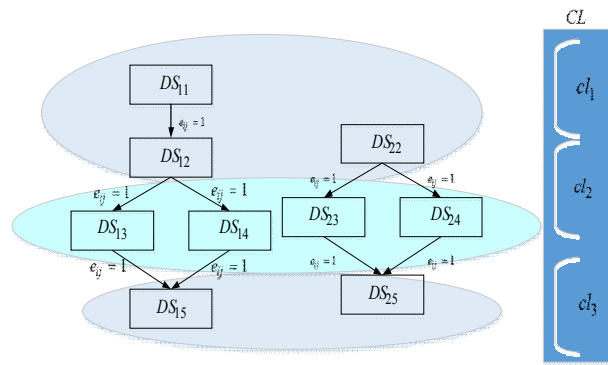


Рис. 3. Горизонтальна кластеризація

Етап розподілу на основі кластеризації полягає в розміщенні підмножини задач одного або декількох КП, відповідно число етапів залежить від способу групування завдань.

Так як вертикальна кластеризація може вважатися окремим випадком горизонтальної, формально спільне завдання кластеризації може бути описана розбиттям множини завдань  $NDS_{mn} = \bigcup_{m=1}^{N_{DS}} \bigcup_{n=1}^{n_m} DS_{mn}$

що належать пакету КЗ  $DS$  на підмножину

$$CL = \bigcup_{k=1}^{Ncl} cl_k \quad (1)$$

де  $Ncl$  - кількість кластерів отриманих при розбитті множини  $NDS_{mn}$ ,  $cl_k$  - кластер з номером  $k$ .

З дотриманням умов що кожна задача  $DS_{mn}$  належить певному кластеру  $cl_k$ , тобто  $\forall DS_{mn} \in NDS_{mn} \exists cl_k \in CL$ , та кожна задача входить лише до одного кластеру  $cl_k$ .

Крім того, при розподілі множини завдань КЗ  $NDS_{mn}$  на кластери  $cl_k$  необхідно враховувати що структура композитних застосунків задана ациклічним графом. Звідси виникає необхідність в дотриманні послідовності виконання ОЗ в залежності від заданої в структурі КЗ. Тобто кожне наступне завдання може виконуватися лише після завершення його попередника.

**3.2.2 Планування наборів композитних застосунків на базі кластеризаційного підходу.** Під час розподілу пакету КЗ  $DS$  методом кластеризації виникає необхідність в знаходженні способу формування кластерів  $cl_k$  з множини існуючих ОЗ  $NDS_{mn}$ . Під час кластеризації створена множина кластерів  $CL$  може характеризуватися такими параметрами як кількість кластерів та параметром кластеру. Параметром кластера може виступати мінімальний або максимальний: час завершення або старту ОЗ, час обчислення ОЗ та інше. Отже під час кластеризації ці параметри можуть бути не задані, попередньо задані, або задаються під час кластеризації.

Якщо під час розподілу ОЗ параметри кластеризації не задані, то виникають такі види кластерів як об'єднання усіх задач КЗ в один кластер, та кластеризація в залежності від КЗ. Перший підхід не дає змогу

навіть мінімально впливати на процес розподілу, тобто ми не маємо змоги впливати на початок чи завершення певного КЗ. Під час другого підходу число кластерів співпадає з кількістю КЗ. Такий розподіл завжди є коректним, а також дає можливість враховувати пріоритетність КЗ. Але такий підхід не дає змогу справедливо розподіляти часові вікна на обчислювальних блоках для виконання пакету КЗ.

Якщо число кластерів  $N_{cl}$  і параметри кластеризації попередньо задані, тоді виникає статична кластеризація з числом кластерів від 1 до  $N_p$ . Після цього ОЗ в випадковому порядку розподіляються по кластерам  $cl_k$  з множини  $CL$ . Під час такої кластеризації необхідно враховувати коректність розподілу після процесу кластеризації.

Якщо число кластерів  $N_{cl}$  задано, а параметри кластеризації задаються дід час кластеризації, то в такому випадку кожна ОЗ призначається певному кластеру  $cl_k$  в залежності від критерію розподілу. Для правильного розподілу ОЗ до кластеру виникає необхідність стежити за коректністю розподілу, а отже розглядати ОЗ від їх попередників або за допомогою методів *PlanT* або *ExecuT*.

Якщо під час розподілу ОЗ параметри кластеризації задаються під час кластеризації, то виникають наступні способи кластеризації:

- кластеризація на базі об'єднання - спочатку кожна ОЗ розподіляється в окремий кластер, після чого задаються параметри об'єднання кластерів і критерій закінчення кластеризації;
- динамічна кластеризація - один етап кластеризації в цьому випадку включає вибір ОЗ, для якої буде визначатися кластер, і призначення ОЗ існуючого кластера або створення нового кластера в залежності від параметру кластера.

Після задання способу кластеризації, виникає необхідність в засобі знаходження коректного порядку розподілу кластерів на ОБ. На даний момент існують засоби які базуються на методах *PlanT* або *ExecuT*. Тобто кластери упорядковуються один раз до початку розподілу на ОБ, або пошук наступного кластера проводиться перед кожним етапом розподілу на ОБ. Таким чином, число і спосіб взаємного розташування трьох основних етапів розподілу з використанням кластеризаційного підходу можуть варіюватися в залежності від конкретного алгоритму.

**3.3.1 Розподіл наборів композитних застосунків на базі мурашиного алгоритму.** Робота мурашиного алгоритму базується на моделюванні здатності мурахи знаходити оптимальний шлях до джерела їжі у дикій природі, а також адаптуватися до зміни умов при пошуку цього шляху [9]. Такі можливості зв'язані з використанням мурахами певного феромону для помітки пройденого шляху. Після того як феромон було нанесено мурахою, інший представник може використовувати цю інформацію для свого пересування, чим більше було нанесено феромону тим привабливіше стає даний шлях. Для підтримки можливості пошуку іншого шляху нанесений феромон з часом вивітрюється якщо його не

поновлювати. Моделювання вивітрювання феромону дає гарантію того що знайдене локально-оптимальне рішення не буде єдиним, тобто мурахи будуть шукати інші шляхи.

Для використання описаного алгоритму проведемо його модифікацію. У нашому випадку мурахами ми будемо вважати існуючі ОЗ  $DS_{mn}$  композитного застосунку  $DS_m$  що прийшов на обчислення, тоді шляхом вважатимемо ОБ певного типу, на якому може обчислюватися представлена ОЗ. Феромоном виступає блок статистики БС який тримає у собі значення критеріїв виконання ОЗ на обчислювальному блоці  $p_{si}$  що обчислює задачу яка надійшла. До критеріїв виконання відносяться: мінімальний або максимальний час виконання ОЗ, та навантаження ОБ, значення утилізації ресурсів, час комунікації між обчислювальними блоками.

Щоб описати роботу методів представленого алгоритму необхідно обрати обчислювальні задачі  $DS_{mn}$  для розподілу їх на множину ресурсів  $P$ , так щоб задача яка підтягаються розподілу не мала попередника. Для цього використаємо метод *ExecuT*, додавання нових задач також будемо проводити за допомогою даного методу. Для початку виконання алгоритму необхідно задати початкові значення евристичних метрик «феромонів», які будуть визначати привабливість *attrac* ОБ для виконання певної типової ОЗ. Це можливо завдяки тому що в мережі Укрзалізниці виконуються типові завдання існуючих АСК.

Графічне зображення наведено на рис. 4.

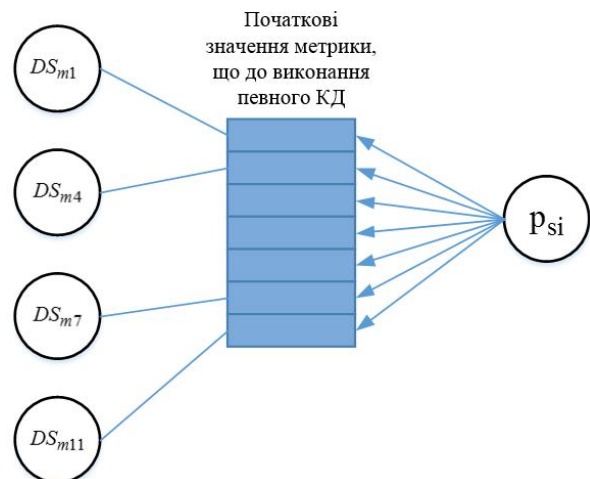


Рис 4. Початкові значення метрик

Якщо розподіл задачі виконуються в перше, тоді необхідно враховувати лише значення початкової привабливості *attrac* обчислювального блоку. Для цього необхідно привабливість певного параметру усіх ОБ з множини обчислювальних блоків  $CP_{mn}$ , за 100%. Тоді вірогідність розподілу ОЗ на певний ресурс буде проводитись випадково, але з урахуванням відсоткового значення привабливості кожного ресурсу. При подальшому розподілі необхідно враховувати як початкові значення привабливості параметрів так і значення статистичної при-



вабливості  $attracSt$  кожного ОБ що знаходяться в блоці статистики.

В залежності від того яке значення ваги у представленому виразі мають параметри  $attrac$  та  $attracSt$ , буде залежати значення зміни відсоткової частки привабливості.

На проміжку певного часу, представлений метод (Жадібної мурахи) дає змогу знайти кращий варіант розподілу ОЗ на обчислювальний блок з певною групою критеріїв. Але виникає проблема з пошуком нових ОБ які будуть виконувати обчислення задач КЗ, так як знайдений блок завжди буде кращим.

У такому випадку необхідно використовувати можливість вивітрювання статистичної привабливості ОБ за деякою функцією  $wind()$ , що буде зменшувати відсоткову привабливість представленого ОБ в загальній множині  $CP_{mn}$ , що здатна виконувати завдання  $DS_{mn}$ .

Представлений підхід дає змогу реалізувати пошук ОБ що буде мати найкращу привабливість для розподілу ОЗ в залежності від вибраного критерію, з урахування можливості зміни множини ресурсів доступних для виконання.

**3.3.2 Розподіл наборів композитних застосунків на базі кластеризаційного підходу з використанням мурашиного алгоритму.** Представлений метод розподілу ресурсів на основі мурашиного алгоритму дає змогу для розширення можливостей кластеризаційного підходу.

У даному випадку за мурахою вважається певний кластер  $cl_k$  з множини  $CL$ , який було сформовано за допомогою певного параметру кластеризації. Початкові значення привабливості  $attrac$  ОБ задаються кожному кластеру  $cl_k$  який приймає участь у розподілі. Під час вибору критерію розподілу кластеру необхідно враховувати параметри за яким було сформовано вибраний кластер, якщо параметри було задано, то і пошук ОБ необхідно виконувати за такими критеріями. Метод розподілу ресурсів на основі мурашиного алгоритму може використовуватися як з вертикальною так і с горизонтальною кластеризацією після їх повного впорядкування. Але для кожного виду кластеризації необхідно використовувати різні початкові параметри привабливості  $attrac$  ОБ.

Наступні кроки виконання мурашиного алгоритму на основі кластеризаційного підходу співпадають з описаними у підрозділі 3.3.1. Але параметри необхідно розуміти що вивітрювання параметрів статистичної привабливості необхідно використовувати в залежності від методу розподілу.

Загальна схема роботи мурашиного алгоритму на основі кластеризаційного підходу представлена на рис 5.

Для того щоб описані методи розподілу ресурсів використовувались з більшою гнучкістю створимо блок розподілу ресурсів (БРР). Створений блок буде нести відповідальність за вибір методу розподілу в залежності від отриманих статистичних даних розподілу пакету КЗ, або в залежності від

пріоритетності виконання КЗ. Тобто, якщо при розподілі та виконанні типового пакету композитних застосунків за певними параметрами кращі результати виконання було досягнуто при виконанні мурашиного алгоритму то в наступному розподілі буде дано пріоритет даному алгоритму.

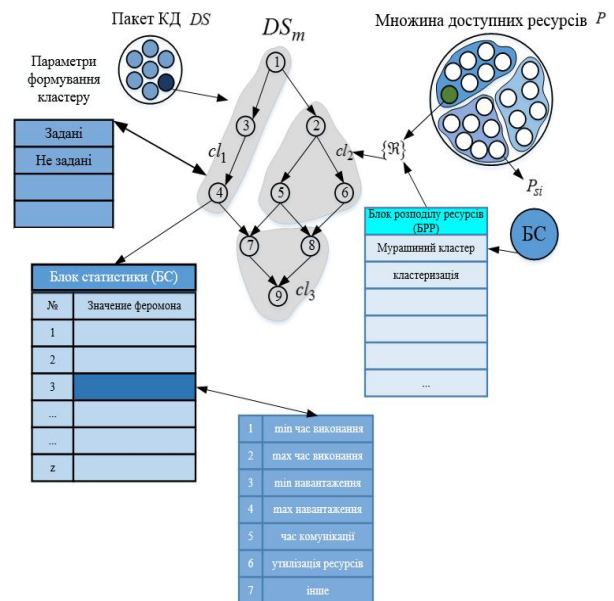


Рис. 5. Загальна схема розподілу за мурашиним алгоритмом на основі кластеризаційного підходу

Але для кращого контролю вибору розподілу пакету КЗ ми можемо також використовувати мурашиний алгоритм що дасть змогу корегувати проведений вибір в залежності від зміни протягом певного часу.

## Висновки

Таким чином, у статті розглянуто узагальнення існуючих евристик динамічного планування, які відносяться до класу жадібних алгоритмів та знаходять на кожному кроці локально оптимальне рішення. Проведено укрупнену класифікацію методів розподілу пакету композитних застосунків. Проаналізовано два типи кластеризації - вертикальну та горизонтальну. Розглянута можливість планування наборів композитних застосунків на базі кластеризаційного підходу. Наведено спосіб розподілу наборів композитних застосунків як з використанням мурашиного алгоритму, так і на базі кластеризаційного підходу з використанням мурашиного алгоритму. Проаналізовано переваги та недоліки кожного із розглянутих підходів, що дало змогу визначити межі застосування кожного із підходів. Запропоновано чотири різних підходи до розподілу ресурсів між композитними застосунками з використанням таких методів: жадібні алгоритми, мурашині алгоритми, вертикальна та горизонтальна кластеризація, сумісне використання мурашиного алгоритму та кластеризації. Розроблений метод дозволяє динамічно провести найкращий вибір.

Подальші дослідження будуть направлені на розробку відповідного алгоритму.

## СПИСОК ЛІТЕРАТУРИ

1. Lord P. Seven Bottlenecks to Workflow Reuse and Repurposing Sattler // P. Lord, C. Goble // The Semantic Web – ISW 2005. – 2005. – P. 323-337.
2. Князьков К. В. Предмет ориентированные технологии разработки приложений в распределенных средах // К. В. Князьков, А. В. Ларченко // Известия вузов. Приборостроение. – 2011. – № 10. – С. 36–43.
3. Merlac V. Resources Distribution Method of University e-learning on the Hyperconvergent platform / V. Merlac, S. Smatkov, N. Kuchuk, A. Nechausov // Conference Proceedings of 2018 IEEE 9th International Conference on Dependable Systems, Service and Technologies. DESSERT'2018. Ukraine, Kyiv, May 24-27, 2018. – P. 136-140.
4. Коваленко А. А. Сучасний стан та тенденції розвитку комп'ютерних систем об'єктів критичного застосування / А. А. Коваленко, Г. А. Кучук // Системи управління, навігації та зв'язку. – Полтава. ПНТУ, 2018. – Вип. 1(47). – С. 110-113.
5. Kuchuk G. Approaches to selection of combinatorial algorithm for optimization in network traffic control of safety-critical systems / G. Kuchuk, V. Kharchenko, A. Kovalenko, E. Ruchkov // East-West Design & Test Symposium (EWDTS). – 2016. –P. 1-6. doi : <https://doi.org/10.1109/EWDTS.2016.7807655>.
6. Кучук, Г. А. Модель процесса эволюции топологической структуры компьютерной сети системы управления объектом критического применения / Г. А. Кучук, А. А. Коваленко, А. А. Янковский // Системы обработки информации: сборник научных трудов. – Х.: ХУ ВС, 2014. – Вып. 7 (123). – С. 93 – 96.
7. Ковальчук С. В. Облачные вычисления второго поколения: композитные приложения, интерактивные системы и семантические технологии [Электронный ресурс] / С. В. Ковальчук, А. В. Бухановский // МНТК «Инфокоммуникационные технологии». – Таруса, 2012. – Режим доступа: <http://keepslide.com/technology/8702#sthash.YH3i15Gy.dpuf>.
8. Кормен, Т., Лейзерсон, Ч., Ривест, Р., Штайн, К. Глава 16. Жадные алгоритмы // Алгоритмы: построение и анализ = Introduction to Algorithms / Под ред. И. В. Красикова. — 2-е изд. — М.: Вильямс, 2005. — 1296 с.
9. M. Dorigo, G. Di Caro & L. M. Gambardella, 1999. «Ant Algorithms for Discrete Optimization». Artificial Life, 5 (2): 137—172.
10. Хайдуков Д. С. Применение кластерного анализа.— М.: МАКС Пресс, 2009. — 287 с.

**Рецензент:** д-р техн. наук, проф. І. В. Рубан,  
Харківський національний університет радіоелектроніки, Харків  
Received (Надійшла) 28.06.2018  
Accepted for publication (Прийнята до друку) 29.08.2018

**Метод распределения ресурсов  
между композитными отношениями**

С. С. Бульба, В. В. Давыдов, Г. А. Кучук

**Предмет рассмотрения** - методы оптимального распределения ресурсов. **Целью статьи** является разработка метода распределения ресурсов между композитными за отношениями, ориентированном на динамический выбор алгоритма оптимизации. методы, используемые, - жадные и муравьиные алгоритмы, кластеризационный подход. **Результаты.** Рассмотрены обобщения существующих эвристик динамического планирования, которые относятся к классу жадных алгоритмов и находят на каждом шагу локально оптимальное решение. Проведено укрупненную классификацию методов распределения пакета композитных приложений. Проанализированы два типа кластеризации - вертикальную и горизонтальную. Рассмотрена возможность планирования наборов композитных приложений на базе кластеризационного подхода. Приведены способ распределения наборов композитных за отношений как с использованием муравьиного алгоритма, так и на базе кластеризационного подхода с использованием муравьиного алгоритма. Проанализированы преимущества и недостатки каждого из рассмотренных подходов, что позволило определить границы применения каждого из подходов. **Выводы.** Предложено четыре различных подхода к распределению ресурсов между композитными за отношениями с использованием жадных алгоритмов, муравьиных алгоритмов, вертикальной и горизонтальной кластеризации, совместного использования муравьиного алгоритма и кластеризации. Разработанный метод позволяет динамично провести лучший выбор. Дальнейшие исследования будут направлены на разработку соответствующего алгоритма.

**Ключевые слова:** композитный приложение, жадный алгоритм, муравьиный алгоритм, кластеризация.

**Method of distribution of resources  
between composite application**

S. Bulba, V. Davydov, H. Kuchuk

Subject of consideration - methods of optimal allocation of resources. The purpose of the article is to develop a method for distributing resources between composite relationships, focused on the dynamic choice of optimization algorithm. methods used - greedy and ant algorithms, clustering approach. Results of work The generalization of existing heuristics of dynamic planning, which relates to the class of greedy algorithms and finds at each step a locally optimal solution, is considered. An enlarged classification of distribution methods for a composite application package is carried out. Two types of clustering, vertical and horizontal, are analyzed. Considered the possibility of planning sets of composite applications based on clustering approach. The method of distribution of sets of composites for relations using both ant algorithm and on the basis of a clustering approach using an ant algorithm is given. The advantages and disadvantages of each of the considered approaches were analyzed, which made it possible to determine the limits of application of each approach. Conclusions A four different approaches to resource allocation for the relationship between the composite using the following methods: Greedy algorithms, ant algorithms, vertical and horizontal clustering, sharing ant algorithm and clustering. The developed method allows dynamically to make the best choice. Further research will be aimed at developing an appropriate algorithm.

**Keywords:** composite application, greedy algorithm, ant algorithm, clusterization.