

О.О. Захаров, Т.В. Селівьорстова

**ПРОГРАМНА РЕАЛІЗАЦІЯ ANDROID ДОДАТКУ
ТРИВИМІРНОГО ПАКУВАННЯ ОБ'ЄКТІВ
ПАРАЛЕЛЕПІПЕДНОЇ ФОРМИ У КОНТЕЙНЕРИ**

Анотація. Розглядаються питання створення програмного забезпечення, для пристроїв на базі операційної системи Android, яке знаходить та візуалізує оптимальне рішення "проблеми ранця" (SKP – Single knapsack problem) щодо паралелепіпедних об'єктів.

Ключові слова: Android, паралелепіпедне пакування, Single knapsack problem, Multy-dimensional knapsack problem.

Вступ

Як відомо, конкурентоспроможність товарів є важливою складовою формування ринкових стратегій. Причому конкурентні переваги набувають товари по низці критеріїв, серед яких вагоме місце займає вартість транспортування товарів, яка в свою чергу безпосередньо залежить від способу транспортування та методики формування контейнерного набору та інше. Питання оптимального пакування товарів у контейнери відноситься до класу задач «проблема ранця» (SKP – Single knapsack problem), яка знаходиться на 18 місці за популярністю досліджень і на 4 по затребуваності з 75 алгоритмічних проблем за версією університету Stony Brook, Нью Йорк. Тому, задача програмної реалізації задачі тривимірного пакування є безперечно затребуваною. Актуальність подібної розробки обумовлена відсутністю аналогів для мобільних пристроїв, які в свою чергу, так само накладають обмеження по ресурси і т.і. Крім того, ця програма може бути імплементована не тільки в традиційні мобільні пристрої, а й завдяки операційній системі Android використовуватись у роботизованих складських комплексах.

Метою роботи є створення мобільного додатку для оптимізації пакування вантажних контейнерів (вантажів) паралелепіпедної форми у контейнери.

Стан питання

Прикладна постановка задачі тривимірного пакування вимагає її розгляду у наступній постановці: існує контейнер, коробки, які необхідно упакувати в нього, мають геометричні відмінності. В даному випадку задача стає NP-повною, тобто не існує відомого досконалого алгоритму, який виконується за поліноміальний час. Складність у математичному описі даної задачі виникає з причини існування великої кількості складно формалізованих обмежень. Наприклад, умови розподілення вантажу по вазі, при цьому розподіл вантажу в контейнері повинен відбуватися з урахуванням навантаження на вісі колісного транспорту, або з урахуванням крену морських кораблів. Або, наприклад, вимоги щодо рядності або неможливості перевертати коробки з тендітним вантажем. І таких прикладів можливо навести безліч, що і робить кожну задачу по розробці алгоритмів та програм для пакування вантажів майже унікальною.

Обмеження і технічне завдання

Хоча сама по собі класична задача передбачає знаходження рішення по вазі, в рамках даної постановки задачі розглядається пакування коробок паралелепіпедної форми в контейнер в тривимірному просторі, якій характеризується висотою, довжиною і шириною. Відповідно задача має обмеження:

1. коробки можуть розміщуватися тільки таким чином, щоб їх границі були паралельні стінам контейнера;
2. всі коробки повинні бути упаковані всередину контейнера, не перетинаючи границі контейнера;
3. коробки не повинні перетинатися одна з одною;
4. коробки не повинні мати вільний простір під нижньою границею;
5. коробки можуть вільно змінювати положення за трьома вимірами.

Рішення даної задачі полягає в знаходженні оптимального розташування коробок для мінімізації вільного простору.

Технічне завдання так само пред'являє низку вимог до додатка:

- наявність користувацько-орієнтованого інтерфейсу, розрахованого на працівників, які не є фахівцями в галузі інформатики та володіють мінімальними для користувача знаннями системи ОС Android;

- вихідні дані строго структуровані, для можливої імплементації у роботизований комплекс;
- можливість імпорту даних з бухгалтерських програмних комплексів в XML форматі;
- покрокова візуалізація процесу пакування з метою її використання в якості інструкції щодо заповнення контейнера для будь-якого користувача;
- збереження набору введених даних для можливості їхнього повторного використання.

Модулі програми

З метою спрощення розробки, програму декомпоновано на кілька модулів, кожен з яких виконує своє завдання і пов'язаний з іншими, рисунок 1.

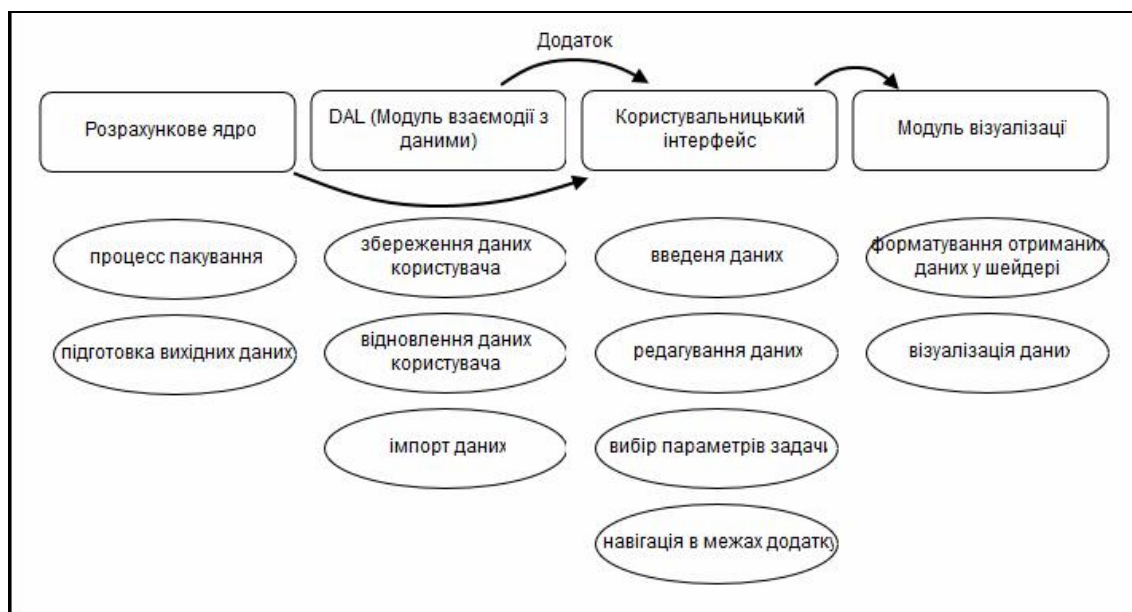


Рисунок 1 - Склад модулів програми

Інтерфейс користувача - забезпечує валідацію даних, які вводяться користувачем на етапі вводу, що запобігає можливість маніпуляції даними, так само він забезпечує користувач-орієнтовану функцію, надаючи користувачеві підказки в потрібних місцях.

Розрахункове ядро - основний модуль програми реалізує алгоритм вирішення задачі і форматує отримані результати для їх подальшого використання.

Модуль взаємодії з даними - даний модуль забезпечує завантаження і збереження введених користувачем даних за допомогою

інтерфейсу Serializable, а також забезпечує імпорт даних в форматі XML і його перетворення в формат даних для програми.

Модуль візуалізації - обробляє отримані від розрахункового ядра дані з метою їх візуалізації за допомогою технології Open GL ES, на кожному кроці процесу проводиться дискретизація, яка використовується для створення візуальної інструкції для користувача.

Вибір алгоритму

Відомо, що «проблема ранця» не має поліноміального алгоритму, до того ж існує велика кількість узагальнень цієї задачі [1 - 4]. Виходячи з наведених обмежень, задача, що розглядається, може бути віднесена до класу задач «проблеми багатовимірного ранця» - Multy-dimensional knapsack problem, коли кожна коробка має кілька ресурсів, замість одного – ваги. Існують різні алгоритми рішення даної задачі, деякі з них дозволяють отримати точне рішення, інші – наближене.

До наближених методів відносяться:

- Жадібний алгоритм (Greedy approximation algorithm);
- Наближена схема повністю поліноміального часу (Fully polynomial time approximation scheme);
- Генетичні алгоритми (genetic algorithm).

Однак згідно обмеженням задачі, що розглядається, бажано отримати оптимальний розв'язок, що виключає використання наближених алгоритмів.

У свою чергу, до точних методів вирішення задачі Multy-dimensional knapsack problem відносяться:

1. Метод повного перебору (Brute force method);
2. Метод гілок і меж (Branch and bound);
3. Методи динамічного програмування (Dynamic programming in-advance algorithm);

Метод динамічного програмування накладає ряд обмежень на умови і не може бути використаний для вирішення «проблеми багатовимірного ранця», а метод гілок і меж - є окремим випадком методу повного перебору. Таким чином рішення задачі паралелепіпедного пакування в межах, поставлених технічним завданням, можливо тільки шляхом декомпозиції методу повного перебору і розробки алгоритмів і методів для кожного етапу.

Дана програмна реалізація є симбіозом декількох алгоритмічних ідей в області розв'язку задачі про пакування у контейнери, зокрема алгоритмів:

- NF «Наступний відповідний». У довільному порядку пакуємо предмети за правилом: перший предмет поміщаємо в перший контейнер, на k -му кроці намагаємося помістити k -й предмет в поточний контейнер. Якщо предмет входить, то поміщаємо його і переходимо до наступного кроку, інакше поміщаємо предмет в новий контейнер.

- FF «Перший відповідний». У довільному порядку пакуємо предмети за правилом: перший предмет поміщаємо в перший контейнер, На k -му кроці знаходимо контейнер з найменшим номером, куди поміщається k -й предмет, і поміщаємо його туди. Якщо такого контейнера немає, то беремо новий порожній контейнер і поміщаємо предмет в нього.

У реалізованому алгоритмі введені деякі додаткові операції, а саме: проводиться сортування набору коробок в порядку зменшення значущого геометричного розміру (наприклад, найбільші спочатку), далі намагаємося помістити коробку починаючи з переднього лівого нижнього кута, в разі успіху отримуємо по суті три додаткових контейнера замість одного (рисунок 2, а). На наступній ітерації оптимізуємо контейнери, намагаючись отримати найбільші за обсягом варіанти контейнерів, з урахуванням максимальної площі нижньої грані, виходячи з того, що контейнер не може мати нижню межу над іншим контейнером.

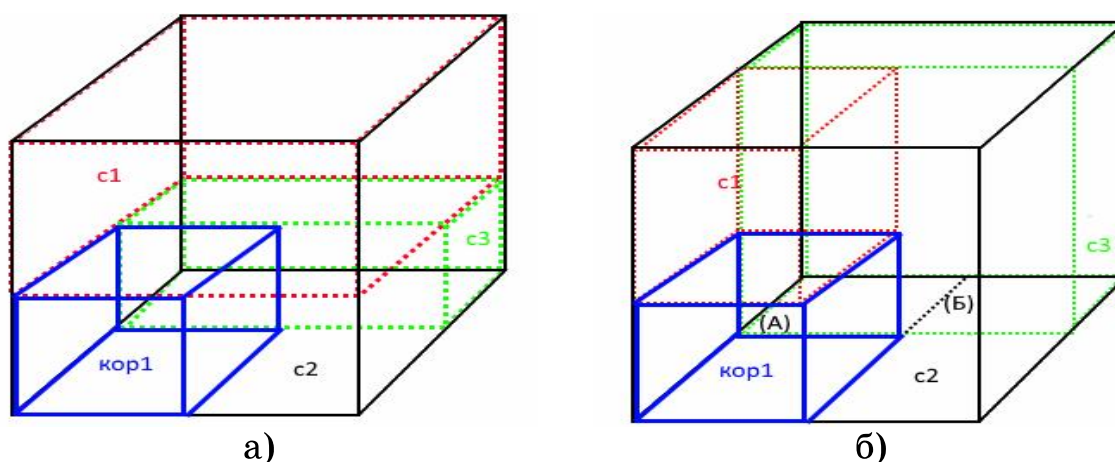


Рисунок 2 – Візуалізація першого кроку розробленого алгоритму тривимірного пакування коробок паралелепіпедної форми

На рисунку 2, б наведений результат розбиття контейнеру на субконтейнери. Контейнер c3 міг бути розбитий по лінії Б, а його права частина приєднана до c2, якби площа нижньої межі була більше у цьому випадку. Далі вибираємо контейнер, який має найбільш близьке розташування до нижнього, лівого, переднього кута, причому саме в такому порядку. Тобто береться контейнер, в нашому випадку це буде c3 кут (А), далі процес проводиться рекурсивно. В загалі кажучи, намагаємося вмістити найбільшу коробку в контейнер згідно наведених вище міркувань, якщо коробка не вміщується, беремо наступний контейнер, якщо дані коробки не входять ні в один з контейнерів беремо меншу коробку і т.д. на кожному етапі оптимізуємо вільний простір контейнерів. На рисунку 3 представлений наступний крок розподілу коробок у контейнері.

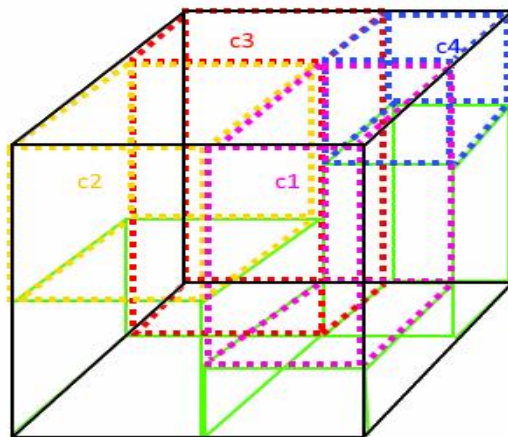


Рисунок 3 – Візуалізація другого кроку розробленого алгоритму тривимірного пакування коробок паралелепіпедної форми

На рисунку 3 пунктиром відзначені порожні місця - контейнери (їх кількість може доходити до декількох десятків), далі процес розподілу коробок у контейнері повторюється за принципом початкового етапу, за невеликим винятком, рекурсивна функція прораховує варіанти перевертаючи коробки, що залишилися. На кожному кроці обчислюється об'єм заповнення і розташування коробок та порівнюється з еталонним (найкращим який був обчислений раніше), якщо він більше ніж попередній, записуємо його в еталон і т.д. У підсумку на виході отримуємо набір вершин і розмірів з першого етапу і кращий з другого та наступних, які об'єднані до масиву значень.

Модуль візуалізації

Одним з основних модулів програми є модуль візуалізації. Дані отримані від розрахункового ядра, модуль візуалізації форматує у вигляді вершин і граней об'єктів за допомогою шейдерів графічного процесору, після чого виводить ці дані на екран [5]. У зв'язку з тим, що отримані дані представляють собою масив об'єктів в заданих координатах, модуль має можливість дискретизувати візуалізацію їх розміщення, а шейдери графічного процесора в свою чергу - змінити масштаб і трансформувати зображення в необхідних користувачеві проекціях (рисунок 4).

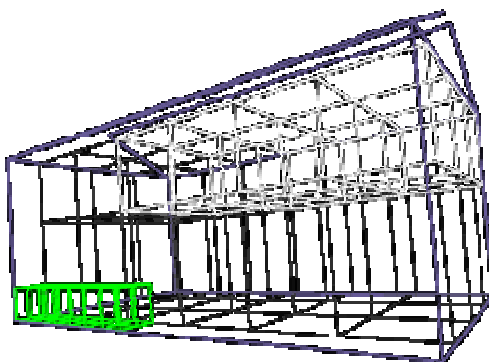


Рисунок 4 – Візуальне представлення процесу пакування

Висновки

В роботі розглянуті основні питання розробки програми для тривимірного пакування паралелепіпедних об'єктів, а саме:

- модулі з яких складається програмне забезпечення;
- зроблено аналіз існуючих алгоритмів та обґрунтовано вибір необхідного алгоритму для вирішення поставленої задачі – пакування коробок в контейнер в тривимірному просторі;
- наведено приклад роботи модуля візуалізації, що вирішує проблеми описані в технічному завданні.

ЛІТЕРАТУРА

1. Garey, Michael R.; David S. Johnson - Computers and Intractability: A Guide to the Theory of NP-Completeness. - W.H. Freeman.1979. - 247 p. ISBN 0-7167-1045-5. А6: МР9,
2. Окулов С. Программирование в алгоритмах. – 1-е. – Бином. Лаборатория знаний, 2007. – 384 с. – ISBN 5-94774-010-9.
3. Kellerer H., Pferschy U., Pisinger D. Knapsack Problems – Springer Science+Business Media, 2004. – 548 p. – ISBN 978-3-642-07311-3
4. Silvano Martelo, Paolo Toth. Knapsack problems. — Great Britain: Wiley, 1990. – 306 с. – ISBN 0-471-92420-2.
5. Эффективное использование потоков в операционной системе Android. Технологии асинхронной обработки данных. Ёранссон А. – ДМК Пресс, 2015. Пер. с англ. – 300 с.