

УДК 004.021

DOI: <https://doi.org/10.17721/1812-5409.2024/1.24>

Василь ТЕРЕЩЕНКО, д-р фіз.-мат. наук, проф.
ORCID ID: 0000-0002-0139-6049
e-mail: vtereshch@gmail.com

Київський національний університет імені Тараса Шевченка, Київ, Україна

Максим ОСІПЬОНОК, асп.
ORCID ID: 0000-0002-3372-3032
e-mail: osipyonok@ukr.net

Київський національний університет імені Тараса Шевченка, Київ, Україна

ПЛАНУВАННЯ ПРОЦЕСУ АДИТИВНОГО ВИРОБНИЦТВА: ОГЛЯД ЗАДАЧ І МЕТОДІВ ЇХНЬОГО РОЗВ'ЯЗАННЯ

Подано огляд актуальних методів і підходів до процесу планування адитивного виробництва (АВ). Наведено аналіз ключових етапів сучасного процесу планування АВ, виокремлено алгоритмічні проблеми та задачі, що постають під час підготовки моделей до 3D-друку, які є актуальними для більшості наявних технологій АВ. До таких задач належать: конвертування САД-даних у меш, генерування мешів, булеві операції над мешами, генерування опорних конструкцій, задача пакування та слайсинг. Кожній із зазначених задач присвячено окремий підрозділ, де наведено постановку задачі, обґрунтування її ролі в процесі планування АВ та розглянуто ключові підходи до її розв'язання. За результатами огляду сучасних підходів до розв'язання зазначених задач у статті подано аналіз структур даних, які необхідно підтримувати комплексній програмній системі планування адитивного виробництва, що реалізує ці алгоритми, базуючись на традиційному пакетно-алгоритмічному підході. Увага акцентується на неефективності застосування такого підходу, оскільки він передбачає окрему реалізацію кожного методу розв'язання тієї чи іншої задачі, ігноруючи взаємозв'язки між ними. Натомість у роботі запропоновано використати альтернативний підхід до побудови такої програмної системи – модель єдиного алгоритмічного середовища (МЄАС). Ключовою ідеєю МЄАС є врахування взаємозалежності задач, що дає змогу будувати ефективні та гнучкі програмні системи. Вивчення оптимальної структури даних для МЄАС визначено як один із перспективних напрямків майбутніх досліджень у галузі планування процесу адитивного виробництва.

Для дослідження сучасного процесу планування адитивного виробництва було застосовано такі методики: літературний огляд та аналіз алгоритмів, порівняльний аналіз структур даних, що лежать в основі розглянутих алгоритмів, критичний аналіз сучасних підходів, синтез альтернативного підходу (МЄАС).

За результатами аналізу було виявлено, що застосування традиційного пакетно-алгоритмічного підходу до побудови комплексної системи планування процесу адитивного виробництва не дає змоги ефективно розв'язувати всю сукупність необхідних задач.

З огляду на виявлену неефективність пакетно-алгоритмічного підходу у роботі пропонується застосувати альтернативний підхід, що базується на МЄАС.

Ключові слова: 3D-друк, планування процесу адитивного виробництва, генерування мешів, генерування опорних конструкцій, задача пакування, слайсинг, модель єдиного алгоритмічного середовища.

Класифікація відповідно до AMS 2020: 68U05, 68U07

Вступ

Адитивне виробництво, також відоме як 3D-друк, це технологія виробництва тривимірних об'єктів шляхом пошарового нанесення матеріалу. Перші технології адитивного виробництва виникли у 1980-х рр., їхньою основною метою було спрощення та прискорення прототипування нових деталей, їхня візуалізація, а також надання можливості перевірити взаємодію нової деталі з уже наявними (Wohlers, & Gornet, 2014). З розвитком технологій 3D-друку дедалі очевиднішими ставали її переваги перед традиційними технологіями виробництва: мінімізація відходів виробництва, можливість децентралізації виробничих потужностей, масова персоналізація виробництва, можливість виробництва деталей складної форми, які неможливо виготовити традиційними методами виробництва (Huang et al., 2012; Kwok et al., 2017). Завдяки цим перевагам адитивне виробництво нині використовують у таких сферах, як аерокосмічна промисловість, автомобілебудування та медицина (Chiu, 2020; Salmi, 2021; Valdivieso, 2021).

Поширення сфер застосування адитивного виробництва, а також підвищення складності дизайну деталей спричинило й зростання складності процесу підготовки моделей до друку. На початкових етапах упровадження адитивного виробництва цей процес зазвичай складався з таких кроків: оптимізації орієнтації моделі, побудова опорних конструкцій, слайсинг і визначення траєкторії друку (Kulkarni, Marsan, & Dutta, 2000). Однак нині ситуація децю змінилась. Методи та підходи, описані вченими (Kulkarni, Marsan, & Dutta, 2000), розраховані на друкування однієї моделі або ж пропонують користувачу слідкувати за взаємним впливом різних моделей під час одночасного друкування. Таке друкування багатьох моделей дає змогу значно зменшити витрати часу без втрати якості завдяки окремим налаштуванням слайсингу для кожної з моделей (Simplify3D Software, 2019), тому цей підхід є популярним. Один із його наслідків – виникнення багатьох нових задач, які необхідно розв'язувати під час підготовки платформи до друку. По-перше, моделі необхідно розмістити на платформі так, щоб вони не перетинались та були розділеними. Також часто зустрічаються випадки, коли модель, яку необхідно надрукувати, потребує попередньої обробки. До таких випадків належать артефакти конвертування моделі з одного представлення в інше (напр., із САД-формату в меш), або нехтування обмеженнями технології друку під час проектування моделі (напр., постобробка моделі за допомогою фрези (Laue et al., 2022)). Іншим прикладом попередньої обробки є генерування внутрішніх пустот у моделі для зменшення її ваги, прискорення друку та економії матеріалу (Akhouri et al., 2021). Після виправлення дефектів моделей, модифікації геометрії та їх розміщення на платформі необхідно побудувати опорні конструкції, водночас алгоритм має враховувати не лише геометрію окремо взятій моделі, а й інших моделей та їхніх опорних конструкцій. Наступний крок підготовки – слайсинг, також має виконуватись індивідуально для кожної з моделей, оскільки різні моделі можуть вимагати різної точності друку для досягнення прийнятної якості надрукованої поверхні. Зрештою, планування траєкторії друку також має враховувати наявність багатьох моделей на платформі одночасно.

© Терещенко Василь, Осіпюнок Максим, 2024

Як бачимо, процес підготовки моделей до друку охоплює великий комплекс взаємопов'язаних задач, які необхідно ефективно розв'язувати. У другій частині запропонованої роботи наведено детальний опис кожної із задач та розглянуто наявні підходи до їхнього розв'язання.

1. Огляд задач. Як ми зазначали, у роботі (Kulkarni, Marsan, & Dutta, 2000) процес планування друку в адитивному виробництві запропоновано розділити на чотири задачі: слайсинг, визначення орієнтації моделі, генерування опорних конструкцій і побудова траєкторії друку. Також автори пропонують розділити ці задачі на дві категорії: задачі домену моделі – тобто задачі, що потребують інформації про геометрію всієї моделі, та задачі домену шару – ті, що працюють над окремо взятими шарами моделі. Ми ж, зі свого боку, пропонуємо розглядати процес планування друку без обмежень на кількість моделей та розширити категорію задач домену моделі, залучивши до неї задачі, що працюють над усіма моделями одночасно або, працюючи над однією моделлю, ураховують геометрію решти.

1.1. Генерування мешів. Задача генерування мешів під час підготовки постає в різних формах. По-перше, дизайн вхідних моделей для друку зазвичай розробляють у системах автоматизованого проектування (CAD systems), вихідними даними яких є файл, що містить інформацію про представлення моделі у вигляді орієнтованих граней і кривих. Прикладом такого файлового формату є STEP (International Organization for Standardization (ISO, 2016)). Однак, де-факто, стандартом представлення моделей в адитивному виробництві є формат STL, що являє собою набір трикутників та їхніх нормалей. Отже, актуальною є задача конвертування даних із формату CAD в STL. Іншою варіацією задачі генерації мешів є задачі модифікації геометрії моделі. До таких модифікацій можна віднести додавання офсету, заокруглення ребер і кутів, генерування пустот усередині моделі та булеві операції над моделями (об'єднання, перетин і різниця моделей).

Підходи до розв'язування задачі генерування мешів можна розділити на загальні, тобто ті, що можна застосувати для будь-якої постановки задачі генерування, та спеціальні, тобто такі, що розв'язують лише конкретну постановку задачі генерування.

1.1.1. Конвертування CAD в STL. На відміну від мешу, CAD-дані можуть описувати геометрію довільної форми та кривизни за допомогою, наприклад, поверхонь Безье або неоднорідних раціональних B-сплайнів (NURBS). З огляду на це меш, що є наслідком конвертації, лише наближено передає геометрію CAD-моделі. Точність наближення може бути задано параметром алгоритму або визначатись автоматично, зважаючи на розміри моделі та технічні характеристики принтера (товщина лазера тощо).

Вхідні дані: модель у CAD-форматі

Вихідні дані: меш трикутників, що з певною точністю репрезентує геометрію вхідної моделі в її початковій орієнтації.

Популярним підходом до конвертування CAD-даних у меш є побудова полігонів на основі циклів, триангуляція області, яку обмежують побудовані полігони, й уточнення триангуляції на основі точок на кривих поверхнях. Алгоритм, що використовує такий підхід, запропоновано в роботі (Yang, & Choi, 2010). Процес побудови мешу автори поділяють на чотири етапи:

1. Побудова обмеженої триангуляції Делоне (ОТД) на ребрах, що апроксимують границю граней CAD-даних.
2. Побудова kd -дерева для вершин триангуляції, отриманої на попередньому кроці.
3. Уточнення триангуляції за допомогою внутрішніх точок поверхні.
4. Видалення «зайвих» трикутників, що перебувають зовні вхідної моделі.

Спершу цикли на границі граней конвертують у ламані, що апроксимують границю. Під час побудови ламаних, відрізки додають до ОТД. На другому кроці алгоритму відбувається побудова kd -дерева для вершин відрізків, знайдених на першому кроці. Автори застосовують модифіковану структуру даних *interworld* (Kim et al., 2006) для збереження триангуляції, та використовуючи той факт, що точки у структурі *interworld* зберігаються у вигляді звичайного масиву, представляють kd -дерево у вигляді перестановки точок у масиві, уникаючи у такий спосіб зайвого використання пам'яті. На третьому кроці відбувається апроксимація кривизни поверхні: спершу параметричну область поверхні розбивають на сітку. У разі, якщо відхилення центру сітки від поверхні більше за наперед задану точність апроксимації, розбиття сітки продовжується. Після досягнення необхідної точності ОТД доповнюють трикутниками, однією з вершин яких є нова точка з поверхні моделі, а інші визначаються відповідно до поточної конфігурації ОТД. На останньому кроці алгоритму відбувається фільтрація трикутників – ті, що перебувають за межами поверхні, видаляють. Зазначається, що складністю алгоритму в найгіршому випадку є $O(n \log n)$, де n – кількість відрізків, що задають границю поверхні.

Інший підхід до конвертування CAD-даних у меш запропоновано в роботі (Yu et al., 2021). Як слушно зазначають автори, меші, отримані внаслідок конвертування із CAD-форматів, часто мають багато дефектів, наприклад самоперетини, невідповідності сусідніх трикутників та дірки. Для виправлення цих та інших дефектів, на практиці застосовують різноманітні методи відновлення мешів, опис яких можна знайти в роботі (Attene, Campen, & Kobbelt, 2012). Однак точне відновлення поверхні не завжди можливе або вимагає наявності додаткової інформації, тому в (Yu et al., 2021), окрім методу побудови меша, запропоновано також метод синхронізації та вирівнювання меша. Завдяки такому підходу, меші окремих кривих і поверхонь CAD-даних об'єднуються в єдиний меш без погіршення якості меша.

1.1.2. Загальні алгоритми генерування мешів. У цьому підрозділі будуть розглянуті алгоритми генерування мешів, які можна застосувати напряму або адаптувати до широкого класу задач, що виникають як у процесі планування платформи в адитивному виробництві, так і загалом під час роботи із тривимірними моделями. Прикладом таких задач може бути вже розглянута в підрозд. 1.1.1 задача конвертування CAD-даних у меш, булеві операції над мешами та різноманітні модифікації геометрії: генерування внутрішніх пустот, заокруглення кутів тощо. Алгоритми цього класу використовують поняття *знакової функції відстані*, що дає змогу задати очікувану поверхню в неявному вигляді.

Визначення. Нехай $D \subset R^N$ – підмножина евклідового простору розмірності $N \geq 2$. Позначимо $\partial D \neq \emptyset$ границю D в R^N , \bar{D} – замикання множини D в R^N , $d(x, y)$ – метрика простору R^N . Знакову функцію відстані $f(x), x \in R^N$ для множини D визначають так:

$$f(x) = \begin{cases} d_{\partial D}(x), & x \in \bar{D} \\ -d_{\partial D}(x), & x \in R^N \setminus \bar{D} \end{cases}, \text{ де } d_{\partial D}(x) = \inf_{y \in \partial D} d(x, y).$$

Формально загальну задачу генерування меша можна визначити у такий спосіб:

Вхідні дані: $f(x)$ – знакова функція відстані для множини $D \subset R^3$, що задає геометрію шуканого меша.

Вихідні дані: меш трикутників, що репрезентує ∂D .

Одним з найперших підходів до розв'язання задачі генерування меша за знаковою функцією відстані є алгоритм Marching cubes (MC) (Lorensen, & Cline, 1987). Уперше цей алгоритм був запропонований ще в 1987 р., та позиціонувався як алгоритм відновлення тривимірних моделей за медичними даними, що містять результати комп'ютерної томографії, магнітно-резонансної томографії й однофотонної емісійної комп'ютерної томографії. Однак він знайшов застосування в багатьох сферах, зокрема і в адитивному виробництві, та його продовжують нині широко використовувати.

На першому кроці алгоритму MC простір розбивають на рівномірну сітку з наперед заданою точністю, та в вершинах сітки обчислюють значення знакової функції відстані $f(x)$. Очевидно, що клітинки (куби) сітки, на вершинах яких значення $f(x)$ змінює знак, мають непорожній перетин із ∂D . Варто зауважити, що в загальному випадку обернене твердження не є правильним, найпростішим контрприкладом є функція $f(x)$, що задає область D , яка збігається з однією з комірок сітки, однак на практиці такі випадки обходяться вибором достатньо малої точності алгоритму (вибором довжини сторони комірки). Припускаючи, що задана точність алгоритму адекватно відображає очікувану якість відновлення поверхні $f(x)$, алгоритм розглядає всі 256 різноманітних конфігурацій знаків значення $f(x)$ у вершинах кубів, які шляхом повороту та відображення можна звести до 15 базових конфігурацій, що зображені на рис. 1.

Визначивши відповідну кожній комірці конфігурацію, алгоритм розміщує вершини мешу на ребрах кубів, положення яких визначається за допомогою лінійної інтерполяції значення знакової функції відстані на кінцях відповідного ребра, та до мешу додаються відповідні даній конфігурації трикутники.

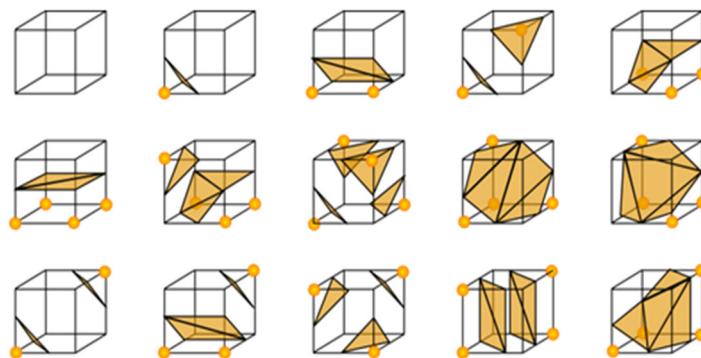


Рис. 1. 15 базових конфігурацій вузлів кубів для алгоритму MC (Marching cubes, 2023)

Значним недоліком такого підходу є недостатня якість апроксимації гострих ділянок поверхні. Наприклад, застосувавши двовимірний аналог алгоритму Marching cubes – Marching squares для генерування квадрата, замість очікуваних прямих кутів отримаємо квадрат зі зрізаними кутами (рис. 2). Очевидно, що якість апроксимації можна поліпшити, зменшивши довжину ребра комірки розбиття простору, однак цей підхід однаково не дає змоги отримати точні прямі кути квадрата, водночас збільшуючи кількість кубів, які розглядає алгоритм, а отже, і час його роботи. Також недоліком Marching cubes є наявність неоднозначних конфігурацій вузлів, що вимагає залучення певної стратегії усунення неоднозначності, вибір якої може різнитись залежно від вимог до алгоритму та сфери його застосування, адже непослідовна стратегія може призвести до утворення дірок у меші. З іншого боку, стратегія усунення неоднозначності, що враховує конфігурації вузлів сусідніх кубів, значно ускладнює можливість паралелізації алгоритму, оскільки куби вже не є незалежними один від одного.

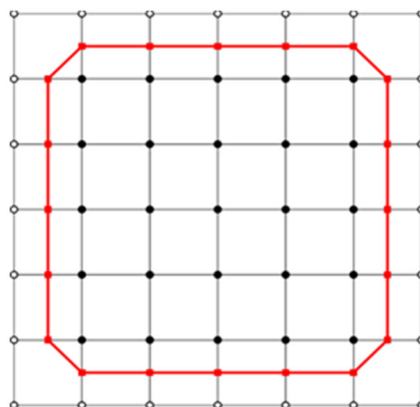


Рис. 2. Апроксимація квадрата алгоритмом MC (Boris, 2018)

З метою подолання зазначених недоліків було запропоновано низку альтернативних підходів (Ho et al., 2005; Kobbelt et al., 2001; Schaefer, Ju, & Warren, 2007), однак особливої уваги заслуговує алгоритм Cubical marching squares (CMS) (Ho et al., 2005), оскільки він розв'язує проблему репрезентації гострих ділянок поверхні, водночас залишаючи куби незалежними один від одного.

Ключовою ідеєю алгоритму CMS є зведення тривимірної задачі полігоналізації до двовимірної, а саме перехід від розгляду тривимірного куба в МС до шести двовимірних квадратів, утворених шляхом розгортання куба. Замість рівномірної сітки пропонується застосовувати октодерево, розбиття вершин якого продовжується, доки всі вершини-листки не будуть мати однозначну конфігурацію вузлів та не міститимуть «складної» поверхні (також автори наводять евристичний критерій перевірки «складності» поверхні). Далі для всіх вершин-листок генерується набір сегментів, використовуючи двовимірний аналог алгоритму МС. На фінальному етапі алгоритм відновлює поверхню, задану сегментами, знайденими на попередньому кроці, та триангулює її.

1.1.3. Булеві операції над мешами. Окремою задачею генерування мешів є булеві операції над ними. Зазвичай розглядають три базові булеві операції над мешами: об'єднання, перетин і різниця. Формально визначити їх можна так:

Визначення. Нехай M_1 та M_2 – два меши. Тоді меш M_U , для якого виконується умова $\forall x \in R^3$

$x \in M_U \Leftrightarrow x \in M_1 \vee x \in M_2$, називають об'єднанням мешів M_1 та M_2 , меш M_{\cap} , для якого виконується умова $\forall x \in R^3$ $x \in M_{\cap} \Leftrightarrow x \in M_1 \wedge x \in M_2$, – перетином мешів M_1 та M_2 , і меш M_{\setminus} , для якого виконується умова $\forall x \in R^3$ $x \in M_{\setminus} \Leftrightarrow x \in M_1 \wedge x \notin M_2$, називають різницею мешів M_1 та M_2 .

Булеві операції над мешами в адитивному виробництві можна застосовувати для генерування опорних конструкцій, генерування внутрішніх пустот, створення складних поверхонь із примітивів тощо.

Задачу виконання булевої операції над двома мешами можна задати так:

Вхідні дані: M_1, M_2 – два меши, та операція, яку необхідно виконати (об'єднання, перетин або різниця).

Вихідні дані: Меш M_U, M_{\cap} або M_{\setminus} , відповідно до вхідної операції.

Одним з популярних підходів до виконання булевих операцій є застосування алгоритмів відтинання, наприклад (Scherzinger, Vrix, & Hinrichs, 2017), однак доцільніше використовувати алгоритми, спеціально розроблені для розв'язку задачі виконання булевих операцій на мешах. Один з таких алгоритмів запропоновано в роботі (Jiang et al., 2016).

Спершу алгоритм апроксимує спільну частину двох мешей за допомогою побудови дерев октантів (Meagher, 1980) та виділення їхньої спільної частини. Далі у вузлах «спільного» дерева октантів визначають пари трикутників, що перетинаються, та належать різним мешам. Для цих пар знаходять перетини трикутників, вхідну триангуляцію моделей уточнюють додаванням перетинів трикутників. Також визначають положення вершин трикутників, що перетинаються, відносно іншого меша, та вершинам присвоюють відповідні мітки: «всередині», «ззовні» та «на поверхні». На основі цих міток визначають такі множини трикутників:

- $M_{M_1 \text{ in } M_2}$ – множина трикутників з M_1 , що розміщуються всередині M_2 .
- $M_{M_1 \text{ out } M_2}$ – множина трикутників з M_1 , що містяться ззовні M_2 .
- $M_{M_2 \text{ in } M_1}$ – множина трикутників з M_2 , що розміщуються всередині M_1 .
- $M_{M_2 \text{ out } M_1}$ – множина трикутників з M_2 , що перебувають ззовні M_1 .
- $M_{\text{on } M_1 M_2}$ – множина трикутників, що одночасно належать M_1 та M_2 .

З огляду на це результати булевих операцій можна задати такими співвідношеннями:

- $M_U = M_{M_1 \text{ out } M_2} + M_{M_2 \text{ out } M_1} + M_{\text{on } M_1 M_2}$,
- $M_{\cap} = M_{M_1 \text{ in } M_2} + M_{M_2 \text{ in } M_1} + M_{\text{on } M_1 M_2}$,
- $M_{\setminus} = M_{M_1 \text{ out } M_2} + M_{M_2 \text{ in } M_1}$.

1.2. Розміщення моделей усередині робочої області принтера. Задача розміщення моделей усередині робочої області принтера належить до ширшого класу оптимізаційних задач – задач пакування. З позиції практики, ця задача є важливою складовою процесу планування адитивного виробництва, оскільки ефективне використання всієї робочої області дає змогу збільшити кількість деталей, що одночасно друкуються, тим самим прискорюючи виробничий процес. Водночас велика кількість задач пакування належать до класу NP-складних задач (Pankratov, Romanova, & Litvinchev, 2020), тому більшість досліджень у цій галузі зосереджені на побудові оптимізаційних алгоритмів, що адекватно апроксимують глобально-оптимальний розв'язок за прийнятний час.

Вхідні дані: Множина тривимірних моделей і габарити робочої області принтера.

Вихідні дані: Множина трансформацій, після застосування яких до відповідних моделей вхідної множини всі моделі будуть розміщуватись усередині робочої області принтера.

Цікавий підхід до розв'язку цієї задачі запропоновано в роботі (Lutters, Dam, & Faneker, 2012). Ключовою ідеєю алгоритму є використання процедури, що нагадує феномен гранулярної конвекції, також відомий як ефект бразильського горіха – фізичного явища, що виявляється в сегрегації за розміром суміші частинок різного розміру під час струшування (Rosato et al., 1987). Алгоритм складається із двох етапів:

- оптимізація орієнтації деталей,
- побудова початкового розміщення та його поліпшення.

Для визначення оптимальної орієнтації автори пропонують використати такі критерії:

- мінімізація об'єму мінімального обмежувального паралелепіпеда, вирівняного за осями координат;
- мінімізація площі поверхні деталі, що не є паралельною або перпендикулярною до напрямку друку (Lan et al., 1997).

Зазначені критерії об'єднуються в один скалярний критерій за допомогою вагових коефіцієнтів, що дає змогу користувачу алгоритму регулювати вплив кожного з критеріїв на вибір орієнтації деталі, залежно від технології друку, матеріалу, технічних вимог до якості поверхні й інших міркувань.

Визначивши орієнтацію деталей, алгоритм випадковим чином будує початкове розміщення деталей усередині робочої області принтера, й імітує вібрації, що застосовуються до деталей. Варто зауважити, що імітування вібрацій є складною процедурою з позиції обчислень. Це обумовлено необхідністю підтримувати положення всіх деталей, яке б

не утворювало перетинів або замикань. З метою спрощення обчислювальної складності перевірки деталей на перетин, алгоритм використовує різні підходи до апроксимації поверхні деталей – обмежувальні сфери, мінімальні обмежувальні паралелепіпеди, орієнтовані мінімальні обмежувальні паралелепіпеди, дискретні орієнтовані многогранники (k-DOP) та випуклі оболонки. Вибір методу апроксимації може змінюватись під час роботи алгоритму, завдяки чому можна регулювати якість результуючого розміщення деталей та витрати часу, на його побудову.

Альтернативний підхід до розв'язування задачі пакування в контексті адитивного виробництва запропоновано в роботі (Pankratov, Romanova, & Litvinchev, 2020). Цільовою функцією сформульованої оптимізаційної задачі є об'єм контейнера, який необхідно мінімізувати. Обмеженнями задачі постають умови належності всіх об'єктів контейнеру та відсутність перетинів між об'єктами. Для представлення обмежень в аналітичному вигляді автори використовують інструментарій ϕ -функцій (Chernov, Stoyan, & Romanova, 2010) та квазі- ϕ -функцій (Stoyan et al., 2019). У роботі зазначається, що побудована оптимізаційна задача складається з $O(N^2)$ нелінійних нерівностей та $O(N^2)$ змінних, де N – кількість об'єктів, які необхідно розмістити. Відповідно, через велику кількість обмежень і змінних, використання програмних пакетів пошуку глобального оптимуму для розв'язку цієї задачі займає багато часу. З огляду на це автори наведено алгоритм пошуку локального оптимуму, результат якого можна використовувати як початкове наближення розв'язку для пошуку глобального оптимуму, або як апроксимацію розв'язку задачі пакування.

1.3. Генерування опорних конструкцій. Під час друкування моделі важливо забезпечити стабільне з'єднання моделі з платформою принтера або попередніми шарами моделі. Інакше є високою ймовірність виникнення дефектів, наприклад провисання матеріалу, деформація геометрії або навіть повне руйнування деталі, яку друкують. У багатьох сферах, де використовують адитивне виробництво, як-от авіабудування, медицина тощо, навіть невелике відхилення результуючої поверхні від очікуваної є неприпустимим, тому для забезпечення можливості отримання якісних деталей, що мають складну форму, необхідно разом з моделлю додатково друкувати опорні структури.

Вхідні дані: Модель із фіксованою орієнтацією.

Вихідні дані: Опорні структури, що забезпечують можливість друку вхідної моделі.

Одним з популярних підходів до побудови опорних структур є використання наперед заданих примітивів, що певним чином об'єднуються один з одним, оптимізуючи структуру опори. Автори в роботі (Vaidya, & Anand, 2016) використовують зрізані октаедри та ромбічні додекаедри з пустотами всередині. На першому кроці виконують вокселізацію моделі, спершу модель апроксимують кубами, а далі ця апроксимація конвертується з кубічної вокселізації у вокселізацію примітивів, які використовуються для генерування підтримки, та визначаються вокселі інтерфейсу, тобто такі, що безпосередньо з'єднуються з моделлю. Серед цих вокселів, виділяють групи 3×3 та 2×2 сусідніх вокселів, що розміщуються на одній висоті, та для цих груп формують вертикальні опорні структури. Далі на основі воксельної структури спеціальним чином будують зважений граф, за допомогою якого на останньому кроці алгоритму формують опори для вокселів інтерфейсу, що не потрапили до жодної з груп 3×3 або 2×2 : для кожного такого вокселя знаходять найближчий воксель, що з'єднує вертикальну опору однієї з груп із платформою принтера або моделлю, та для вершин графу, що відповідають цій парі вокселів, знаходять найкоротший шлях між ними за допомогою алгоритму Дейкстри (Dijkstra, 1959), на вокселях зі знайденого шляху будують опору, ваги ребер графу оновлюються та процес продовжується.

Інший підхід до генерування деревовидних опорних структур запропоновано в роботі (Zhang et al., 2020). Спершу алгоритм визначає ділянки, що нависають, для яких необхідно збудувати опори. Автори пропонують вважати такими ділянками ті трикутники, кут нахилу яких перевищує 45° . На зазначених трикутниках генерують точки, в яких майбутні підтримки будуть з'єднуватись з моделлю. Для цього трикутники, для яких необхідно збудувати опори, проєктують на площину XOY , на якій будують рівномірну сітку та знаходять такий поворот сітки, що мінімізує кількість точок сітки, які містяться всередині проєкції трикутників, що нависають на площину. Точки, у яких опори будуть приєднуватись до моделі, визначають шляхом проєктування точок сітки на поверхню моделі. Для генерування деревовидних структур використовують L -системи (Rozenberg, & Salomaa, 1980). З метою забезпечення генерування дерев, які можна друкувати, автори наводять чотири «базові» типи параметричних L -систем, на які накладають обмеження для кута розгалуження гілок дерева – 11.25° , 15° , 22.5° , або 45° . Параметрами є: базова довжина стовбура та коефіцієнт масштабування базової довжини за кожного розгалуження. Процес генерування опор формуюється у вигляді розв'язку багатокритеріальної оптимізаційної задачі: мінімізувати об'єм дерева та кількість перетинів дерева та моделі, що залежить від п'яти змінних: вид L -системи, базова довжина стовбура, коефіцієнт масштабування довжини стовбура при розгалуженні, позиція дерева та його орієнтація в просторі. Для розв'язання цієї задачі автори використовують генетичний алгоритм NSGA-II (Deb et al., 2002), результатом роботи якого є множина оптимальних розв'язків за Парето, для оцінки яких виконують моделювання теплообміну, та зрештою вибирають те дерево, для якого теплова деформація є найменшою. На останньому кроці алгоритму виконують постобробку вибраного дерева – гілки, що перетинаються з моделлю, розділяють на менші за правилами L -системи, та їх кінцівки замінюють конусами, з метою спрощення їхнього видалення після друку.

1.4. Слайсинг. На фінальному етапі підготовки моделі до друку відбувається її слайсинг, тобто нарізання моделі на тонкі шари площинами, перпендикулярними до осі z . Слайси моделі є прототипами тих шарів, що будує принтер, друкуючи модель. Розглядають два види слайсингу: рівномірний слайсинг, за якого товщина всіх шарів є сталою, та адаптивний, за якого товщина шарів може змінюватись. Для підвищення точності апроксимації контуру моделі в місцях з високою деталізацією є сенс використовувати меншу товщину слайсів, а друкуючи ділянки однорідні ділянки, або ті, де вимоги до точності не є строгими (напр. опорні конструкції) товщину слайсів можна збільшити, щоб прискорити процесу друкування.

Задачу слайсингу можна сформулювати так:

Вхідні дані: Вхідна модель та функція $h(z)$, яка повертає товщину слайса, що починається на висоті z . У разі рівномірного слайсингу $h(z) = const$.

Вихідні дані: набір полігонів, утворених перетинами моделі з площинами, перпендикулярними до осі z у визначених функцією $h(z)$ місцях.

Асимптотично-оптимальний алгоритм слайсингу запропоновано в роботі (Minetto et al., 2017). Автори доводять, що їхній підхід у найгіршому випадку має складність $O(n \log k + k + m)$, де n – загальна кількість трикутників меша, k – кількість слайсів (кількість площин, якими перетинається модель), m – кількість перетинів трикутників із площинами. Також пропонуються модифікації загального алгоритму для випадку впорядкованих за зростанням мінімальної z координати трикутників моделі, та для випадку рівномірного слайсингу, які дають змогу зменшити витрати часу до $O(n + k + m)$.

Автори розділяють розв'язок задачі слайсингу на такі підзадачі:

- пошук перетинів площин із трикутниками моделі,
- відновлення контурів за перетинами трикутників із площиною,
- визначення направленості контурів (за годинниковою стрілкою або проти годинникової стрілки).

Для ефективного пошуку всіх перетинів площин із трикутниками моделі автори використовують техніку замітаючої площини. Спершу формують список L , де L_i – множина трикутників, мінімальна z координата яких розміщується між i -ю та $i + 1$ -ю площинами. У загальному випадку, побудова списку L виконується визначенням для кожного трикутника відповідного елементу в L , за допомогою бінарного пошуку, що призводить до складності $O(n \log k + k)$. У разі, якщо трикутники впорядковані за мінімальною z координатою, список L можна побудувати, застосувавши техніку двох вказівників, отримавши складність $O(n + k)$. Аналогічну складність можна отримати й у випадку рівномірного слайсингу – у такому разі позиція трикутника у списку визначається цілою частиною від ділення мінімальної z координати трикутника на товщину слайса. Далі алгоритм проходить площинами, починаючи з найнижчої, та підтримує множину A трикутників, що перетинаються з поточною площиною, використовуючи підрахований на попередньому кроці список L , та визначає відрізки перетинів трикутників і площини.

На попередньому кроці алгоритм знайшов неупорядкований набір неорієнтованих відрізків, тепер необхідно із цього набору відновити замкнені контури слайсу. Для розв'язку цієї задачі автори використовують хеш-таблицю, ключем якої є точка на площині відповідного слайсу, а значенням – пара точок, що з'єднані ребрами з відповідним ключем. Відповідно, після додавання всіх відрізків до хеш-таблиці, алгоритм бере перший ключ з таблиці, вилучає його та переходить по першому ребру, що з'єднане з обраною точкою до наступної точки. У такий спосіб отримують перше ребро контуру. Далі процес продовжується, аж поки чергове ребро не приведе у вихідну точку. Якщо після цього в хеш-таблиці ще залишилися значення, процес продовжується для довільного з них. Зрештою, залишається визначити направленість отриманих контурів. Для цього пропонуємо використати підхід, що базується на перевірці належності точки до многокутника (Volpato et al., 2013).

2. Аналіз структур даних, що використовують у процесі планування адитивного виробництва. Розглянемо структури даних, які застосовуються алгоритмами розв'язання зазначених задач.

Для виконання булевих операцій над мешами необхідно побудувати дерева октантів для кожного з мешів, а також уміти визначити спільну частину (перетин) декількох дерев октантів (див. 1.1.3).

Алгоритми, що використовуються для розміщення моделей усередині робочої області принтера, під час своєї роботи здійснюють велику кількість перевірок наявності колізій між моделями (Lutters, Dam, & Faneker, 2012), що може бути виконано за допомогою *BVH*-дерева для трикутників моделі або дерева октантів. У разі застосування підходу, що базується на побудові *phi*-функцій для вхідних моделей (Pankratov, Romanova, & Litvinchev, 2020), доцільним є використання *BVH*-дерева, обмежувальні об'єкти вузлів якого складаються з прямих та скошених круглих циліндрів, конусів і сфер, оскільки саме цими об'єктами алгоритм апроксимує геометрію вхідних моделей.

Генерування опорних конструкцій, залежно від обраного алгоритму, базується на таких структурах даних, як рівномірна воксельна сітка (алгоритм (Vaidya, & Anand, 2016)), або варіації *kd*-дерева, у вузлах якого містяться трикутники (напр. Surface Area Heuristic *kd*-tree (Wald, & Havran, 2006)), для розв'язування таких задач, як пошук найближчої точки на поверхні меша до заданої точки у просторі та трасування променя, що використовуються алгоритмом (Zhang et al., 2020).

Алгоритм слайсингу працює безпосередньо з трикутниками мешу (Minetto et al., 2017), та, як наслідок, може бути застосованим без залучення додаткових структур даних. Однак, якщо вхідна послідовність трикутників є впорядкованою за зростанням мінімальної координати вершин трикутника, то задачу слайсингу можна розв'язати за лінійний час, використовуючи модифікований алгоритм слайсингу (Minetto et al., 2017). Маючи *kd*-дерево або дерево октантів для трикутників моделі, побудувати таку послідовність її трикутників можна за один обхід дерева, подібно до процедури злиття класичного алгоритму сортування Merge Sort (Knuth, 1998, p. 158–166).

Відповідно, програмна система розв'язку комплексу задач планування адитивного виробництва на базі класичного пакетно-алгоритмічного підходу, що містить розглянуті методи, має реалізовувати та підтримувати великий набір структур даних, виконувати попередню та постобробку даних для забезпечення їхньої сумісності між різними алгоритмами протягом виконання програми. Схему архітектури такого програмного комплексу зображено на рис. 3. За такого підходу значна частина часу роботи програмної системи буде витрачатись на побудову структур даних та конвертування вихідних даних одного алгоритму до формату вхідних даних іншого. У наступному розділі запропоновано альтернативний підхід до побудови програмної системи розв'язку комплексу задач планування адитивного виробництва, який дає змогу уникнути зазначених недоліків.

3. Модель єдиного алгоритмічного середовища для розв'язку комплексу задач планування адитивного виробництва. Альтернативою пакетно-алгоритмічного підходу є модель єдиного алгоритмічного середовища (МЄАС) (Tereshchenko, & Anisimov, 2010). Ключовою ідеєю МЄАС є використання спільної уніфікованої структури даних для всіх компонентів програмної системи, завдяки чому зникає необхідність підтримки окремої структури даних для кожного алгоритму, що дає змогу зменшити споживання пам'яті. Також зменшується сукупний час виконання алгоритмів, оскільки попередня обробка виконується лише один раз. Натепер існують приклади ефективного використання МЄАС для розв'язання взаємопов'язаних задач обчислювальної геометрії на базі таких структур даних, як діаграма Вороного та зціплена черга (Tereshchenko, & Anisimov, 2010; Tereshchenko, Budjak, & Fisunen, 2013; Терещенко та ін., 2015).

З огляду на велике різноманіття структур даних, що використовуються на різних етапах планування адитивного виробництва, доцільним є використання підходу МЄАС, який дає можливість не лише позбутися від проміжних етапів

попередньої обробки та конвертування даних, а й спрощує інтеграцію нових алгоритмів до програмної системи, роблячи її гнучкішою, порівняно із класичним підходом. Архітектуру такого програмного комплексу зображено на рис. 4.



Рис. 3. Схематична архітектура пакетно-алгоритмічного підходу

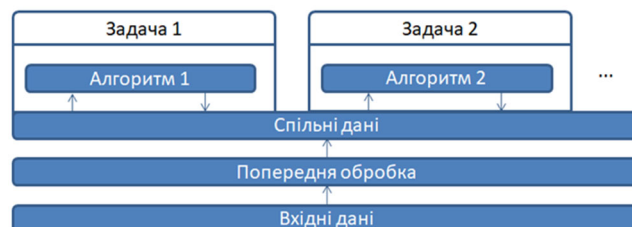


Рис. 4. Схематична архітектура підходу МЄАС

Вибір структури даних для розв'язання розглянутого комплексу задач за допомогою МЄАС, що забезпечить найбільший приріст ефективності (з погляду часу виконання та витрат пам'яті), вимагає проведення подальших досліджень, однак, зважаючи на велику частоту використання задачі пошуку найближчого сусіда та споріднених задач (пошук кількох найближчих сусідів, пошук найближчого сусіда в заданому напрямку тощо), перспективним є дослідження МЄАС на базі таких структур даних, як *kd*-дерево, дерево октантів та *BVH*-дерево. Також актуальним напрямком досліджень є узагальнення алгоритмів генерування мешів, яке б дозволяло отримати вихідний результат у форматі, на якому базується МЄАС, тим самим позбуваючись необхідності виконувати попередню обробку та побудову структур даних для мешів, що були згенеровані.

Дискусія і висновки

Розвиток технології адитивного виробництва та поширення сфер її застосування на такі галузі, як автомобілебудування, аерокосмічну промисловість та медицину, обумовлює зростання складності як самих моделей, що друкуються, так і процеси планування друку. У пропонуваній роботі було розглянуто ключові задачі, що постають на етапі планування адитивного виробництва, та наведено наявні підходи до їхнього розв'язання. У розд. 2 було наведено огляд структур даних, які має реалізовувати та підтримувати програмний комплекс для планування адитивного виробництва, що базується на пакетно-алгоритмічному підході, та акцентовано увагу на неефективності такого підходу, ураховуючи необхідність підтримки актуального стану декількох структур даних одночасно, або їх побудови перед викликом чергового алгоритму. Розділ 3 присвячено перспективним напрямкам досліджень у розглянутій галузі, які допоможуть розв'язати актуальні проблеми, окреслені в розд. 2. Ключем до створення ефективної програмної системи розв'язку комплексу задач планування адитивного виробництва визначено побудову МЄАС на основі уніфікованої структури даних для всіх алгоритмів. Також актуальним напрямком досліджень є алгоритми генерування мешів, що будують результуючий меш у формі, сумісній з МЄАС без додаткового конвертування.

Внесок авторів: Василь Терещенко – концептуалізація; методологія; написання (перегляд і редагування); Максим Осіпюнок – аналіз джерел; формальний аналіз; написання (оригінальна чернетка).

Список використаних джерел

- Терещенко, В., Осадчий, Б., Петльована, М., Селезньов, А., & Тара, А. (2015). Архітектура моделі єдиного алгоритмічного середовища (МЄАС) для розв'язання задач обчислювальної геометрії. *Вісник Київського національного університету імені Тараса Шевченка. Серія: Фізико-математичні науки*, 2, 183–188. http://nbuv.gov.ua/UJRN/VKNU_fiz_mat_2015_2_38
- Akhouri, D., Karmakar, D., Banerjee, D., & Mishra, S. B. (2021). Various Infill Patterns and their Effect in 3D Printable Materials. *International Journal of Innovative Science and Research Technology*, 6(9), 538–542.
- Attene, M., Campen, M., & Kobbelt, L. (2013). Polygon Mesh Repairing: An Application Perspective. *ACM Computing Surveys*, 45(2), 1–33. <https://doi.org/10.1145/2431211.2431214>
- Boris. (2018, April 15). *Dual contouring tutorial*. BorisTheBrave.Com. <https://www.boristhebrave.com/2018/04/15/dual-contouring-tutorial>
- Chernov, N., Stoyan, Yu., & Romanova, T. (2010). Mathematical model and efficient algorithms for object packing problem. *Computational Geometry*, 43(5), 535–553. <https://doi.org/10.1016/j.comgeo.2009.12.003>
- Chiu, B. W. (2020). *Additive manufacturing applications and implementation in Aerospace* (thesis). Massachusetts Institute of Technology.
- Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2), 182–197. <https://doi.org/10.1109/4235.996017>
- Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1), 269–271. <https://doi.org/10.1007/bf01386390>
- Ho, Chien.-Chang, Wu, F., Chen, B., Chuang, Y., & Ouhyoung, M. (2005). Cubical marching squares: Adaptive feature preserving surface extraction from volume data. *Computer Graphics Forum*, 24(3), 537–545. <https://doi.org/10.1111/j.1467-8659.2005.00879.x>
- Huang, S. H., Liu, P., Mokasdar, A., & Hou, L. (2012). Additive Manufacturing and its societal impact: A literature review. *The International Journal of Advanced Manufacturing Technology*, 67(5–8), 1191–1203. <https://doi.org/10.1007/s00170-012-4558-5>
- International Organization for Standardization. (2016). *Industrial automation systems and integration — Product data representation and exchange. Part 21: Implementation methods: Clear text encoding of the exchange structure* (ISO Standard No. 10303-21:2016). <https://www.iso.org/standard/63141.html>
- Jiang, X., Peng, Q., Cheng, X., Dai, N., Cheng, C., & Li, D. (2016). Efficient booleans algorithms for triangulated meshes of geometric modeling. *Computer-Aided Design and Applications*, 13(4), 419–430. <https://doi.org/10.1080/16864360.2015.1131530>
- Kim, D.-S., Kim, D., Cho, Y., & Sugihara, K. (2006). Quasi-triangulation and interworld data structure in three dimensions. *Computer-Aided Design*, 38(7), 808–819. <https://doi.org/10.1016/j.cad.2006.04.008>
- Knuth, D. E. (1998). *The Art of Computer Programming, volume 3: Sorting and searching. Second edition*. Addison Wesley.
- Kobbelt, L. P., Botsch, M., Schwanecke, U., & Seidel, H.-P. (2001). Feature sensitive surface extraction from volume data. *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*. <https://doi.org/10.1145/383259.383265>
- Kulkarni, P., Marsan, A., & Dutta, D. (2000). A review of process planning techniques in layered manufacturing. *Rapid Prototyping Journal*, 6(1), 18–35. <https://doi.org/10.1108/13552540010309859>

- Kwok, T.-H., Ye, H., Chen, Y., Zhou, C., & Xu, W. (2017). Mass customization: Reuse of digital slicing for additive manufacturing. *Journal of Computing and Information Science in Engineering*, 17(2). <https://doi.org/10.1115/1.4034010>
- Lan, P.-T., Chou, S.-Y., Chen, L.-L., & Gemmill, D. (1997). Determining fabrication orientations for rapid prototyping with stereolithography apparatus. *Computer-Aided Design*, 29(1), 53–62. [https://doi.org/10.1016/s0010-4485\(96\)00049-8](https://doi.org/10.1016/s0010-4485(96)00049-8)
- Laue, R., Colditz, P., Möckel, M., & Awiszus, B. (2022). Study on the milling of additive manufactured components. *Metals*, 12(7), 1167. <https://doi.org/10.3390/met12071167>
- Lorensen, W. E., & Cline, H. E. (1987). Marching cubes: A high resolution 3D surface construction algorithm. *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*. <https://doi.org/10.1145/37401.37422>
- Lutters, E., ten Dam, D., & Faneker, T. (2012). 3D nesting of complex shapes. *Procedia CIRP*, 3, 26–31. <https://doi.org/10.1016/j.procir.2012.07.006>
- Marching cubes. (2023, November 15). In *Wikipedia*. https://en.wikipedia.org/wiki/Marching_cubes
- Meagher, D. (1980) *Octree Encoding: A New Technique for the Representation, Manipulation and Display of Arbitrary 3-D Objects by Computer*, Technical report. Rensselaer, N. Y.
- Minetto, R., Volpato, N., Stoffi, J., Gregori, R. M. M. H., & da Silva, M. V. G. (2017). An optimal algorithm for 3D triangle mesh slicing. *Computer-Aided Design*, 92, 1–10. <https://doi.org/10.1016/j.cad.2017.07.001>
- Pankratov, A., Romanova, T., & Litvinchev, I. (2020). Packing oblique 3D objects. *Mathematics*, 8(7), 1130. <https://doi.org/10.3390/math8071130>
- Rosato, A., Strandburg, K. J., Prinz, F., & Swendsen, R. H. (1987). Why the Brazil nuts are on top: Size segregation of particulate matter by shaking. *Physical Review Letters*, 58(10), 1038–1040. <https://doi.org/10.1103/physrevlett.58.1038>
- Rozenberg, G., & Salomaa, A. (1980). *The mathematical theory of L Systems*. Acad. Press.
- Salmi, M. (2021). Additive manufacturing processes in medical applications. *Materials*, 14(1), 191. <https://doi.org/10.3390/ma14010191>
- Schaefer, S., Ju, T., & Warren, J. (2007). Manifold dual contouring. *IEEE Transactions on Visualization and Computer Graphics*, 13(3), 610–619. <https://doi.org/10.1109/tvcg.2007.1012>
- Scherzinger, A., Brix, T., & Hinrichs, K. (2017). An efficient geometric algorithm for clipping and capping solid triangle meshes. *Proceedings of the 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*. <https://doi.org/10.5220/0006097201870194>
- Simplify3D Software. (2019, April 23). *Multi-part printing*. Simplify3D Software. <https://www.simplify3d.com/resources/articles/multi-part-printing>
- Stoyan, Y., Pankratov, A., Romanova, T., Fasano, G., Pintér, J. D., Stoian, Y. E., & Chugay, A. (2019). Optimized Packings in space engineering applications: Part I. *Springer Optimization and Its Applications*, 395–437. https://doi.org/10.1007/978-3-030-10501-3_15
- Tereshchenko, V. N., & Anisimov, A. V. (2010). Recursion and parallel algorithms in geometric modeling problems. *Cybernetics and Systems Analysis*, 46(2), 173–184. <https://doi.org/10.1007/s10559-010-9196-z>
- Tereshchenko, V., Budjak, I., & Fisunen, A. (2013). The unified algorithmic platform for solving complex problems of Computational Geometry. *Lecture Notes in Computer Science*, 424–428. https://doi.org/10.1007/978-3-642-39958-9_39
- Vaidya, R., & Anand, S. (2016). Optimum support structure generation for additive manufacturing using unit cell structures and support removal constraint. *Procedia Manufacturing*, 5, 1043–1059. <https://doi.org/10.1016/j.promfg.2016.08.072>
- Valdivieso, C. (2021, September 9). *The role of AM in the automotive industry*. 3Dnatives. <https://www.3dnatives.com/en/the-role-of-am-in-the-automotive-industry/#!>
- Volpato, N., Franzoni, A., Luvizon, D. C., & Schramm, J. M. (2013). Identifying the directions of a set of 2D contours for additive manufacturing process planning. *The International Journal of Advanced Manufacturing Technology*, 68(1–4), 33–43. <https://doi.org/10.1007/s00170-012-4706-y>
- Wald, I., & Havran, V. (2006). On building fast KD-trees for Ray Tracing, and on doing that in $O(N \log N)$. *2006 IEEE Symposium on Interactive Ray Tracing*. <https://doi.org/10.1109/irt.2006.280216>
- Wohlert, T., & Gornet, T. (2014). History of Additive Manufacturing. *Wohlert Report*, 1–34.
- Yang, S. W., & Choi, Y. (2010). Triangulation of CAD data for visualization using a compact array-based triangle data structure. *Computers & Graphics*, 34(4), 424–429. <https://doi.org/10.1016/j.cag.2010.02.001>
- Yu, F., Cao, J., Shan, J., Lo, S. H., & Guan, Z. (2021). PASM: Parallel aligned surface meshing. *International Journal for Numerical Methods in Engineering*, 122(15), 3705–3732. <https://doi.org/10.1002/nme.6678>
- Zhang, Y., Wang, Z., Zhang, Y., Gomes, S., & Bernard, A. (2020). Bio-inspired generative design for Support Structure Generation and optimization in Additive Manufacturing (AM). *CIRP Annals*, 69(1), 117–120. <https://doi.org/10.1016/j.cirp.2020.04.091>

References

- Akhouri, D., Karmakar, D., Banerjee, D., & Mishra, S. B. (2021). Various Infill Patterns and their Effect in 3D Printable Materials. *International Journal of Innovative Science and Research Technology*, 6(9), 538–542.
- Attene, M., Campen, M., & Kobbelt, L. (2013). Polygon Mesh Repairing: An Application Perspective. *ACM Computing Surveys*, 45(2), 1–33. <https://doi.org/10.1145/2431211.2431214>
- Boris. (2018, April 15). *Dual contouring tutorial*. BorisTheBrave.Com. <https://www.boristhebrave.com/2018/04/15/dual-contouring-tutorial>
- Chernov, N., Stoyan, Yu., & Romanova, T. (2010). Mathematical model and efficient algorithms for object packing problem. *Computational Geometry*, 43(5), 535–553. <https://doi.org/10.1016/j.comgeo.2009.12.003>
- Chiu, B. W. (2020). *Additive manufacturing applications and implementation in Aerospace* (thesis). Massachusetts Institute of Technology.
- Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2), 182–197. <https://doi.org/10.1109/4235.996017>
- Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1), 269–271. <https://doi.org/10.1007/bf01386390>
- Ho, Chien-Chang, Wu, F., Chen, B., Chuang, Y., & Ouhyoung, M. (2005). Cubical marching squares: Adaptive feature preserving surface extraction from volume data. *Computer Graphics Forum*, 24(3), 537–545. <https://doi.org/10.1111/j.1467-8659.2005.00879.x>
- Huang, S. H., Liu, P., Mokasdar, A., & Hou, L. (2012). Additive Manufacturing and its societal impact: A literature review. *The International Journal of Advanced Manufacturing Technology*, 67(5–8), 1191–1203. <https://doi.org/10.1007/s00170-012-4558-5>
- International Organization for Standardization. (2016). *Industrial automation systems and integration – Product data representation and exchange. Part 21: Implementation methods: Clear text encoding of the exchange structure* (ISO Standard No. 10303-21:2016). <https://www.iso.org/standard/63141.html>
- Jiang, X., Peng, Q., Cheng, X., Dai, N., Cheng, C., & Li, D. (2016). Efficient booleans algorithms for triangulated meshes of geometric modeling. *Computer-Aided Design and Applications*, 13(4), 419–430. <https://doi.org/10.1080/16864360.2015.1131530>
- Kim, D.-S., Kim, D., Cho, Y., & Sugihara, K. (2006). Quasi-triangulation and interworld data structure in three dimensions. *Computer-Aided Design*, 38(7), 808–819. <https://doi.org/10.1016/j.cad.2006.04.008>
- Knuth, D. E. (1998). *The Art of Computer Programming, volume 3: Sorting and searching. Second edition*. Addison Wesley.
- Kobbelt, L. P., Botsch, M., Schwanecke, U., & Seidel, H.-P. (2001). Feature sensitive surface extraction from volume data. *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*. <https://doi.org/10.1145/383259.383265>
- Kulkarni, P., Marsan, A., & Dutta, D. (2000). A review of process planning techniques in layered manufacturing. *Rapid Prototyping Journal*, 6(1), 18–35. <https://doi.org/10.1108/13552540010309859>
- Kwok, T.-H., Ye, H., Chen, Y., Zhou, C., & Xu, W. (2017). Mass customization: Reuse of digital slicing for additive manufacturing. *Journal of Computing and Information Science in Engineering*, 17(2). <https://doi.org/10.1115/1.4034010>
- Lan, P.-T., Chou, S.-Y., Chen, L.-L., & Gemmill, D. (1997). Determining fabrication orientations for rapid prototyping with stereolithography apparatus. *Computer-Aided Design*, 29(1), 53–62. [https://doi.org/10.1016/s0010-4485\(96\)00049-8](https://doi.org/10.1016/s0010-4485(96)00049-8)
- Laue, R., Colditz, P., Möckel, M., & Awiszus, B. (2022). Study on the milling of additive manufactured components. *Metals*, 12(7), 1167. <https://doi.org/10.3390/met12071167>
- Lorensen, W. E., & Cline, H. E. (1987). Marching cubes: A high resolution 3D surface construction algorithm. *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*. <https://doi.org/10.1145/37401.37422>
- Lutters, E., ten Dam, D., & Faneker, T. (2012). 3D nesting of complex shapes. *Procedia CIRP*, 3, 26–31. <https://doi.org/10.1016/j.procir.2012.07.006>
- Marching cubes. (2023, November 15). In *Wikipedia*. https://en.wikipedia.org/wiki/Marching_cubes
- Meagher, D. (1980) *Octree Encoding: A New Technique for the Representation, Manipulation and Display of Arbitrary 3-D Objects by Computer*, Technical report. Rensselaer, N. Y.

- Minetto, R., Volpato, N., Stolfi, J., Gregori, R. M. M. H., & da Silva, M. V. G. (2017). An optimal algorithm for 3D triangle mesh slicing. *Computer-Aided Design*, 92, 1–10. <https://doi.org/10.1016/j.cad.2017.07.001>
- Pankratov, A., Romanova, T., & Litvinchev, I. (2020). Packing oblique 3D objects. *Mathematics*, 8(7), 1130. <https://doi.org/10.3390/math8071130>
- Rosato, A., Strandburg, K. J., Prinz, F., & Swendsen, R. H. (1987). Why the Brazil nuts are on top: Size segregation of particulate matter by shaking. *Physical Review Letters*, 58(10), 1038–1040. <https://doi.org/10.1103/physrevlett.58.1038>
- Rozenberg, G., & Salomaa, A. (1980). *The mathematical theory of L Systems*. Acad. Press.
- Salmi, M. (2021). Additive manufacturing processes in medical applications. *Materials*, 14(1), 191. <https://doi.org/10.3390/ma14010191>
- Schaefer, S., Ju, T., & Warren, J. (2007). Manifold dual contouring. *IEEE Transactions on Visualization and Computer Graphics*, 13(3), 610–619. <https://doi.org/10.1109/tvcg.2007.1012>
- Scherzinger, A., Brix, T., & H. Hinrichs, K. (2017). An efficient geometric algorithm for clipping and capping solid triangle meshes. *Proceedings of the 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*. <https://doi.org/10.5220/0006097201870194>
- Simplify3D Software. (2019, April 23). *Multi-part printing*. Simplify3D Software. <https://www.simplify3d.com/resources/articles/multi-part-printing/>
- Stoyan, Y., Pankratov, A., Romanova, T., Fasano, G., Pintér, J. D., Stoian, Y. E., & Chugay, A. (2019). Optimized Packings in space engineering applications: Part I. *Springer Optimization and Its Applications*, 395–437. https://doi.org/10.1007/978-3-030-10501-3_15
- Tereshchenko, V. N., & Anisimov, A. V. (2010). Recursion and parallel algorithms in geometric modeling problems. *Cybernetics and Systems Analysis*, 46(2), 173–184. <https://doi.org/10.1007/s10559-010-9196-z>
- Tereshchenko, V., Budjak, I., & Fisunen, A. (2013). The unified algorithmic platform for solving complex problems of Computational Geometry. *Lecture Notes in Computer Science*, 424–428. https://doi.org/10.1007/978-3-642-39958-9_39
- Tereshchenko, V., Osadchii, B., Petlyovana, M., Seleznyov, A., Tara, A. (2015). Architecture of model of the united algorithmic environment (MUAЕ) for solving computational geometry problems. *Bulletin of Taras Shevchenko National University of Kyiv. Series: Physics and Mathematics*, 2, 183–188 [in Ukrainian].
- Vaidya, R., & Anand, S. (2016). Optimum support structure generation for additive manufacturing using unit cell structures and support removal constraint. *Procedia Manufacturing*, 5, 1043–1059. <https://doi.org/10.1016/j.promfg.2016.08.072>
- Valdivieso, C. (2021, September 9). *The role of AM in the automotive industry*. 3Dnatives. <https://www.3dnatives.com/en/the-role-of-am-in-the-automotive-industry/#/>
- Volpato, N., Franzoni, A., Luvizon, D. C., & Schramm, J. M. (2013). Identifying the directions of a set of 2D contours for additive manufacturing process planning. *The International Journal of Advanced Manufacturing Technology*, 68(1–4), 33–43. <https://doi.org/10.1007/s00170-012-4706-y>
- Wald, I., & Havran, V. (2006). On building fast KD-trees for Ray Tracing, and on doing that in $O(N \log N)$. *2006 IEEE Symposium on Interactive Ray Tracing*. <https://doi.org/10.1109/irt.2006.280216>
- Wohlert, T., & Gomet, T. (2014). History of Additive Manufacturing. *Wohlert Report*, 1–34.
- Yang, S. W., & Choi, Y. (2010). Triangulation of CAD data for visualization using a compact array-based triangle data structure. *Computers & Graphics*, 34(4), 424–429. <https://doi.org/10.1016/j.cag.2010.02.001>
- Yu, F., Cao, J., Shan, J., Lo, S. H., & Guan, Z. (2021). PASM: Parallel aligned surface meshing. *International Journal for Numerical Methods in Engineering*, 122(15), 3705–3732. <https://doi.org/10.1002/nme.6678>
- Zhang, Y., Wang, Z., Zhang, Y., Gomes, S., & Bernard, A. (2020). Bio-inspired generative design for Support Structure Generation and optimization in Additive Manufacturing (AM). *CIRP Annals*, 69(1), 117–120. <https://doi.org/10.1016/j.cirp.2020.04.091>
- Zhang, Y., Wang, Z., Zhang, Y., Gomes, S., & Bernard, A. (2020). Bio-inspired generative design for Support Structure Generation and optimization in Additive Manufacturing (AM). *CIRP Annals*, 69(1), 117–120. <https://doi.org/10.1016/j.cirp.2020.04.091>

Отримано редакцією журналу / Received: 23.02.24
Прорецензовано / Revised: 26.04.24
Схвалено до друку / Accepted: 27.05.24

Vasyl TERESHCHENKO, DSc (Phys. & Math.), Prof.
ORCID ID: 0000-0002-0139-6049
e-mail: vtereshch@gmail.com
Taras Shevchenko National University of Kyiv, Kyiv, Ukraine

Maksym OSIPONOK, PHD Student
ORCID ID: 0000-0002-3372-3032
e-mail: osipyonok@ukr.net
Taras Shevchenko National University of Kyiv, Kyiv, Ukraine

PROCESS PLANNING IN ADDITIVE MANUFACTURING: A REVIEW OF PROBLEMS AND METHODS OF THEIR SOLUTION

The paper provides an in-depth analysis of the modern techniques and strategies for process planning in additive manufacturing (AM). It delves into the key stages of the AM process planning, examines algorithmic challenges and tasks that emerge while preparing models for 3D printing, which are applicable to a wide range of existing AM technologies. These tasks include the conversion of CAD data to meshes, mesh generation, Boolean operations on meshes, support structure generation, packing problems, and slicing. Each task is examined in a dedicated subsection, offering a problem statement, rationale for its significance in the AM process planning, and key approaches to resolving it. Based on the results of a review of existing approaches to solving these problems, the paper analyzes the necessary data structures for an integrated software system designed for AM process planning. It highlights the drawbacks of the conventional batch-algorithmic approach and introduces an alternative solution based on the model of a unified algorithmic environment (MUAЕ). MUAЕ considers task interdependence to create software systems that are both efficient and extendable. The examination of the optimal data structure for MUAЕ is recognized as a promising direction for future research in the realm of additive manufacturing process planning.

To conduct an analysis of modern AM process planning, the following methods were applied: literature review and algorithm analysis, comparative examination of data structures supporting the reviewed algorithms, critical review of existing methodologies, and the development of an alternative approach (MUAЕ).

The analysis has shown that the traditional batch-algorithmic approach to constructing an integrated system for AM process planning is insufficient for effectively addressing all necessary tasks.

In response to the inefficiencies revealed in the batch-algorithmic approach, the paper proposes the application of an alternative approach based on the model of a unified algorithmic environment (MUAЕ).

Keywords: 3D printing, process planning in additive manufacturing, mesh generation, support structures generation, packing problem, slicing, model of the unified algorithmic environment.

Автори заявляють про відсутність конфлікту інтересів. Спонсори не брали участі в розробленні дослідження; у зборі, аналізі чи інтерпретації даних; у написанні рукопису; в рішенні про публікацію результатів.

The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses or interpretation of data; in the writing of the manuscript; in the decision to publish the results.