

Національна академія наук України
Інститут проблем машинобудування ім. А.М. Підгорного

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Кваліфікаційна наукова
праця на правах рукопису

СТОЯНЮРІЙ ЄВГЕНОВИЧ

УДК 519.859

ДИСЕРТАЦІЯ

**МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ ТА МЕТОДИ РОЗВ'ЯЗАННЯ
ОПТИМІЗАЦІЙНИХ ЗАДАЧ УПАКОВКИ ДОВІЛЬНИХ
БАГАТОГРАННИКІВ**

01.05.02 – математичне моделювання та обчислювальні методи,
технічні науки

Подається на здобуття наукового ступеня кандидата наук

Дисертація містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

Підпис Ю.Є. Стоян

Науковий керівник:

Романова Тетяна Євгенівна, доктор технічних наук, професор

Цей примірник дисертаційної роботи
ідентичний за змістом з іншими, поданими
до спеціалізованої вченої ради Д 64.052.02

Вчений секретар спецради Д64.052.02

Підпис Л.В. Колесник
Печатка

Харків – 2019

АНОТАЦІЯ

Стоян Ю.Є. Математичне моделювання та методи розв'язання оптимізаційних задач упаковки довільних багатогранників. – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття наукового ступеня кандидата технічних наук за спеціальністю 01.05.02 – «Математичне моделювання та обчислювальні методи». – Інститут проблем машинобудування ім. А.М. Підгорного НАН України, Харківський національний університет радіоелектроніки МОН України, Харків, 2018.

На сучасному етапі інтерес до ефективного розв'язання задач упаковки тривимірних геометричних об'єктів (3D об'єктів) стрімко зростає. Це пояснюється надзвичайною складністю методів їх розв'язання та широким спектром практичних застосувань. У зв'язку з цим наукового та практичного значення набуває проблема створення інтелектуальних інформаційних технологій розв'язання оптимізаційних задач упаковки 3D об'єктів на основі конструктивних засобів математичного та комп'ютерного моделювання, що легко адаптуються для конкретних технологічних процесів. Зазначимо, що 3D об'єкти довільної просторової форми завжди можуть бути апроксимовані багатогранниками з наперед заданою точністю.

Метою роботи є підвищення ефективності розв'язання оптимізаційних задач оптимальної упаковки довільних багатогранників шляхом розробки конструктивних засобів математичного і комп'ютерного моделювання, нових математичних моделей та ефективних оптимізаційних алгоритмів із застосуванням методів нелінійного програмування і сучасних NLP-солверів (програме забезпечення, що використовується для пошуку локальних та глобальних розв'язків задач нелінійного програмування).

Оптимізаційні задачі упаковки довільних багатогранників виникають під час дослідження актуальних проблем матеріалознавства, медицини, адитивного виробництва (3D друку), космічної інженерії, логістики, робототехніки, машинобудування та інших пріоритетних галузей науки та техніки.

Останні п'ять років у зарубіжній літературі з'являються публікації, в яких розглядаються практичні задачі, що зводяться до оптимізаційних задач упаковки багатогранників. Оскільки задачі є NP-складними, методології розв'язання зазвичай використовують евристичні методи.

Основна проблема полягає в складності формалізації обмежень розміщення довільних багатогранників (неперервні обертання багатогранників, включення багатогранників у контейнер, що має довільну просторову форму, мінімально допустимі відстані між багатогранниками).

Як відомо, найбільш потужним засобом математичного моделювання відношень геометричних об'єктів є метод ρ -функцій. На цей час побудовані ρ -функції та квазі- ρ -функції для деяких форм 3D об'єктів (опуклі тривимірні тіла, поверхня яких сформована циліндричними, конічними та сферичними поверхнями).

Однак, побудувати засоби математичного моделювання для аналітичного опису найважливіших з практичної точки зору обмежень на мінімально допустимі відстані між довільними (опуклими та неопуклими) багатогранниками, що допускають неперервні повороти, не вдалося.

Слід зазначити, що існуючі математичні моделі оптимізаційних задач розміщення багатогранників у вигляді задач нелінійного програмування не враховують одночасно неперервні обертання, обмеження на мінімально допустимі відстані між об'єктами та обмеження балансу (мінімальне відхилення системи від заданої точки).

Відкритим питанням також є оптимізація упаковки багатогранників у різноманітних видах контейнерів, оскільки в більшості публікацій розглядаються тільки кубоїдні контейнери.

Тому актуальною є проблема створення ефективних методів розв'язання наукових і практичних оптимізаційних 3D задач упаковки N довільних багатогранників, що допускають неперервні обертання і трансляції, з урахуванням мінімально допустимих відстаней та обмежень балансу, в довільний опуклий контейнер, границя якого формується циліндричною, еліптичною, сферичною поверхнями та площинами, мінімального об'єму (*OPP, Optimal Polytope Packing*), що вимагає побудови нових засобів математичного моделювання, точних математичних моделей та ефективних методів розв'язання.

Через це в роботі вперше побудовані: *нормалізовані та псевдонормалізовані квазі ρ -функції* з метою аналітичного опису обмежень на мінімально допустимі відстані між довільними багатогранниками; *нормалізовані та псевдонормалізовані ρ -функції* з метою аналітичного опису обмежень на мінімально допустимі відстані між довільним багатогранником та границею довільного опуклого контейнера. Побудовані вільні від радикалів псевдонормалізовані квазі ρ -функції та псевдонормалізовані ρ -функції дозволяють описати обмеження розміщення в задачі OPP у вигляді системи нерівностей з гладкими функціями. *Обмеження балансу* для задачі OPP формалізовані у вигляді системи нерівностей з *гладкими функціями*.

Вперше побудована математична модель задачі OPP у вигляді неперервної задачі нелінійного програмування (що включає всі її глобально оптимальні розв'язки) для неопуклих багатогранників в довільному опуклому контейнері, границя якого формується за допомогою сферичних, циліндричних, еліптичних поверхонь та площини з урахуванням обмежень розміщення та обмежень балансу, що

дозволяє використовувати сучасні *NLP-солвери*. Як функції цілі розглядаються об'єм, метрична характеристика або коефіцієнт гомотетії контейнера; область допустимих розв'язків математичної моделі задачі *OPP* описується системою нерівностей з гладкими *функціями*. Як системи додаткових обмежень можуть розглядатися обмеження на метричні характеристики контейнера та параметри розміщення багатогранників.

Вперше побудована математична модель задачі кластерінгу неопуклих багатогранників (*OPC*) в сферичній, кубоїдній та циліндричній охоплюючих областях мінімального об'єму, що дозволяє генерувати ефективні допустимі стартові точки для пошуку локальних екстремумів задачі *OPP*.

Набули подальшого розвитку методи розв'язання задач геометричного проектування: запропоновано стратегію та розроблено ефективні методи розв'язання задачі *OPP* та її основних реалізацій з використанням сучасних *NLP-солверів*, які (на відміну від існуючих евристичних підходів): дозволяють розміщувати 3D об'єкти довільної просторової форми, апроксимовані багатогранниками з наперед заданою точністю; враховують одночасно *неперервні* трансляції та обертання багатогранників, *мінімально допустимі відстані* і *обмеження балансу*; дозволяють отримувати локально оптимальні розв'язки для задач *OPP*, кращі за значенням цільової функції (порівняно з *benchmark instances* – відомими опублікованими результатами).

Розроблено метод побудови допустимих стартових точок, що ґрунтується на гомотетичних перетвореннях багатогранників, та метод декомпозиції POLYDEC для пошуку локальних екстремумів, що дозволяє звести задачу *OPP* розмірності $O(n^2)$ з числом нерівностей $O(n^2)$ до послідовності підзадач нелінійного програмування з числом нерівностей $O(n)$ і з числом змінних $O(n)$, де n – кількість опуклих

багатогранників, що формують неопуклі багатогранники. В основі цього методу лежить ітераційна процедура, ідея якої полягає в такому: для допустимої стартової точки задачі ОРР для кожного опуклого багатогранника, що формує неопуклий багатогранник, будується система шести лінійних ϵ -нерівностей, яка гарантує належність опуклого багатогранника індивідуальному кубічному контейнеру; на кожній ітерації в область допустимих розв'язків підзадачі нелінійного програмування включається система ϵ -нерівностей для всіх опуклих багатогранників та псевдонормалізовані квазі ϕ -функції тільки для тих пар опуклих багатогранників, що є « ϵ -сусідами», і псевдонормалізовані ϕ -функції для тих опуклих багатогранників, чиї індивідуальні контейнери перетинаються з границею контейнера. Точка локального мінімуму підзадачі на останній ітерації оптимізаційної процедури формує точку локального мінімуму задачі ОРР.

Грунтуючись на запропонованих в роботі засобах математичного моделювання, математичних моделях, методах і алгоритмах, розроблено програмне забезпечення для розв'язання основних реалізацій задачі ОРР та програмний модуль для розв'язання задачі ОРС. Проведено відповідні обчислювальні експерименти, отримано локально оптимальні розв'язки для оптимізації упаковки опуклих та неопуклих багатогранників в шарі, циліндрі, кубоїді, еліпсоїді, опуклому багатограннику та в опуклому контейнері, границя якого формується за допомогою зазначених вище поверхонь, з урахуванням обмежень балансу та обмежень на мінімально допустимі відстані. Наведено результати порівняльного аналізу з найкращими світовими результатами.

Отримано довідку про використання програмного модуля оптимізації упаковки довільних неорієнтованих багатогранників, який застосовується для розв'язання задачі оптимального заповнення заданого об'єму частинками несферичної форми, під час проведення

досліджень на кафедрі прикладного матеріалознавства та обробки матеріалів Національного університету «Львівська політехніка», щодо можливості застосування вітчизняних порошків титанових сплавів несферичної форми, які є на порядок дешевшими та можуть бути застосовані в адитивних технологіях.

Отримано листа підтримки від Дж. Фазано (Giorgio Fasano) – провідного вченого та спеціаліста в галузі математичного моделювання та оптимізації систем (systems modeling and optimization) європейської компанії «Thales Alenia Space» (<https://www.thalesgroup.com/en>), в якому зазначено важливість отриманих в роботі результатів в області ракетно-космічного машинобудування, зокрема для оптимізації компоновки обладнання космічних апаратів.

Отримано акт про впровадження запропонованих в роботі засобів, моделей, методів та програмних модулів в ІТ компанії «Cloud Works» для розв'язання: задач оптимізації процесу 3D-друку в адитивному виробництві, що використовує SLS технологію; задач оптимального пакування вантажів у довільних контейнерах у сфері логістики.

Запропоновані засоби математичного моделювання задач *OPP* впроваджено в навчальний процес у Харківському національному університеті радіоелектроніки в курсах: «Математичне та комп'ютерне моделювання в системах підтримки прийняття рішень», «Моделювання геометричних об'єктів».

Достовірність отриманих у роботі результатів підтверджується аргументованим викладенням матеріалу, доведеними твердженнями, використанням NLP-solvers, проведенням порівняння результатів обчислювальних експериментів за значенням функції цілі та часу розв'язання з існуючими найкращими світовими аналогами.

Ключові слова: упаковка, опуклі контейнери, неорієнтовані багатогранники, допустимі відстані, обмеження балансу, квазі phi-

функції, математичне моделювання, нелінійна оптимізація, метод декомпозиції.

Список публікацій здобувача

1. Stoian Y. E., Chugay A. M., Pankratov A. V., Romanova T. E. Two Approaches to Modeling and Solving the Packing Problem for Convex Polytopes // Cybernetics and Systems Analysis. 2018. № 4 (54). P. 585-593.
2. Стоян Ю. Е., Панкратов А.В., Романова Т.Е. Упаковка многогранников в выпуклый контейнер минимального объема // Системи управління, навігації та зв'язку. 2018. № 2 (48). С. 48-54
3. Chugay A., Stoian Ye. Cluster packing of concave non-oriented polyhedra in a cuboid // Advanced Information Systems. 2018. № 2(1). P.16-21
4. Stoian Y., Pankratov A., Romanova T. Optimal clustering of polyhedra // Bionics Intelligence. 2017. № 2 (89). P. 12-22
5. Чугай А.М., Панкратов О.В., Романова Т.Є., Стоян Ю.Є. Оптимізація процесу 3D-друку для SLS технології адитивного виробництва // Системи управління, навігації та зв'язку. 2017. № 6 (46). С. 127-130
6. Стоян Ю.Е., Романова Т.Е., Панкратов А.В. Математическая модель задачи оптимальной компоновки многогранников в выпуклой многогранной области // Системи управління, навігації та зв'язку. 2016. № 3 (39). С. 62-66.
7. Панкратов А.В., Романова Т.Е., Стоян Ю.Е., Чугай А.М. Задача оптимизации упаковки многогранников в сферическом и цилиндрическом контейнерах // Восточно-Европейский журнал передовых технологий. 2016. № 1/4 (79). С. 39-47.
8. Панкратов А.В., Романова Т.Е., Стоян Ю.Е. Псевдонормализованные квази ϕ -функции для многогранников // Радиоэлектроника и информатика. 2015. № 3. С. 22-26.

9. Стоян Ю.Е. Упаковка многогранников в выпуклый контейнер минимального объема. Современные проблемы машиностроения: тез. докл. конф. молодых ученых и специалистов, Харьков: Ин-т. проблем машиностроения им. А. М. Подгорного НАН Украины. 2018. С. 18.

10. Romanova T., Bennell J.A., Stoyan Y., Pankratov A. Packing of arbitrary polyhedra with continuous rotations using nonlinear optimisation. ESICUP meeting: тез. докл. 15-ой международной конференции, Zoetermeer, Netherlands. 2018. P. 17-18

11. Stoian Y.E., Romanova T.E., Pankratov A.V. Packing of polyhedra within a convex container. Problem of Decision Making under Uncertainties: тез. докл. XXX международной конференции, Вильнюс, Литва. 2017. С. 116

12. Стоян Ю.Е. Математическое моделирование оптимизации упаковки многогранников в выпуклой многогранной области. Радиоэлектроника и молодёжь в XXI веке: тез. докл. XXI Международного молодёжного форума, Харьков: ХНУРЭ. 2017. С. 104-105.

13. Romanova T.E., Pankratov A.V., Stoian Y.E. Packing of Polytopes within Spherical and Cylindrical Containers. ESICUP meeting: тез. докл. 14-ой международной конференции, Льеж, Бельгия. 2017. С. 23.

14. Chugay A., Pankratov A., Romanova T. Stoian Yu. Optimization of the process of 3D-Printing for SLS technologies of additive production. Application of Information and Communication Technology and Statistics in Economy and Education: тез. докл. 7-ой международной конференции. – София, Болгария. 2017. FA24.4p.

15. Stoian Yu. Packing polyhedrons using quasi-phi-functions. Современные проблемы машиностроения: тез. докл. конф. молодых

ученых и специалистов, Харьков: Институт проблем машиностроения им. А.М. Подгорного НАН Украины. 2016. С. 33.

16. Stoian Yu. Mathematical model and solution algorithm for packing problem of convex polyhedra. Information Technologies and Computer Modelling: тез. докл. международной научно-практической конф., Ивано-Франковск – Яремча. 2016. С. 187-188

17. Стоян Ю.Е., Панкратов А.В., Романова Т.Е., Чугай А.М. О NLP-задаче оптимальной упаковки невыпуклых многогранников в сферическом и цилиндрическом контейнерах. Математическое моделирование, оптимизация и информационные технологии: тез. докл. 5-ой Межд. науч. конф., Кишинев, Молдавия. 2016. С. 345-348

18. Стоян Ю.Е., Романова Т.Е., Панкратов А.В. Упаковка невыпуклых многогранников в кубоиде минимального объема. Информационные системы и технологии: тез. докл. 5-ой Международная научно-техническая конф., Коблево. 2016. С. 129-130

19. Стоян Ю.Е., Панкратов А.В., Романова Т.Е. Об особенностях задачи оптимизации компоновки невыпуклых многогранников. Математичне та програмне забезпечення інтелектуальних систем: тезиси докладов XIII міжнародної науково-практичної конференції, Днепропетровск. 2016. С. 198-199

20. Стоян Ю.Е. Псевдонормализованная квази ρ -функция для невыпуклых многогранников. Современные проблемы машиностроения: тез. докл. конф. молодых ученых и специалистов, Харьков: Ин-т. проблем машиностроения им. А. М. Подгорного НАН Украины. 2015. С. 18.

ABSTRACT

Stoian Y.E. Mathematical modeling and methods for solving optimization problems of arbitrary polytopes packing. -- Qualifying scientific work on the manuscript.

Thesis for a candidate degree in specialty 01.05.02 – «Mathematical modeling and computational methods». – A.N. Podgorny Institute for mechanical engineering problems of National Academy of Science of Ukraine, Kharkiv national university of radioelectronics of Ministry of Education and Science of Ukraine, Kharkiv, 2018.

At present, interest in effective solving of three-dimensional (3D) objects packing problems is growing rapidly. This is due to extreme complexity of methods of solving such problems and a wide range of practical applications. In this regard, the problem of creating intellectual information technologies for solving optimization problems of 3D objects packing using constructive tools of mathematical and computer modeling, that are easily adaptable to specific technological processes is scientifically and practically important. Note, that 3D objects of arbitrary spatial form, always can be approximated by the polytopes with predetermined accuracy.

Purpose of this work is to improve the efficiency of solving the optimization problems of arbitrary polytopes packing by developing constructive tools of mathematical and computer modeling, new mathematical models and effective optimization algorithms using nonlinear programming methods and modern NLP-solvers (software used for searching the local and global solutions of non-linear programming problems).

Optimization packing problems are arising in the study of actual problems of materials science, medicine, additive technology (3D printing), space engineering, logistics, robotics, mechanical engineering and other priority areas of science and technology.

There are a lot of publications appeared in the foreign literature for the last five years, that deal with practical problems, that can be reduced to the optimization problems of packing of polytopes. Since such problems are NP-complex, solution methodology commonly uses heuristic methods.

The main problem is in the complexity of formalization of the placement constraints for arbitrary polytopes (continuous rotations of polytopes, containment of polytopes within the container, non-overlapping of polytopes, minimal allowable distance between polytopes).

It is known, that the most powerful tool for analytical modeling of relationships between geometric objects is a phi-functions method. For the time being, the phi-functions for some forms of 3D objects were built (convex three-dimensional objects, surface if formed by cylindrical, conical and spherical surfaces, convex polytopes).

However, it was failed to build the tools of mathematical modelling for analytical description of the most important, from the practical stand point, constraints for minimal allowable distances between arbitrary (convex and non-convex) polytopes, that allow continuous rotations.

Note, that existing mathematical models of optimizational placement problems presented in the form of non-linear programming problems don't take into account simultaneously the continuous rotations, the constraints for minimal allowable distances between objects and equilibrium constraints (the minimal deviation of the system from a given point).

The open question, that still remains, is the optimization of the polytopes packing within the various types of the containers, since only cuboids are considered in the most of publications.

Therefore, the problem of developing the effective methods for solving scientific and practical optimization 3D problems of packing the N arbitrary polytopes, which allow continuous rotation and translation, considering the minimal allowable distances and equilibrium constraints in an arbitrary

convex container, which boundary is formed by cylindrical, elliptical, spherical surfaces and planes of minimal volume (*OPP, Optimal Polytope Packing*), that requires the construction of new mathematical modelling tools, exact mathematical models and effective solution methods is actual.

As a result, in this work for the first time there were built the: *normalized* and *pseudonormalized quasi phi-functions* for the analytical description of constraints for minimal allowable distances between the arbitrary polytopes; *normalized* and *pseudonormalized phi-functions* for the analytical description of constraints for minimal allowable distances between the arbitrary polytope and the boundary of an arbitrary convex container. The built radical free pseudonormalized quasi phi-functions allow to describe the domain of feasible solutions of OPP problem as a system of inequalities with smooth functions. The *equilibrium constraints* are formalized for the OPP problem in the form of system of inequalities with smooth functions.

For the first time, a mathematical model of the OPP problem is built as a continuous problem of nonlinear programming (including all its globally optimal solutions) for nonconvex polytopes within a convex container, which boundary is formed by spherical, cylindrical, elliptical surfaces and planes, taking into account the placement and equilibrium constraints that allows to use modern NLP-solvers. The volume, metric characteristic or coefficient of homothety of the container is considered as the objective function of the model; the domain of feasible solutions of mathematical model of OPP problem is described by the system of inequalities with smooth functions. The constraints for metric characteristics of the container, constraints for placement parameters of polytopes can be considered as the systems of additional constraints.

For the first time, a mathematical model of the problem of clustering of nonconvex polytopes (OPC) in spherical, cylindrical and cuboid covering regions of minimum volume is built that allows to generate efficient starting

points to search for local extrema of OPP problem.

The methods of solving problems of geometric design were further developed: a strategy of solving of the OPP problem is proposed and efficient methods of solving of the basic implementations of OPP problem using the modern NLP-solvers were developed that (unlike the existing heuristic approaches): allow to place 3D objects of arbitrary shape approximated by the polytopes with predetermined accuracy; take into account continuous translation and rotation of objects, minimum allowable distances and equilibrium constraints; allow to get a locally optimal solutions for OPP problem, with a better objective function value (compared to the benchmark instances - known published results).

The method of construction of feasible starting point, based on the homothetic transformation of the objects and a decomposition method POLYDEC for the search for local extrema, which allows to reduce the OPP problem of $O(n^2)$ dimension with the $O(n^2)$ number of inequalities to a sequence of subproblems of a $O(n)$ number of inequalities and $O(n)$ number of variables, where n – is a number of convex polytopes that form the non-convex polytopes, were developed. The basis of this method is an iterative procedure, the basic idea of which is as follows: for an efficient starting point of the OPP problem for each convex polytope, that forms the nonconvex polytope, the system of six linear ε -inequalities is being constructed, that guarantees the containment of convex polytope within the individual cuboid container; on each iteration, the system of ε -inequalities for all convex polytopes, pseudonormalized quasi phi-functions only for the pairs of convex polytopes that are “ ε -neighbors” and pseudonormalized phi-functions for those convex polytopes, whose individual containers intersect with the boundary of the container are being included to the domain of feasible solutions of non-linear programming subproblem. The point of the local

minimum of the subproblem on the last iteration of the optimization procedure forms the point of local minimum of the OPP problem.

Based on the proposed in this work tools of mathematical modeling, mathematical models, methods and algorithms, the software for solving the general realizations of OPP problems and a software module for solving the OPC problem were developed. The computational experiments were conducted, the local optimal solutions of the optimization of convex and non-convex polytopes packing within the sphere, cylinder, cuboid, ellipsoid, convex polytope and the convex container, which boundary is formed by the above-mentioned surfaces, with taking into account the equilibrium constraints and constraints for minimal allowable distances are obtained. The results of the comparative analysis with the published worldwide benchmark results are provided.

The certification about the usage of the developed software module of optimization of arbitrary non-oriented polytopes packing, that is used for solving the problem of optimal filling of a given volume with particles of nonspherical shape, during the researches at the "Applied Materials and Materials Processing" department of "Lviv Polytechnic" National University, regarding the possibility of using native powders of titanium alloys of a non-spherical shape, which are by far cheaper and can be applied in additive technologies, is received.

The letter of support from G. Fasano – senior researcher and optimization specialist in field of mathematical modelling and systems optimization of European company “Thales Alenia Space” (<https://www.thalesgroup.com/en>), which highlights the importance of the results obtained in the field of rocket and space engineering, in particular to optimize the layout of spacecraft equipment, is received.

An act of implementation of the proposed in the work tools, models, methods and developed software modules in IT company "Cloud Works" for

solving: optimization problems of 3D printing process in additive manufacturing, that uses the SLS technology; problems of optimal cargo packing within the arbitrary containers in the logistics.

The proposed tools of mathematical modeling of OPP problem are implemented in the educational process of Kharkiv national university of Radioelectronics in course "Mathematical and computer modeling in decision support systems", "Modeling of geometrical objects".

The reliability of the results obtained in the work is confirmed by the reasoned statement of the material, proven statements, the use of modern NLP-solvers, comparing the results of computational experiments by the value of the objective function and time of solution with the existing best world counterparts.

Keywords: Packing, convex containers, non-oriented polytopes, allowable distances, equilibrium constraints, quasi phi-functions, mathematical modeling, nonlinear optimization, decomposition method.

List of publications of the applicant

1. Stoian Y. E., Chugay A. M., Pankratov A. V., Romanova T. E. Two Approaches to Modeling and Solving the Packing Problem for Convex Polytopes // Cybernetics and Systems Analysis. 2018. № 4 (54). P. 585-593.
2. Стоян Ю. Е., Панкратов А.В., Романова Т.Е. Упаковка многогранников в выпуклый контейнер минимального объема // Системи управління, навігації та зв'язку. 2018. № 2 (48). С. 48-54
3. Chugay A., Stoian Ye. Cluster packing of concave non-oriented polyhedra in a cuboid // Advanced Information Systems. 2018. № 2(1). P.16-21
4. Stoian Y., Pankratov A., Romanova T. Optimal clustering of polyhedra // Bionics Intelligence. 2017. № 2 (89). P. 12-22
5. Чугай А.М., Панкратов О.В., Романова Т.Є., Стоян Ю.Є. Оптимізація процесу 3D-друку для SLS технології адитивного виробництва // Системи управління, навігації та зв'язку. 2017. № 6 (46). С. 127-130
6. Стоян Ю.Е., Романова Т.Е., Панкратов А.В. Математическая модель задачи оптимальной компоновки многогранников в выпуклой многогранной области // Системи управління, навігації та зв'язку. 2016. № 3 (39). С. 62-66.
7. Панкратов А.В., Романова Т.Е., Стоян Ю.Е., Чугай А.М. Задача оптимизации упаковки многогранников в сферическом и цилиндрическом контейнерах // Восточно-Европейский журнал передовых технологий. 2016. № 1/4 (79). С. 39-47.
8. Панкратов А.В., Романова Т.Е., Стоян Ю.Е. Псевдонормализованные квази ϕ -функции для многогранников // Радиоэлектроника и информатика. 2015. № 3. С. 22-26.

9. Стоян Ю.Е. Упаковка многогранников в выпуклый контейнер минимального объема. Современные проблемы машиностроения: тез. докл. конф. молодых ученых и специалистов, Харьков: Ин-т. проблем машиностроения им. А. М. Подгорного НАН Украины. 2018. С. 18.

10. Romanova T., Bennell J.A., Stoyan Y., Pankratov A. Packing of arbitrary polyhedra with continuous rotations using nonlinear optimisation. ESICUP meeting: тез. докл. 15-ой международной конференции, Zoetermeer, Netherlands. 2018. P. 17-18

11. Stoian Y.E., Romanova T.E., Pankratov A.V. Packing of polyhedra within a convex container. Problem of Decision Making under Uncertainties: тез. докл. XXX международной конференции, Вильнюс, Литва. 2017. С. 116

12. Стоян Ю.Е. Математическое моделирование оптимизации упаковки многогранников в выпуклой многогранной области. Радиоэлектроника и молодёжь в XXI веке: тез. докл. XXI Международного молодёжного форума, Харьков: ХНУРЭ. 2017. С. 104-105.

13. Romanova T.E., Pankratov A.V., Stoian Y.E. Packing of Polytopes within Spherical and Cylindrical Containers. ESICUP meeting: тез. докл. 14-ой международной конференции, Льеж, Бельгия. 2017. С. 23.

14. Chugay A., Pankratov A., Romanova T. Stoian Yu. Optimization of the process of 3D-Printing for SLS technologies of additive production. Application of Information and Communication Technology and Statistics in Economy and Education: тез. докл. 7-ой международной конференции. – София, Болгария. 2017. FA24.4p.

15. Stoian Yu. Packing polyhedrons using quasi-phi-functions.

Современные проблемы машиностроения: тез. докл. конф. молодых ученых и специалистов, Харьков: Институт проблем машиностроения им. А.М. Подгорного НАН Украины. 2016. С. 33.

16. Stoian Yu. Mathematical model and solution algorithm for packing problem of convex polyhedra. Information Technologies and Computer Modelling: тез. докл. международной научно-практической конф., Ивано-Франковск – Яремча. 2016. С. 187-188

17. Стоян Ю.Е., Панкратов А.В., Романова Т.Е., Чугай А.М. О NLP-задаче оптимальной упаковки невыпуклых многогранников в сферическом и цилиндрическом контейнерах. Математическое моделирование, оптимизация и информационные технологии: тез. докл. 5-ой Межд. науч. конф., Кишинев, Молдавия. 2016. С. 345-348

18. Стоян Ю.Е., Романова Т.Е., Панкратов А.В. Упаковка невыпуклых многогранников в кубоиде минимального объема. Информационные системы и технологии: тез. докл. 5-ой Международная научно-техническая конф., Коблево. 2016. С. 129-130

19. Стоян Ю.Е., Панкратов А.В., Романова Т.Е. Об особенностях задачи оптимизации компоновки невыпуклых многогранников. Математичне та програмне забезпечення інтелектуальних систем: тезиси докладов XIII международной научно-практической конференции, Днепропетровск. 2016. С. 198-199

20. Стоян Ю.Е. Псевдонормализованная квази ρ -функция для невыпуклых многогранников. Современные проблемы машиностроения: тез. докл. конф. молодых ученых и специалистов, Харьков: Ин-т. проблем машиностроения им. А. М. Подгорного НАН Украины. 2015. С. 18.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	4
ВСТУП	9
1 ОГЛЯД ЛІТЕРАТУРИ І ВИБІР НАПРЯМУ ДОСЛІДЖЕННЯ	20
1.1 Аналіз робіт, що присвячено задачам упаковки тривимірних об'єктів	20
1.2 Огляд методів розв'язання задач упаковки.....	26
1.3 Огляд методів геометричного проектування	40
1.4 Висновки по розділу	43
2 ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ	44
2.1 Математичне моделювання в геометричному проектуванні	44
2.1.1 Phi-функція, нормалізована та псевдонормалізована phi-функція.	44
2.1.2 Квазі phi-функція, нормалізована та псевдонормалізована квазі phi-функція.	50
2.2 Постановка базової задачі оптимальної упаковки багатогранників (ОРР).....	53
2.3 Висновки по розділу	65
3 МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ ОПТИМІЗАЦІЙНИХ ЗАДАЧ УПАКОВКИ БАГАТОГРАННИКІВ (ОРР).....	66
3.1 Засоби математичного моделювання обмежень розміщення в задачі ОРР із застосуванням phi -функцій	68
3.1.1 Phi-функції для моделювання обмежень включення.	68
3.1.2 Псевдонормалізовані та нормалізовані phi-функції для моделювання обмежень включення з урахуванням допустимих відстаней.	76
3.2 Засоби математичного моделювання обмежень розміщення в задачі ОРР із застосуванням квазі phi-функцій.....	80

3.2.1	Квазі ρ -функції для моделювання обмежень неперетину.	80
3.2.2	Псевдонормалізовані та нормалізовані квазі ρ -функції для моделювання обмежень на допустимі відстані.....	82
3.3	Математична модель базової задачі оптимальної упаковки багатогранників та її властивості	85
3.4	Висновки по розділу	91
4	МЕТОДИ РОЗВ'ЯЗАННЯ ОПТИМІЗАЦІЙНИХ ЗАДАЧ УПАКОВКИ БАГАТОГРАННИКІВ (ОРР).....	93
4.1	Стратегія розв'язання задачі ОРР	93
4.2	Метод побудови допустимих стартових точок (FSPA).....	94
4.3	Метод пошуку локальних екстремумів (POLYDEC).....	99
4.4	Висновки по розділу	117
5	РЕЗУЛЬТАТИ ОБЧИСЛЮВАЛЬНИХ ЕКСПЕРИМЕНТІВ.....	119
5.1	Вихідні дані для неопуклих багатогранників	119
5.2	Результати обчислювальних експериментів для <i>ОРС</i>	123
5.3	Результати обчислювальних експериментів для ОРР.....	128
5.4	Висновки по розділу	152
	ВИСНОВКИ	153
	ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	155
	Додаток А.....	178
	Додаток Б.....	181
	Додаток В.....	185
	Додаток Г.....	190

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

- C & P – (Cutting and Packing) задача упаковки та розкрою
- OPP – (Optimal Polytopes Packing) задача оптимальної упаковки неопуклих багатогранників
- OPC – (Optimal Polytopes Clustering) задача оптимального кластерінгу пари неопуклих багатогранників
- NLP-solver – (Non Linear Programing Solver) програмне забезпечення для розв’язання задач нелінійного програмування
- POLYDEC – (Polytopes Decomposition) метод декомпозиції
- FSPA – (Feasible Starting Point Algorithm) алгоритм пошуку допустимих стартових точок
- IPOPT – (Interior Point Optimization) програма на основі модифікації методу внутрішньої точки
- Ω – опуклий контейнер
- S** – контейнер: куля
- B** – контейнер: кубоїд
- C** – контейнер: циліндр
- E** – контейнер еліпсоїд
- P** – контейнер: опуклий багатогранник
- A** – контейнер: довільна опукла область, границя якої формується циліндричної, сферичної, еліптичної поверхнями або площиною
- p – вектор змінних метричних характеристик контейнера Ω
- λ – коефіцієнт гомотетії контейнера Ω
- \mathbb{Q}_q – q -й неопуклий багатогранник
- $J_N = \{1, 2, \dots, N\}$ – індексна множина, що містить індекси неопуклих багатогранників
- $q \in J_N$ – індекси неопуклих багатогранників

N – число неопуклих багатогранників

M_q – маса неопуклого багатогранника

c_q – центр мас неопуклого багатогранника

$u_q = (v_q, \theta_q)$ – параметри розміщення неопуклого багатогранника

$v_q = (x_q, y_q, z_q)$ – вектор трансляції неопуклого багатогранника

\mathbb{Q}_q

$\theta_q = (\theta_q^1, \theta_q^2, \theta_q^3)$ – вектор кутів повороту неопуклого багатогранника \mathbb{Q}_q

$\theta_q^1, \theta_q^2, \theta_q^3$ – кути Ейлера

$M(\theta_q)$ – матриця повороту неопуклого багатогранника \mathbb{Q}_q

K_i^q – i -й опуклий багатогранник, що формує неопуклий багатогранник \mathbb{Q}_q

$i \in I_{n_q}$ – індекси опуклих багатогранників

$I_{n_q} = \{1, 2, \dots, n_q\}$ – індексна множина, що містить індекси опуклих багатогранників, які формують неопуклий багатогранник \mathbb{Q}_q

n_q – число опуклих багатогранників, які формують неопуклий багатогранник \mathbb{Q}_q

p_s^{qi} – s -а вершина опуклого багатогранника $K_i^q \subset \mathbb{Q}_q$

$s = 1, \dots, m_i^q$ – індекс вершини опуклого багатогранника $K_i^q \subset \mathbb{Q}_q$

m_i^q – кількість вершин опуклого багатогранника $K_i^q \subset \mathbb{Q}_q$

ρ_{qg} – мінімально допустима відстань між неопуклими багатогранниками \mathbb{Q}_q і \mathbb{Q}_g

ρ_q – мінімально допустима відстань між неопуклим багатогранником \mathbb{Q}_q і границею контейнера Ω

$\mu(p, u) \geq 0$ – функція балансу

$\Omega^* = R^3 \setminus \text{int } \Omega$ – доповнення до контейнера $\Omega \in \{\mathbf{C}, \mathbf{S}, \mathbf{B}, \mathbf{E}, \mathbf{P}, \mathbf{A}\}$

$\Phi^{\mathbb{Q}_q \Omega^*}(u_q)$ – phi-функція для неопуклого багатогранника і доповнення до контейнера Ω

$\Phi^{K_i^q \Omega^*}(u_q)$ – phi-функція для опуклого багатогранника $K_i^q \subset \mathbb{Q}_q$ і доповнення до контейнера Ω^*

$\tilde{\Phi}^{K_i^q \Omega^*}(u_q)$ – нормалізована phi-функція для опуклого багатогранника $K_i^q \subset \mathbb{Q}_q$ і доповнення до контейнера Ω^*

$\hat{\Phi}^{K_i^q \Omega^*}(u_q)$ – псевдонормалізована phi-функція для опуклого багатогранника $K_i^q \subset \mathbb{Q}_q$ і доповнення до контейнера Ω^*

$\Phi'_{qg}(u_q, u_g, u'_{qg})$ – квазі phi-функція для неопуклих багатогранників \mathbb{Q}_q і \mathbb{Q}_g

u'_{qg} – вектор додаткових змінних квазі phi-функції для неопуклих багатогранників \mathbb{Q}_q і \mathbb{Q}_g

$\Phi'_{ij}(u_q, u_g, u'_{ij})$ – квазі phi-функція для опуклих багатогранників $K_i^q \subset \mathbb{Q}_q$ і $K_j^g \subset \mathbb{Q}_g$,

u'_{ij} – вектор додаткових змінних квазі phi-функції для опуклих багатогранників $K_i^q \subset \mathbb{Q}_q$ і $K_j^g \subset \mathbb{Q}_g$,

$\hat{\Phi}'_{ij}(u_q, u_g, u'_{ij})$ – псевдонормалізована квазі phi-функція для

опуклих багатогранників $K_i^q \subset \mathbb{Q}_q$ і $K_j^g \subset \mathbb{Q}_g$

$\Phi^{K_i^P}(u_i, u')$ – phi-функція для багатогранника $K_i^q \subset \mathbb{Q}_q$ і площини P

$u = (\zeta, \tau) \in R^\sigma$ – вектор змінних задачі ОРР

$\zeta = (p, u_1, u_2, \dots, u_N)$ – вектор змінних метричних характеристик контейнера Ω і багатогранників $\mathbb{Q}_q, q \in J_N$

$\tau = (u'^1, \dots, u'^m)$ – вектор додаткових змінних для квазі phi-функцій

$u'^s, s = 1, \dots, m$ – вектор додаткових змінних для s -ї пари опуклих багатогранників

$m = \text{card}(\Xi)$ – потужність індексної множини Ξ

$\Xi = \{(i, j), a_i \neq a_j, i < j = 1, \dots, n\}$ – множина, що включає індекси

пар опуклих багатогранників

$\mathbf{a} = (a_1, \dots, a_N)$ – «склеювальний» вектор

$a_i \in \{1, 2, \dots, N\}$ – компонента «склеювального» вектора \mathbf{a}

$\sigma = p + 6N + 3m$ – число змінних задачі ОРР

$F(u)$ – функція цілі задачі ОРР

W – область допустимих розв'язків задачі ОРР

S_q – сфера, описана навколо неопуклого багатогранника \mathbb{Q}_q

S_i – сфера, описана навколо опуклого багатогранника $K_i^q \subset \mathbb{Q}_q$

k – номер ітерації процедури, що реалізує метод POLYDEC

ε – крок ітерації

Λ_k^ε – штучна множина додаткових « ε -обмежень» на вектори трансляції багатогранників на k -му кроці

$W_k = W \cap \Lambda_k^\varepsilon$ – область допустимих розв'язків підзадачі на k -му

кроці

Ξ_1^k – множина, що містить індекси тих пар опуклих багатогранників, чії індивідуальні контейнери перетинаються на k -му кроці

Ξ_2^k – множина, що містить індекси тих опуклих багатогранників, чії індивідуальні контейнери перетинаються з границею контейнера на k -му кроці

u^0 – стартова точка з області допустимих розв'язків W задачі ОРР

u^* – точка локального екстремуму задачі ОРР

ВСТУП

Актуальність теми. На сучасному етапі інтерес до ефективного розв'язання задач розміщення тривимірних (3D) об'єктів стрімко зростає. Це пояснюється надзвичайною складністю методів їхнього розв'язання та широким спектром практичних застосувань, у тому числі в таких важливих галузях науки та техніки:

– *Промислові технології:* розкрій та пакування промислових матеріалів; компонування приладів та обладнання; багатоканальне мовлення та комунікаційні мережі; компонування приладів і панелей керування; кодування інформації та передачі сигналів; задачі пакування в нанотехнологіях під час моделювання мікроструктури матеріалів;

– *Аерокосмічна галузь:* компонування приладів та обладнання; моделювання структур ракетних палив; повітряне та космічне спостереження;

– *Біотехнології, сільське господарство та медицина:* планування променевої терапії; пакування хромосом у клітині; планування штучних систем зрошення в сільському господарстві;

– *Матеріалознавство:* пакування гранул та їх ущільнення; взаємодія частинок між собою; імітація спікання; атомістичне моделювання; моделювання структур бетону, піску, вугілля, пористих вибухових речовин та твердих матеріалів; дослідження геометрії та відносної стабільності структур супрамолекул;

– *Хімія:* пакування каталізаторів; моделювання абсорбції газу; задачі кристалографії;

– *Дослідження з охорони довкілля та неядерна енергетика:* екологічний контроль стану навколишнього середовища; розміщення об'єктів у зонах підвищеного ризику;

– *Дослідження з енергетики:* розміщення повітряних

енергетичних установок;

– *Ядерна енергетика та її безпека*: моделювання щільних пакувань елементів що виділяють тепло, в активній зоні ядерного реактора;

– *Фізика*: моделювання фізичних процесів горіння шару вугілля.

У зв'язку з цим наукового та практичного значення набуває проблема створення інтелектуальних інформаційних технологій розв'язання оптимізаційних задач розміщення на основі конструктивних засобів математичного та комп'ютерного моделювання, сучасних методів оптимізації та засобів візуалізації різноманітних варіантів розміщення, що легко адаптуються для конкретних технологічних процесів. Це дозволяє за вихідними даними одержати найкращий варіант розміщення відповідно до заданого критерію якості.

Створення комп'ютерних технологій в області оптимізації, системного аналізу і математичного моделювання, математичної кібернетики, обчислювальної математики, математичного забезпечення ЕОМ і автоматизованих систем управління та обробки даних, штучного інтелекту присвячені роботи багатьох відомих українських вчених, зокрема І.В. Сергієнко [1-3], С.І. Ляшко [4, 5], А.І. Шевченко [6, 7], Л.Ф. Гуляницького [8-10], П.І. Бідюка [11, 12], Ю.В. Крака [13, 14], П.І. Стецюка [15-17], Є.М. Кисельової [6, 18, 19], В.П. Шило [20, 21], Р.П. Базилевича [22, 23] і багатьох інших.

Розробка сучасних інформаційних систем для вирішення задач розміщення 3D об'єктів вимагає автоматичної побудови математичних моделей, однак ця операція, будучи одним з найбільш складних і відповідальних етапів операційного дослідження, до сих пір погано формалізована і залишається в значній мірі мистецтвом.

Практика використання існуючих пакетів програм для розв'язання

оптимізаційних задач розміщення часто передбачає, що математична модель задачі побудована і залишається тільки представити її в певному форматі. Таким чином, наявні програмні засоби залишають неавтоматизованим етап побудови моделі, що робить актуальним пошук нових комп'ютерних технологій моделювання постановок оптимізаційних задач в різних предметних областях.

Міжнародна організація ESICUP <http://paginas.fe.up.pt/~esicup/> поєднує провідних учених світу, які досліджують методи розв'язання задач упаковки та розкрою. Серед них Dowsland K., Bennell J., Burke, Kendall G. (Велика Британія), Milenkovic M., Daniels K., Pinter J. (США), Oliveira J., Ferreira S., Gomes M. (Португалія), Birgin (Бразилія), Wascher G., Scheithauer G., Bortfeldt A. (Німеччина), Fasano G. (Італія), Alvarez-Valdes R. (Іспанія), Hifi M. (Франція), Imahori S. (Японія), Egeblad J., Pisinger D. (Данія), Ikonen I. (Фінляндія), S. X. Li, J. Zhao (Китай).

Останні п'ять років у зарубіжній літературі з'являються публікації, в яких розглядаються практичні задачі, що зводяться до оптимізаційних задач упаковки тривимірних об'єктів (зокрема багатогранників), серед яких слід зазначити [24-30].

Оскільки задачі є NP-складними, методології розв'язання зазвичай використовують евристичні методи. Деякі дослідники пропонують підходи на основі математичного моделювання і загальних процедур оптимізації.

Створення ефективних методів розв'язання наукових і практичних оптимізаційних 3D задач упаковки заданого набору, в загальному випадку, неопуклих багатогранників, що допускають неперервні обертання і трансляції, з урахуванням мінімально допустимих відстаней, в контейнер, що оптимізується, вимагає побудови нових засобів математичного моделювання та точних математичних моделей у вигляді задач нелінійного програмування.

Як відомо, найбільш потужним засобом аналітичного моделювання відношень геометричних об'єктів є метод ϕ -функцій.

У рамках наукової школи Ю. Г. Стояна розробці засобів математичного моделювання, математичних моделей та методів розв'язання задач упаковки та розкрою присвячено роботи багатьох його учнів, у тому числі М. І. Гіля, В. М. Комяк, С. В. Яковлева, О. О. Ємця, М. В. Новожилової, Т. Є. Романової, О. В. Панкратова, І. В. Гребенніка, В. М. Пацука, Г. М. Яськова, А. М. Чугая.

Слід зазначити публікації Ю. Г. Стояна, М. І. Гіля, В. М. Комяк, Л. Д. Пономаренко, С. Л. Магаса, О. К. Пандоріна, М. В. Новожилової, О. В. Панкратова, О. В. Карташова, А. Ф. Лойко, Т. Є. Романової, А. М. Чугая, І. В. Магдаліни, А. А. Черноморця, М. В. Злотника, що присвячені оптимізаційним задачам упаковки багатокутників та багатогранників.

На даний час побудовані ϕ -функції для деяких просторових форм орієнтованих 3D об'єктів (опуклі тривимірні тіла, поверхня яких сформована циліндричними, конічними та сферичними поверхнями, багатогранники) та квазі ϕ -функції для деяких просторових форм неорієнтованих 3D об'єктів (сфероціліндри). Більшість з ϕ -функцій (зокрема, для опуклих багатогранників) досить складні, що робить неефективним використання NLP-solvers для пошуку локальних екстремумів. Проте побудувати псевдонормалізовані ϕ -функції для довільних багатогранників не вдалося. Через це з метою аналітичного опису обмежень розміщення з урахуванням мінімально допустимих відстаней між довільними неопуклими багатогранниками, що допускають неперервні повороти і трансляції, може бути використаний клас квазі ϕ -функцій і псевдонормалізованих квазі ϕ -функцій. Вільні від радикалів псевдонормалізовані квазі ϕ -функції дозволять описати область допустимих розв'язків у вигляді системи нерівностей з

гладкими функціями.

Крім того, що об'єкти довільної просторової форми, для яких не існує засобів аналітичного опису відношень, можуть бути апроксимовані багатогранниками з наперед заданою точністю.

Слід зазначити, що існуючі математичні моделі задач упаковки 3D об'єктів з використанням максимінних ϕ -функцій відносяться до класу задач негладкої оптимізації, оскільки область допустимих розв'язків представляється у вигляді об'єднання підобластей, кожна з яких описується системою $O(n^2)$ нерівностей (з функціями, в загальному випадку, що містять радикали), що іноді ускладнює безпосереднє використання сучасних локальних і глобальних NLP-solvers.

Таким чином, математичне і комп'ютерне моделювання оптимізаційних задач упаковки довільних неорієнтованих багатогранників з використанням квазі ϕ -функцій і псевдонормалізованих квазі ϕ -функцій, побудова математичної моделі задачі у вигляді задачі нелінійного програмування (яка включає всі глобально оптимальні розв'язки), область допустимих розв'язків якої описується системою нерівностей з гладкими функціями та розробка методів з оцінкою складності, що лінійна до кількості багатогранників є актуальним.

Зв'язок роботи з науковими програмами, планами, темами.
Дисертаційна робота виконана в період з 2015 р. по 2018 р. у відділі математичного моделювання і оптимального проектування Інституту проблем машинобудування ім. А.Н. Підгорного НАН України відповідно до планів науково-технічних робіт з держбюджетних тем «Створення інтелектуальних інформаційних технологій розв'язання оптимізаційних задач розміщення об'єктів довільних просторових форм» (2012–2016 рр. № ДР 0112U002488), «Розробка математичних моделей та комп'ютерних технологій розв'язання оптимізаційних задач компонування тривимірних об'єктів» (2017–2018 рр. № ДР

0117U000877) в яких здобувач був одним із співвиконавців.

Мета та задачі дослідження. Мета роботи – підвищення ефективності розв'язання оптимізаційних задач оптимальної упаковки довільних багатогранників шляхом розробки конструктивних засобів математичного і комп'ютерного моделювання, нових математичних моделей та ефективних методів локальної оптимізації із застосуванням сучасних *солверів (NLP-solvers)*.

Для досягнення цієї мети поставлені такі наукові задачі:

- сформулювати задачу оптимальної упаковки *неопуклих багатогранників (OPP)* у *опуклому контейнері*, границя якого формується за допомогою сферичних, циліндричних, еліптичних поверхонь та площини, з урахуванням *мінімально допустимих відстаней* та *обмеження балансу (відхилення центра мас системи від заданої точки)*;

- розробити засоби математичного моделювання *обмежень розміщення* (неперетин багатогранників та їх включення в контейнер з урахуванням мінімально допустимих відстаней) з використанням методу *phi-функцій*. Формалізувати *обмеження балансу*;

- побудувати математичну модель задачі *OPP* у вигляді задачі нелінійного програмування (*NLP-model*). Побудувати основні реалізації математичної моделі задачі *OPP* залежно від форми контейнера (*прямий круговий циліндр, кубоїд, куля, опуклий багатогранник, еліпсоїд*); обмежень на мінімально допустимі відстані (*не задані, задані між кожною парою багатогранників, задані між багатогранниками і боковою поверхнею контейнера*); вигляду функції цілі (*метричні характеристики або коефіцієнт гомотетії контейнера*);

- розробити стратегію розв'язання задачі *OPP* на основі особливостей її математичної моделі;

- розробити методи пошуку допустимих та локально

оптимальних розв'язків для основних реалізацій задачі *OPP*;

– створити програмне забезпечення для розв'язання основних реалізацій задачі *OPP*.

Об'єкт дослідження – процес оптимізації упаковки неопуклих багатогранників в контейнері з урахуванням обмежень балансу та обмежень розміщення, включаючи мінімально допустимі відстані.

Предмет дослідження – засоби математичного моделювання, математичні моделі і методи розв'язання задач оптимальної упаковки багатогранників.

Методи дослідження. В роботі застосовуються аналітична геометрія та функціональний аналіз для побудови *phi*-функцій, псевдонормалізованих *phi*-функцій, квазі *phi*-функцій та псевдонормалізованих квазі *phi*-функцій; методи геометричного проектування для побудови математичних моделей та розробки методів пошуку допустимих стартових точок і методів локальної оптимізації для задачі *OPP*.

Наукова новизна отриманих результатів полягає у такому:

а) набув подальшого розвитку метод *phi*-функцій: вперше побудовані *phi*-функції, псевдонормалізовані *phi*-функції, квазі *phi*-функції та псевдонормалізовані квазі *phi*-функції як засоби математичного моделювання обмежень розміщення для задачі *OPP*, що дозволяє описати в аналітичному вигляді: неперетин довільних багатогранників; включення багатогранників в опуклий контейнер; мінімально допустимі відстані між довільними багатогранниками та між багатогранниками та границею контейнера;

б) вперше побудована математична модель задачі *OPP* у вигляді задачі нелінійного програмування (що включає всі глобально оптимальні розв'язки) для неопуклих багатогранників в опуклому контейнері, границя якого формується за допомогою сферичних,

циліндричних, еліптичних поверхонь та площини з урахуванням обмежень розміщення та обмежень балансу, що дозволяє використовувати сучасні *NLP-солвери*;

в) вперше побудована математична модель задачі кластерингу неопуклих багатогранників (*OPC – Optimal Polytopes Clustering*) в сферичній, кубоїдній та циліндричній областях мінімального об'єму, що дозволяє генерувати ефективні допустимі стартові точки для пошуку локальних екстремумів задачі *OPP*;

г) набули подальшого розвитку методи розв'язання задач геометричного проектування: запропонована стратегія розв'язання задачі *OPP* та розроблені ефективні методи для основних її реалізацій, які на відміну від існуючих підходів: враховують одночасно *неперервні трансляції та обертання об'єктів, мінімально допустимі відстані і обмеження балансу*; дозволяють отримувати локально оптимальні розв'язки для задач *OPP*, що є кращими за значенням цільової функції (порівняно з *benchmark instances* – відомими опублікованими результатами).

Практичне значення одержаних результатів. Наукові результати дисертаційної роботи є подальшим розвитком математичного моделювання і обчислювальних методів в геометричному проектуванні: створено нові математичні моделі та розроблено ефективні методи розв'язання оптимізаційних задачах оптимальної упаковки довільних багатогранників, що мають широкий спектр застосувань в пріоритетних областях науки і техніки (включаючи адитивні технології, матеріалознавство, логістику, мінералогію, медицину, нанотехнології, робототехніку, системи розпізнавання образів, системи керування, системи керування космічними апаратами, енергетику, машинобудування, авіабудування, будівництво).

Світовий рівень створених засобів математичного моделювання та

ефективність запропонованих методів підтверджується найкращими результатами обчислювальних експериментів, порівняно із зарубіжними аналогами.

Розроблений програмний модуль оптимізації упаковки довільних неорієнтованих багатогранників застосовується на кафедрі «Прикладне матеріалознавство та обробка матеріалів» Національного університету «Львівська політехніка» для розв'язання задачі оптимального заповнення заданого об'єму частинками неферичної форми.

Отримано листа підтримки від Дж. Фазано – провідного вченого та спеціаліста в області математичного моделювання та оптимізації систем (systems modeling and optimization) європейської компанії «Thales Alenia Space» (<https://www.thalesgroup.com/en>), в якому зазначено важливість отриманих в роботі результатів для розв'язання оптимізаційних задач компоновки в області ракетно-космічного машинобудування.

Запропоновані в роботі засоби, моделі, методи та програмні модулі використовуються в ТОВ «Cloud Works» для розв'язання задач оптимізації процесу 3D-друку, що використовує SLS технологію, а також для розв'язання задач оптимального пакування вантажів у довільних контейнерах у сфері логістики.

Запропоновані засоби математичного моделювання задач *OPP* впроваджені в навчальний процес у Харківському національному університеті радіоелектроніки в курсах: «Математичне та комп'ютерне моделювання в системах підтримки прийняття рішень», «Моделювання геометричних об'єктів».

Особистий внесок здобувача. Усі основні наукові результати дисертаційної роботи були отримані особисто автором. У роботах, написаних в співавторстві, дисертантові належать такі результати: [31, 36, 38, 42, 43, 50] – квазі ρ -функції і псевдонормалізовані квазі ρ -

функції для неопуклих багатогранників, що допускають неперервні обертання; [32, 34, 35, 37, 45–50] – NLP математичні моделі задач упаковки неопуклих багатогранників в різних контейнерах (куля, паралелепіпед, циліндр, опуклий багатогранник, еліпсоїд, опукла область, утворена перетином двох куль) із урахуванням обмежень балансу; [33, 35, 36] – NLP модель задачі кластерінгу пари неопуклих багатогранників в сферичній, кубоїдній та циліндричній областях мінімального об'єму; [39–41, 44] – NLP моделі; метод розв'язання, що включає метод побудови допустимої стартової точки, метод декомпозиції для задач упаковки довільних багатогранників; результати числових експериментів для задачі ОРР у контейнерах, що мають форму кулі, паралелепіпеда, циліндра, опуклого багатогранника, еліпсоїда, опуклої області, утвореної перетином двох куль.

Апробація результатів дисертації. Основні результати роботи доповідались і отримали схвалення на міжнародних конференціях і наукових семінарах: конференції молодих учених і фахівців «Сучасні проблеми машинобудування» ІПМаш ім. А.М. Підгорного НАН України (Харків, Україна, 2015, 2016, 2018 рр.); 5-й та 6-й міжнародних наукових конференціях «Математичне моделювання, оптимізація і інформаційні технології» (Кишинів, Молдова, 2016 р.); 5-й міжнародній науково-технічній конференції «Інформаційні системи і технології» (Коблево, Україна, 2016 р.); міжнародній науково-практичній конференції «Інформаційні технології і комп'ютерне моделювання» (Івано-Франківськ – Яремча, Україна, 2016 р.); XIII і XVI міжнародних науково-практичних конференціях «Математичне та програмне забезпечення інтелектуальних систем» (Дніпропетровськ, Україна, 2016 р.); XIII міжнародній науково-практичній конференції «Теоретичні і прикладні аспекти побудови програмних систем» (Київ, Україна, 2016 року); 7-й міжнародній конференції «Додаток інформаційних і

комунікаційних технологій і статистики в економіці і освіті» (Софія, Болгарія, 2017 р.); міжнародному молодіжному форумі «Радіоелектроніка і молодь в XXI столітті» (Харків, Україна, 2017 р.); XXX міжнародній конференції «Проблеми ухвалення рішень в умовах невизначеності» (Вільнюс, Литва, 2017 р.); 14-й міжнародній конференції «ESICUP meeting» (Льєж, Бельгія, 2017 р.); 15-й міжнародній конференції «ESICUP meeting» (Зотермеєр, Нідерланди, 2018 р.); семінарах Харківської секції Наукової ради з проблеми «Кібернетика» (Харків, Україна, 2017, 2018 рр.); семінарі відділу методів негладкої оптимізації, Інституту кібернетики ім. В.М. Глушкова НАН України (Київ, Україна, 2018 р.).

Публікації. За темою дисертації опубліковані 20 наукові роботи [31-50], у тому числі 6 статей в наукових спеціалізованих виданнях, які входять до переліку ДАК МОН України (з них 2 – до наукометричної бази SCOPUS) та 2 наукові статті у технічних виданнях України; 12 тез доповідей на міжнародних наукових конференціях.

Структура та обсяг дисертації. Робота складається із вступу, п'яти розділів, висновків та додатків. Загальний обсяг роботи складає 210 сторінок. Дисертація містить перелік використаних джерел із 211 найменувань на 22 сторінках.

1 ОГЛЯД ЛІТЕРАТУРИ І ВИБІР НАПРЯМУ ДОСЛІДЖЕННЯ

1.1 Аналіз робіт, що присвячено задачам упаковки тривимірних об'єктів

Високотехнологічні галузі – авіабудування, машинобудування та космічна промисловість – потребують виготовлення виробів з підвищеним ресурсом роботи, мінімальними витратами фінансів та енергоресурсів. Таким вимогам відповідають адитивні технології, для яких потрібні конструкційні матеріали, з яких на лазерних 3D принтерах можна виготовляти не тільки окремі деталі, а й елементи конструкцій складної конфігурації. Ці технології дають можливість виготовляти продукцію за короткий проміжок часу, без додаткової фінішної обробки та здатністю відновлення ушкоджених виробів. Актуальною задачею для сучасних підприємств є оптимізація матеріальних і фінансових ресурсів, підвищення ефективності виробництва та зміна в цілому світогляді промислового виробництва [51]. Прискорити процеси проектування, розробки та виготовлення конструкторських деталей сьогодні дозволяють адитивні технології (3D друк). В останні 10 років технології 3D друку бурхливо розвиваються (основні споживачі – аерокосмічна промисловість, медицина, енергетика, автомобільна промисловість): з'являються нові, удосконалюються існуючі, виникають перспективні напрями використання існуючих технологій (SLS, SLM, FDM та ін.) (рис. 1.1) [52].

Тому актуальними є дослідження з розробки інформаційних технологій, які дозволять суттєво зменшити фінансові витрати та вдосконалити процес промислового виробництва, що застосовують сучасні адитивні технології, за рахунок економії часу та енергії.

Економія часу та енергії може бути досягнута завдяки:

максимізації кількості виробів, що одночасно виготовляються; мінімізації кількості шарів друку за один робочий цикл 3D принтера.

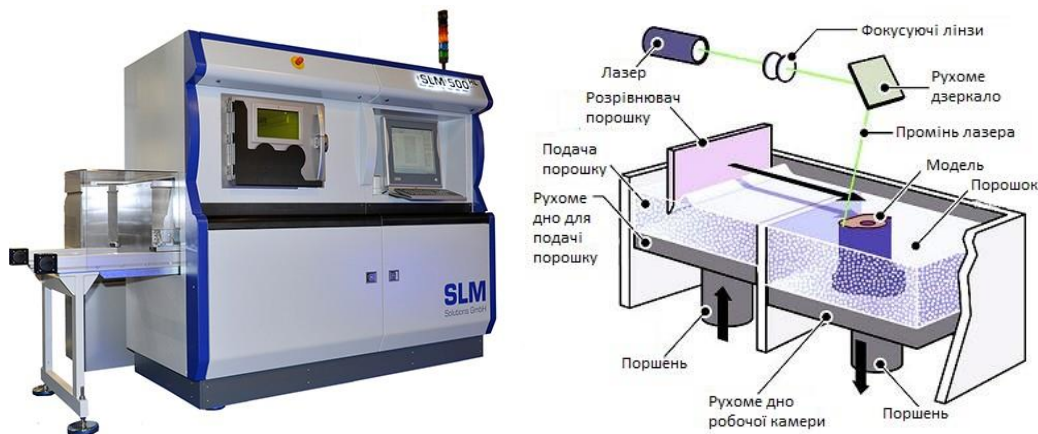


Рисунок 1.1 – Адитивне виробництво

Головною особливістю адитивної технології є застосування порошоків, що складаються з частинок металу, покритих полімером. Після процесу спікання деталь поміщається у високотемпературну піч, де пластик вигорає, а його місце займає легкоплавка бронза. До переваг технології можна віднести відсутність необхідності в матеріалах підтримки. Деталь занурена в порошок, який і виконує функцію підтримки [53].

Тому розробка методів прискорення циклів друку за рахунок одночасного друку відразу декількох моделей із забезпеченням щільного заповнення усього об'єму робочої камери 3D принтера, що використовує SLS технологію, є на сьогодні також актуальною задачею (рис. 1.2) [54].

З цією метою розв'язуються оптимізаційні задачі компонування тривимірних об'єктів на основі конструктивних засобів математичного та комп'ютерного моделювання, математичних моделей, ефективних методів оптимізації та сучасних NLP-solvers (програмне забезпечення для розв'язування задач нелінійного програмування).

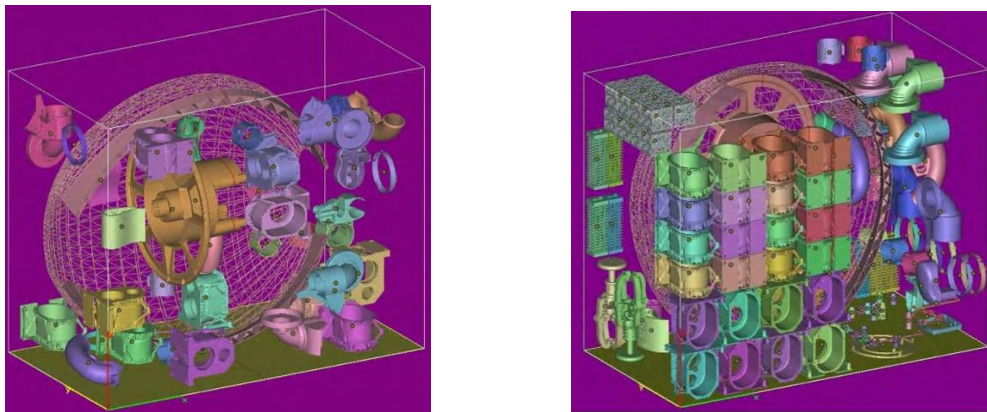


Рисунок 1.2 – Максимізація кількості виробів, які можуть одночасно виготовлятися за один запуск принтера

Також цікавим є застосування досліджуваного класу оптимізаційних задач розміщення тривимірних об'єктів у матеріалознавстві.

На момент у багатьох галузях промисловості використовують порошкові матеріали різного хімічного та фазового складу. В нашому випадку – це порошки титанових сплавів різних систем легування.

Вважається, що сферична форма частинок порошку є найкращою для 3D друку, однак технологія виготовлення таких порошкових матеріалів складна, у зв'язку з чим їх собівартість становить 800-1000\$ за кілограм. Тому основна задача – визначити можливість застосування порошків титанових сплавів несферичної форми, які пройшли процес гідрування і дегідрування, є на порядок дешевшими та можуть з успіхом використовуватись в адитивних технологіях. Тобто завдання полягає у оптимальному заповненні заданого об'єму частинками несферичної форми, які показані на рис. 1.3.

В процесі логістики також застосовуються задачі оптимізації упаковки та розкрою, наприклад, для упаковки товарів в контейнер, розподілі завантаження вантажівок (рис. 1.4 [55]) на підставі маршруту доставки або для оптимальної упаковки товарів в контейнер, який має

бути відправлений морем в місця призначення, з урахуванням обмеження балансу.

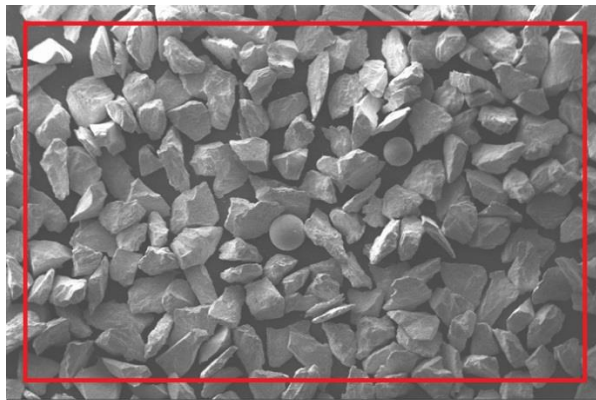


Рисунок 1.3 – Морфологія будови несферичних частинок порошку титанового сплаву



Рисунок 1.4 – Приклад оптимізації упаковки вантажів

Вагомий внесок у розвиток методів розв'язання оптимізаційних задач в галузі ракетно-космічного машинобудування внесли такі вчені, як Janos D. Pinter та Giorgio Fasano. Janos D. Pinter – відомий дослідник в області математичного моделювання та оптимізації систем (systems modeling and optimization), включаючи розробку алгоритмів і програмного забезпечення. Giorgio Fasano працює у великій європейській компанії «Thales Alenia Space» [56]. Ця компанія

спеціалізується на розробці, побудові, тестуванні та керуванні інноваційними космічними системами.

У монографіях серії Springer Optimization, опублікованих під редакцією G. Fasano та J. Pintér, наводиться ряд досліджень, присвячених оптимізаційним задачам упаковки в ракетно-космічному машинобудуванні [57, 58, 59]. Як сучасні підходи до розв'язання задач упаковки в галузі космічного машинобудування розглядаються MILP-алгоритми (Mixed Integer Linear Programming Algorithms), для застосування яких використовується апроксимація розміщуваних 3D об'єктів δ -паралелепіедами (рис. 1.5) [57].

Аналіз публікацій з цієї тематики показав, що оптимізація компоновки окремих агрегатів космічного апарату (КА) є надзвичайно важливим завданням як для виведення КА на орбіту й забезпечення працездатності всіх агрегатів в умовах космічного польоту, так і для успішного управління орієнтацією КА. На етапі проектування КА при компонованні його складових необхідно витримати необхідні геометричні обмеження, зумовлені вимогами щодо аеродинамічних характеристик та міцності, які має задовольняти ракета-носій з КА. Для забезпечення працездатності всієї сукупності агрегатів в умовах космічного польоту необхідно також задовольнити обмеження по взаємному розташуванню окремих агрегатів з точки зору відведення тепла або взаємовпливу агрегатів один на одного в процесі їх роботи з метою зниження взаємних перешкод або взаємного впливу. І, нарешті, розміщення агрегатів впливає на геометрично-масові характеристики КА, які виключно важливі при розробці системи управління орієнтацією КА. Однак задачі балансної компоновки є найменш дослідженими в класі задач розкрою та пакування. Для розв'язання задач балансної компоновки використовуються евристичні та мета-евристичні алгоритми, що зазвичай не дозволяє отримати оптимальних розв'язків.

До того ж запропоновані підходи не враховують допустимих відстаней між об'єктами.

Таким чином, виникає необхідність в побудові математичних моделей для задач балансного компоунання з використанням багатогранної апроксимації просторових форм об'єктів розміщення та контейнерів з урахуванням заданих мінімально допустимих відстаней. Важливою проблемою також є розробка ефективних алгоритмів розв'язання задач балансного компоунання з використанням сучасних програм для розв'язування багатоекстремальних задач нелінійного програмування.



Рисунок 1.5 – Апроксимація обладнання, що розміщується в європейському лабораторному модулі «Каламбус» Міжнародної космічної станції

Одним з найновіших застосувань задач нерегулярної упаковки багатогранників є використання в ортопедичній хірургії, при створенні сучасних комп'ютерних технологій для проведення операцій з відновлення пошкоджених твердих тканин.

1.2 Огляд методів розв'язання задач упаковки

Серед класу задач, присвячених оптимальному розміщенню геометричних об'єктів, особливий інтерес становлять задачі упаковки тривимірних геометричних об'єктів (3D Cutting&Packing, у подальшому C&P).

Базова задача оптимальної упаковки 3D об'єктів формулюється таким чином: упакувати заданий набір геометричних 3D об'єктів в деякій області так, щоб виконувалися задані обмеження на їхнє розміщення, а функція мети досягала свого екстремального значення.

Оптимізаційні задачі упаковки тривимірних геометричних об'єктів мають широкий спектр застосування, наприклад, в сучасній біології, мінералогії, медицині, матеріалознавстві, нанотехнологіях, робототехніці, системах розпізнавання образів, системах керування космічними апаратами, а також в хімічній промисловості, енергетиці, машинобудуванні, суднобудуванні, авіабудуванні, будівництві, 3D друку, логістиці.

Задачі оптимальної упаковки є NP-складними [60] (у загальному випадку точний розв'язок задачі упаковки неможливо отримати за час, обмежений поліномом, міра якого дорівнює розмірності задачі). Через це для розв'язання 3D задач упаковки, як правило, використовуються евристики (див., наприклад, [27, 61–70]). Деякі дослідники пропонують підходи, що ґрунтуються на математичному моделюванні і оптимізаційних процедурах; наприклад, див. [71–74].

Уперше класифікацію задач упаковки та розкрою запропонував Duschhoff [64]. У роботі Wascher та ін. [65] наведена вдосконалена класифікація задач упаковки та розкрою, згідно з якою задачі упаковки можна розділити на два класи. До першого класу (output maximisation) належать задачі, в яких необхідно упакувати максимальну кількість

об'єктів із заданої множини в область фіксованих розмірів, зокрема: упаковка однакових об'єктів (Identical Item Packing Problem), упаковка близьких за розмірами об'єктів (Placement Problem), упаковка різних за розмірами об'єктів (Knapsack Problem). До другого класу (input minimisation) належать задачі, в яких необхідно або мінімізувати метричні характеристики області розміщення (Open Dimension Problem), або мінімізувати кількість областей розміщення фіксованих розмірів (Cutting Stock Problem і Bin Packing Problem).

Слід зазначити, що у більшості робіт розглядаються методи розв'язання задач упаковки куль [75–79]. Алгоритми упаковки куль можуть бути віднесені до таких груп: оптимізація по групам змінних, колективна перестановка, змінні межі та інші оптимізаційні підходи [80].

У дисертаційній роботі [81] наведений досить повний аналіз літератури, що присвячено задачам упаковки тривимірних об'єктів. Цей аналіз показав, що як розміщувані об'єкти, в основному, виступають або кулі, або паралелепіпеди. У роботі [82] автор описує аналітичну модель ортогональної упаковки «тетрисоподібних» елементів. У [83] будується математична модель упаковки сферополієдрів і пропонується швидкий алгоритм розв'язання.

Автори статті [84] пропонують ефективний метод розв'язання задач упаковки орієнтованих багатогранників у опуклий багатогранний контейнер.

Метод моделювання щільних матеріалів на основі алгоритму упаковки сферополієдрів, запропонований в роботі [85], може бути застосований для віртуального проектування щільних матеріалів, що мають в складі несферичні частинки.

Lee та ін. [86] наводять комп'ютерну модель, призначену для визначення щільності гранульованих матеріалів шляхом упаковки

тривимірних еліпсоїдів. Запропонований спеціальний підхід, що запобігає блокуванню частинок у міру їхнього просування вглиб контейнера.

У [87] наведений огляд робіт, присвячених упаковці сфероїдів, циліндрів і тіл довільної форми в задачах порошкової металургії. Описані в роботі підходи ґрунтуються на різних апроксимаціях довільних частинок і використанні евристичних методів.

Розв'язанню задачі оптимального розміщення вантажних контейнерів різних форм (циліндри, паралелепіпеди та інші нестандартні форми, утворені за допомогою паралелепіпедів) присвячені роботи [88–90].

У статті Egeblad та ін. [91] викладається ефективний спосіб розв'язання задачі упаковки багатогранників усередині багатогранного контейнера. Основна ідея методу полягає в точній одновимірній трансляції заданого багатогранника в положення, яке мінімізує його об'єм перетину з усіма іншими багатогранниками. Алгоритм трансляції використовує евристичні та мета-евристичні методики. Наводяться результати чисельних експериментів в задачі упаковки багатогранників в прямокутному контейнері. Використання простору контейнера покращується в середньому понад 14 відсотків порівняно з попередніми методами, запропонованими в [92]. Проте в експериментах допускається загальний об'єм перекриття, що відповідає одному відсотку від загального об'єму усіх багатогранників для цієї задачі.

У статті Liu та ін. [28] пропонується евристичний алгоритм, НАРЕЗD, що ґрунтується на принципі мінімальної загальної потенційної енергії для задачі 3D упаковок неопуклих багатогранників в прямокутному паралелепіпеді фіксованої ширини і довжини, але змінної висоти. Мета полягає в тому, щоб максимізувати коефіцієнт заповнення контейнера. Розглядаються довільні багатогранники, які можна обертати

навколо кожної осі координат. НАРЕЗD не будує No-Fit багатогранник. Результати обчислювальних експериментів демонструють високу продуктивність НАРЕЗD і доводять, що він може досить добре поєднуватися з мета-евристичним алгоритмом для подальшого поліпшення якості упаковки.

Серед найбільш важливих задач слід виділити задачу пошуку найбільш вигідного взаємного розташування об'єктів різної природи з урахуванням різних обмежень. Під час розв'язання такого роду задач важливо дотримуватися мінімально допустимих відстаней між розміщуваними об'єктами, щоб уникнути різного роду впливів (теплого, електромагнітного тощо) одного об'єкта на інший. Наприклад, в роботі [93] побудовано математичну модель і запропоновано метод розв'язання задачі розміщення циліндрів, прямокутних паралелепіпедів, а також спеціальних об'єктів в прямій призмі з урахуванням мінімально допустимих відстаней і зон заборони.

У роботі [94] Torquato і Jiao досліджують щільну упаковку багатогранників. Автори формулюють задачу отримання щільної упаковки об'єктів у фундаментальну сітку з урахуванням граничних умов як оптимізаційну, яку вони називають адаптивною схемою елементів, що стискаються.

Через те що в початковий період багато дослідників незалежно займалися розв'язанням задач, що належать до проблематики розкрою і упаковки, в промисловості (і науковій літературі), виникла безліч найменувань задач, часто різних для задач одного типу, або, навпаки однакових для задач різного типу. Таким чином, виникла потреба в конструктивній класифікації задач на основі дослідження їхніх характерних особливостей. У 1990 році Duskhoff запропонував типологію задач розміщення [64]. Вдосконалена класифікація типів задач C&P, наведена Wäscher, Haußner і Schumann в [65].

Одними з найбільш поширених методів розв'язання 3D задач упаковки прямокутних паралелепіпедів в контейнер (container loading problem) є підходи побудови стін (wall-building approaches) [95, 96], коли секції контейнера по усій ширині і висоті заповнюються в першу чергу. Існує і ряд інших, що ґрунтуються на правилах алгоритмів розв'язання задач упаковки прямокутних паралелепіпедів: гільйотинне різання (guillotine cutting) [97], упорядковування паралелепіпедів (cuboid arrangements) [98], алгоритми пошуку із заборноюю (Tabu search algorithms) [99].

Dai та ін. [100] запропонували спеціальний алгоритм для отримання упаковки тривимірних об'єктів різних форм. У роботі [101] було використано подання октадера для апроксимації геометричних форм об'єктів.

У роботі [90] для розв'язання нестандартних тривимірних задач упаковки за наявності додаткових обмежень наведений евристичний підхід, що ґрунтується на частково цілочисельному програмуванні.

Алгоритми, що ґрунтуються на правилах, є детермінованими методами, які ефективні для конкретних типів задач. Проте важко встановити правила, вживані для упаковки об'єктів різних просторових форм або для функцій мети, що є відмінними від щільності упаковки.

Традиційні оптимізаційні алгоритми, такі, як методи гілок і меж, методи лінійного програмування і градієнтні методи, також можуть бути використані для розв'язання задач упаковки.

Martello, Pisinger і Vigo [102], а також den Boef та ін. [103] навели алгоритм гілок і меж для задач про ранець (knapsack problems) та метод гілок і меж для 3D задач упаковки різних прямокутних паралелепіпедів, що ґрунтується на цьому алгоритмі, в мінімальну кількість однакових прямокутних контейнерів.

Методи градієнтного спуску (Gradient Descent). Підхід, що

ґрунтується на використанні методів градієнтного спуску, включає зведення задачі оптимального розміщення до задачі математичного програмування, побудова одного допустимого розв'язку евристичним методом з подальшою оптимізацією спуском з допустимої стартової точки.

Landon і Balling [104] використовують градієнтні методи під час розміщення тривимірних об'єктів в контейнерах відповідно до критеріїв, що визначаються властивостями простору і маси. Об'єкти моделюється за допомогою контурного подання. Мінімальна відстань між парою об'єктів (minimum separation distance) розраховується шляхом знаходження двох найближчих одна до однієї точок на відповідних тілах. Відстань накладення (interference distance) розраховується як відстань, на яку один об'єкт має транслюватися в певному напрямі від іншого об'єкта так, щоб об'єкти торкнулися один одного. Обидва види відстаней обчислюються шляхом розв'язання оптимізаційної задачі з обмеженнями.

Градієнтні методи є детермінованими, що може значно зменшити час обчислень, необхідний для отримання локально-оптимального розв'язку, оскільки безліч напрямів пошуку обмежена лише перспективними напрямками. Проте розв'язок в цих випадках є лише найближчим локальним оптимумом по відношенню до початкової точки. Тому потрібні багаторазові повторення процесу розв'язання для різних початкових точок. За відсутності можливості вичислити градієнти явно, як це має місце для багатьох задач упаковки, різниці апроксимації можуть бути неточними, що може призвести до помилок в пошуку напряму руху.

Одним зі способів розв'язання цієї проблеми, який пропонується, є метод мультистарту [105] – випадкове відновлення пошуку для певної кількості ітерацій. Якщо нова точка має кращу оцінку, ніж поточний

розв'язок, він замінюється на кращий варіант, і пошук триває шляхом генерації точок з околу нового розв'язку. Якщо ж нова точка має гіршу оцінку, ніж поточний розв'язок, то вона відкидається, і процес пошуку триває в околі вихідної точки.

Пошук із заборонами, розширений локальний пошук (Tabu Search).

Метаевристичний алгоритм пошуку заборонами є розвитком методу градієнтного спуску і був уперше запропонований Glover [106], але став широко відомий і отримав визнання лише після публікацій цього ж автора [107, 108]. На відміну від методу градієнтного спуску, в методі пошуку із заборонами оцінюється підмножина сусідніх розв'язків на кожній ітерації і здійснюється перехід до розв'язку з кращою оцінкою з розглянутих. Поточний розв'язок завжди замінюється на кращий сусідній, навіть якщо новий розв'язок має гіршу оцінку якості. До того ж, зберігається обмежений список, що містить історію раніше розглянутих розв'язків. Цей "табу" список використовується, щоб уникнути повторного розгляду областей, які вже були розглянуті в недавній історії пошуку.

Методи покрокового поліпшення (Iterative Improvement Searches).

Для методів покрокового поліпшення наступний розв'язок для оцінки лежить в околі поточного розв'язку. Таких методів вимагає визначення відповідної локальної функції мети для оцінки сусідніх розв'язків. Найчастіше використовується різновид методу оптимізації по групах змінних, що називається методом послідовно-одиничного розміщення. Слід зазначити, що для початку роботи методу потрібний початковий розв'язок, причому бажано такий, що має гарну оцінку якості. Як правило, такий розв'язок отримується за рахунок застосування конструктивної (зазвичай швидкої) евристики. Часто для задач розміщення початковий розв'язок будується на основі послідовності, отриманої шляхом сортування об'єктів в порядку зменшення висоти,

ширини або площі. Це, як правило, забезпечує досить хороший розв'язок, тому що великі об'єкти розміщуються спочатку і менші об'єкти можуть бути розміщені в отворах між великими.

Генетичні та еволюційні алгоритми (Genetic and Evolutionary Algorithms). Для пошуку розв'язків задач упаковки широко застосовуються генетичні алгоритми. Вони часто використовуються під час створення надвеликих інтегральних схем (НВІС) [109, 110]. У роботі [111] запропонований генетичний алгоритм для 3D упаковки прямокутних паралелепіпедів в контейнер.

Для швидкого прототипування Ikonen та ін. [112] застосували генетичний алгоритм під час упаковки 3D неопуклих об'єктів в циліндричний контейнер. Допускаються тільки повороти з кроком 45 навколо кожної координатної осі.

Використовується метод штрафних функцій, в якому на неприпустимі розв'язки накладається штраф. Розрахунок перетину між об'єктами ґрунтується на перетині трикутників, що покривають поверхню об'єктів. Щоб скоротити кількість обчислень під час визначення перетину об'єктів, в першу чергу перевіряються на перетин їхні обмежувальні рамки.

Генетичні алгоритми є стохастичними алгоритмами, в яких закладена велика міра випадковості. Це алгоритми нульового порядку, для яких потрібно тільки значення функцій. Вони надійні й ефективні у разі нерівних і розривних функцій. Ціна, яку треба платити за таку універсальність, полягає в тому, що ці методи часто вимагають великої кількості обчислень функцій для досягнення оптимального розв'язку, навіть для простих задач. Тому вони вважаються найбільш корисними для задач, в яких обчислення функцій не потребують великих витрат.

Детальний аналіз генетичних алгоритмів, зроблений Goldberg [113]

Штучні нейронні мережі (Artificial Neural Networks). Штучні нейронні мережі ґрунтуються на моделюванні процесу обробки інформації, що відбувається в мозку. Система складається з багатьох простих вузлів (процесорів) або нейронів. Кожен нейрон має один або декілька зважених входів і виводить деяке значення, залежне від того, чи перевищив рівень активації нейрона деяке порогове значення. Можливість навчання – одна з головних переваг нейронних мереж перед традиційними алгоритмами.

Основоположне дослідження нейронних мереж було проведено в 1943 році McCulloch і Pitts [114]. В області розкрою та упаковки нейронні мережі, як правило, використовуються у поєднанні з іншими методами пошуку і можуть бути застосовані для отримання вхідної послідовності, що задає порядок, в якому об'єкти мають бути розміщені. Вони також були використані для розробки стратегій розміщення.

Мурашиний алгоритм (Ant Algorithms). Алгоритм є відносно новим і ґрунтується на моделюванні здатності мурашок до пошуку найкоротшого маршруту між житлом і джерелом їжі. Основна ідея, що лежить в основі алгоритмів мурашиної колонії, – використання механізму позитивного зворотного зв'язку, який допомагає знайти найкращий наближений розв'язок в складних задачах оптимізації. Кожна ітерація полягає в запуску агента («мурашки»), який "намагається" вибрати найкращий маршрут до оптимуму функції («їжі»), за деяким правилом, що враховує «феромонні мітки» своїх попередників.

Приклад використання алгоритму мурашки для розв'язання простої задачі відшукування шляху між двома точками обговорюється в основоположній роботі Dorigo [115]. Система працює з позитивним зворотним зв'язком, внаслідок чого чим більша кількість мурашок прямує певним шляхом, тим сильніше стає слід феромону і вище

вірогідність вибору цього шляху наступними мурашками. Крім того, з часом «феромон випаровується», і привабливість рідко відвідуваних шляхів падає.

Метод імітації відпалу (Simulated Annealing). Імітаційний відпал [116] є загальнозастосовним стохастичним методом, що ґрунтується на аналогії між процесом відпалу металів і розв'язком комбінаторних оптимізаційних задач великої розмірності. У алгоритмі вибирається початкова точка і для неї обчислюється значення цільової функції.

Робиться крок з метою переходу в нову точку шляхом застосування оператора з доступного набору рухів. Обчислюється значення цільової функції в новій точці. Якщо крок приводить до поліпшення цільової функції, нова точка приймається і стає поточною. Якщо цільова функція погіршується, крок все ще може бути прийнятий з деякою вірогідністю. Ця вірогідність залежить від параметра, що називається температурою. Чим більша температура, тим вище ця вірогідність. Спочатку, коли значення температури велике, алгоритм набуває будь-якого нового стану незалежно від того, наскільки він хороший. Це призводить до випадкових блукань в просторі. Потім температура повільно зменшується, досягаючи, врешті-решт, нуля, після чого алгоритм вироджується в звичайний алгоритм локального пошуку. Швидкість, з якою знижується температура, називається розкладом охолодження. Чим більше за часом «розтягується» цей розклад, тим довше йде дослідження простору пошуку.

Алгоритми імітаційного відпалу успішно застосовуються під час компонування мікросхем [117]. Jajodia та ін. [118] за допомогою імітаційного відпалу навели розв'язок міжелементних (inter cell) і внутрішньоелементних (intra-cell) задач упаковки, суть якого полягає в такому взаємному розміщенні об'єктів, за якого зводиться до мінімуму загальний матеріальний потік між ними.

Застосування алгоритму імітаційного відпалу було порівняне із застосуванням інших методів упаковки об'єктів, і під час розв'язання кожної з безлічі класичних тестових задач було отримано або такий же, або кращий розв'язок.

Szykman і Cagan [119, 120] розширили застосування цього підходу на тривимірні механічні та електромеханічні компонування. У їхньому підході об'єктами виступають паралелепіпеди та циліндри, які допускають повороти, що є кратними 90° . Використовувався підхід, що ґрунтується на збуреннях, в якому неприпустимі точки з перетином об'єктів і порушеннями обмежень дозволяються та штрафуються [121]. Безліч рухів включає трансляцію об'єктів, їхні повороти та перестановки. Адаптивний розклад відпалу (adaptive annealing schedule) [122] було використано для контролю температури, а стратегію імовірнісного вибору руху [123] – для вибору рухів на основі попередніх характеристик.

Комплексний підхід до задач тривимірного компонування і задач трасування був поданий в роботі Szykman і Cagan [124]. Експерименти показали, що одночасне виконання компонування і трасування приводить до кращих результатів, ніж за типового підходу, коли спочатку відбувається компонування, а тільки потім трасування (layout – then – route approach).

Kolli та ін. [125] узагальнили роботу Szykman і Cagan, ослабивши обмеження на геометричну форму об'єктів і повороти. Для швидкої оцінки перетину між частинами об'єктів складних форм використовуються октадерева. Можна регулювати рівень деталізації октадерева, що використовується на певній стадії алгоритму імітаційного відпалу, щоб зробити алгоритм ефективнішим.

Імітаційний відпал є загальним методом для найрізноманітніших комбінаторних задач. Метод є гнучким і потужним. Для алгоритму

імітаційного відпалу потрібна велика кількість обчислень, щоб алгоритм був зведений до хорошого розв'язку. Коли цільова функція є складною, обчислення можуть бути дорогими і займати багато часу. Campbell та ін. [126] використали ієрархічний аналіз теплопередачі (hierarchical heat transfer analysis) з метою зменшити час обчислень під час розміщення тепловиділяючих електронних компонент.

Алгоритми прямого пошуку за шаблоном. (Pattern search algorithm). Алгоритми прямого пошуку за шаблоном є підмножиною алгоритмів прямого пошуку, наведених Hooke і Jeeves [127]. Вони ефективні у випадках, коли цільова функція є нелінійною, багатоекстремальною, для якої є недоступною інформація про похідні.

Методи прямого пошуку визначаються як послідовне дослідження пробних розв'язків, що включають порівняння кожного пробного розв'язку з найкращим вже отриманим розв'язком, а також стратегію визначення наступного пробного розв'язку.

Torczon і Grosset [128] розвинули загальну структуру і ключові елементи цих методів і визначили загальну концепцію, що називається узагальненим методом пошуку за шаблоном.

Порівняно з імовірнісними методами імітаційного відпалу, методи пошуку за шаблоном досліджують простір розв'язків у більш обмеженій манері. Рух допускається тільки уздовж шаблонних напрямів. Довжина кроку змінюється відповідно до певних правил так, що спочатку кроки є великими і в процесі пошуку поступово зменшуються. Це дозволяє методам пошуку за шаблоном сходитися за менше число ітерацій, ніж в методі імітаційного відпалу.

Yin і Sagan [129] уперше застосували алгоритм пошуку за шаблоном до 3D задач розміщення. Введено п'ять модифікацій, покликаних допомогти отримувати якісні розв'язки задач розміщення, що в результаті привело до розширеного алгоритму пошуку за шаблоном.

Цей алгоритм був застосований до ряду тестових і промислових задач [130] і показав поліпшення швидкості на один-два порядки порівняно з алгоритмом імітаційного відпалу для розв'язків аналогічної якості. В алгоритм були включені різні евристики, щоб вести пошук уздовж перспективних напрямів [131].

У розширеному алгоритмі пошуку за шаблоном множина шаблонних напрямів може бути оновлена відповідно до ефективності попередніх ітерацій. Дозволено як перетворення трансляції, так і повороти. Перестановки використовуються тільки тоді, коли за цієї довжини кроку неможливо перейти в кращу точку шляхом трансляції і поворотів.

Порядок вибору компонент для пробних рухів і напрямів пошуку випадковий. Це робить послідовність цих рухів менш детермінованою, щоб уникнути попадання в одні і ті ж точки локального екстремуму. Крок-стрибки (*step-jumps*) можуть бути застосовані, щоб дозволити довжині кроку час від часу збільшуватися. Це робить пошук більше вичерпним, відкладаючи виконання критерію зупинки.

Розширений алгоритм пошуку за шаблоном більше детермінований, ніж алгоритми імітаційного відпалу з точки зору напрямку руху, довжини кроку та прийняття нової точки як змінної. Проте він має також і стохастичні елементи.

Aladahalli та ін. [132, 133] для розв'язання 3D компонувальних задач розробили спеціальний алгоритм пошуку за шаблоном, в якому функції мети та обмеження об'єднуються в одну оцінну функцію (*objective function-based pattern search*). В першу чергу застосовуються рухи, які приводять до найбільшої зміни (поліпшення) цільової функції.

Змішані підходи (Combined approaches). Змішані підходи мають переваги двох або більше алгоритмів пошуку і поєднують їх в процесі розв'язання задач. Dai і Cha [100] подали змішаний підхід, що поєднує в

собі евристичні правила й алгоритм нейронних мереж, для розв'язання задач компонування.

Allen та ін. [134] запропонували спеціальний змішаний алгоритм для ортогональної упаковки прямокутних паралелепіпедів (блоків) в прямокутний контейнер мінімальної довжини, ширина і висота якого фіксовані. Метод складається з двох основних етапів: евристичний і метаевристичний алгоритми. На початковому етапі за допомогою евристичного алгоритму виконується швидка упаковка певної кількості блоків (three-dimensional best fit). Метаевристичний етап служить для упаковки частини блоків, що залишилися. Як показано, така комбінація покращує отримувані розв'язки порівняно з розв'язками, що отримуються тільки за допомогою евристичного етапу.

Peng та ін. [135] навели змішаний алгоритм для розв'язання задачі завантаження контейнера. Допускається тільки ортогональна упаковка блоків в контейнер. Спочатку автори вводять основний евристичний алгоритм, що полягає в отриманні допустимого розв'язку. Водночас використовується спеціальна структура, що називається послідовністю упаковки і служить для вибору завантажуваного блоку на кожному етапі упаковки. Далі розглядається алгоритм імітаційного відпалу для пошуку наближення до оптимального розв'язку.

Eisenbrand та ін. [136] з метою розрахунку місткості багажників автомобілів запропонували новий підхід до розв'язання задачі розміщення максимальної кількості прямокутних паралелепіпедів в область складної форми. Цей підхід поєднує в собі комбінаторний метод, що ґрунтується на сітках, і метод імітаційного відпалу. Відмітно, що паралелепіпеди допускають довільні повороти.

Детальний огляд методів розв'язання задач C&P наведений в дисертаційній роботі О. Панкратова [137].

1.3 Огляд методів геометричного проектування

Детальний огляд історії та напряму розвитку теорії геометричного проектування наведений в [138].

Створення академіком НАН України В. Л. Рвачовим теорії R -функцій [139, 140] послужило фундаментом теорії геометричного проектування. R -функції дали можливість аналітично описувати об'єкти складної геометричної форми і використовувати математичний апарат класичного аналізу для їхнього дослідження.

Розвитку нового наукового напряму – оптимального розміщення геометричних об'єктів, створеного професором В. Л. Рвачовим [141–143], присвятив свої наукові дослідження професор Ю. Г. Стоян.

Для аналітичного опису умов торкання геометричних об'єктів в науковій школі професора Ю. Г. Стояна розроблений конструктивний математичний апарат функції щільного розміщення [144, 145]. Поняття годографа функції щільного розміщення, запропонованого в [146], є основою методології послідовно-одиничного розміщення. Цей підхід в [147, 148] використаний для знаходження наближення до локальних екстремумів в задачі нерегулярного розміщення об'єктів. Багатовимірність задачі викликала необхідність перебору отримуваних наближень [149].

Значну увагу в роботах професора Ю. Г. Стояна та його учнів приділено дослідженню й формалізації поняття геометричної інформації, дослідженню просторових форм і метричних характеристик геометричних об'єктів [150, 151].

У роботі [152] було введено поняття ϕ -об'єкта, який є алгебро-топологічною моделлю реальних об'єктів.

Унікальне поняття ϕ -функції [152] дозволило побудувати Ю. Г. Стояну математичну модель основної задачі геометричного

проектування, формалізувати відношення взаємного неперетину пари геометричних об'єктів і відношення належності розміщеного об'єкта області розміщення. Це визначення започаткувало дослідження з математичного моделювання задач розміщення.

Формулювання задачі розміщення як задачі математичного програмування стало можливим з появою апарату структур лінійних нерівностей [153], що дозволило використати розроблені у рамках теорії математичного програмування підходи як основу для розробки методів розв'язання. Застосування сучасних обчислювальних засобів розв'язування оптимізаційних задач теорії дослідження операцій до предметної області задач геометричного проектування стимулювало появу робіт, в яких, зокрема, розглядалися підходи до побудови аналітичних моделей задач розміщення неопуклих орієнтованих [154] і неорієнтованих [155, 156] багатокутників.

Паралельно з розвитком теорії і методів розв'язання двовимірних задач розміщення закладалися основи методології розв'язання тривимірних задач розміщення. У перших роботах наукової школи Ю. Г. Стояна розглядалася щільна упаковка паралелепіпедів в паралелепіпеді [157], [158]. У [159] був досліджений клас задач тривимірного періодичного розміщення. Задача розміщення опуклих багатогранників досліджена в [160]. Методи локальної оптимізації в задачах упаковки опуклих багатогранників наведені в [161, 162]. Для цих робіт важливе значення мали дослідження, присвячені аналітичному опису умов торкання багатогранників [163, 164]. В усіх перелічених вище роботах область розміщення розглядалася у вигляді односторонньо обмеженого прямокутного паралелепіпеда.

Останнім часом теорія ϕ -функцій набула істотного розвитку для об'єктів в R^3 [165–172]. На базі побудованих засобів математичного моделювання відношень між геометричними об'єктами побудовані

математичні моделі і запропоновані методи розв'язання тривимірних оптимізаційних задач упаковки [173–181].

Так, у дисертаційній роботі [81] побудовані: квазі ρ -функції і псевдонормалізовані квазі ρ -функції для конусів, усічених конусів, сфероконусів, циліндрів, сфероциліндрів, сферичних сегментів, сферичних дисків, куль і багатогранників, що допускають неперервні трансляції і повороти; запропонована математична модель задача упаковки позначених вище опуклих тривимірних геометричних об'єктів з урахуванням мінімально допустимих відстаней; розроблена модифікація методу отримання допустимих початкових точок, що ґрунтується на гомотетичних перетвореннях об'єктів; запропонований метод локальної оптимізації; розроблена модифікація детермінованого алгоритму переходу від одного локального екстремуму до іншого.

У роботі [182] розроблено підхід до упаковки опуклих неорієнтованих тривимірних тіл та проведено чисельні розрахунки; в [183, 184, 193, 194, 197] запропоновано математичні моделі та методи розв'язання різних реалізацій базової математичної моделі задачі упаковки тривимірних тіл; в [185, 186] побудовано ρ -функції для відповідних пар тривимірних тіл; в [187, 189, 195, 196, 198] здійснено постановку задачі, запропоновано підходи до побудови ρ -функції та квазі ρ -функцій і досліджено їхні властивості; в [188, 200] побудовано нормалізовану ρ -функцію для циліндра та сфероциліндра, сфероциліндра та кулі; в [190, 192] запропоновано модифікацію методу локальної оптимізації; в [191] розроблено метод перебору локальних екстремумів в задачі пакування тривимірних тіл; в [199] запропоновано методологію розв'язання задачі упаковки тривимірних неорієнтованих тіл; в [200, 201, 202] запропоновано підхід до розв'язання задач упаковки багатогранників, який ґрунтується на використанні ρ -функцій.

1.4 Висновки по розділу

Аналіз наукової літератури за темою дисертації показав, що як розміщені об'єкти, як правило, виступають або кулі, або прямокутні паралелепіпеди. Публікації, присвячені розв'язанню задач оптимізації упаковки багатогранників, зустрічаються у край рідко, і до того ж багатогранники переважно є опуклими. У більшості робіт повороти розміщуваних багатогранників або не дозволяються, або можуть бути здійснені тільки з певним дискретним кроком. Методології розв'язання зазвичай використовують евристики.

Найменш дослідженими є задачі упаковки довільних неопуклих багатогранників, які разом з неперервними трансляціями допускають неперервні повороти, враховують мінімально допустимі відстані та обмеження балансу.

Найбільш потужним засобом аналітичного моделювання відношень геометричних об'єктів є метод ρ -функцій. Нині побудовані ρ -функції для деяких 3D об'єктів. Деякі з ρ -функцій (зокрема, для багатогранників) досить складні, що робить неефективним використання NLP-solvers для пошуку локальних екстремумів. Проте побудувати ρ -функції і псевдонормалізовані ρ -функції для довільних багатогранників не вдалося.

Тому виникає необхідність в розробці нових засобів математичного та комп'ютерного моделювання процесу оптимізації упаковки довільних багатогранників, що допускають неперервні трансляції і повороти, з урахуванням мінімально допустимих відстаней і обмежень балансу.

Список джерел, які використано у даному розділі наведено у повному списку використаних джерел [27, 28, 51 – 202]

2 ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

В цьому розділі формулюються основні положення (в рамках теорії геометричного проектування), необхідні для побудови засобів математичного і комп'ютерного моделювання відношень між геометричними об'єктами (зокрема, багатогранниками), що виникають в задачах оптимального розміщення. Розглядаються поняття ϕ -функції, нормалізованої та псевдонормалізованої ϕ -функції, квазі ϕ -функції, нормалізованої та псевдонормалізованої квазі ϕ -функції, наводяться їх основні властивості. Обґрунтовується необхідність побудови різних класів ϕ -функцій і квазі ϕ -функцій, необхідних для побудови математичних моделей задач оптимального розміщення довільних (опуклих і неопуклих) багатогранників з урахуванням обмежень на мінімально допустимі відстані. Визначаються типи контейнерів, описуються багатогранники, як об'єкти розміщення, визначаються обмеження розміщення та балансу. Формулюється постановка базової задачі оптимальної упаковки довільних багатогранників.

2.1 Математичне моделювання в геометричному проектуванні

2.1.1 Φ -функція, нормалізована та псевдонормалізована ϕ -функція.

У класі задач геометричного проектування як математичних моделей матеріальних об'єктів розглядаються ϕ -об'єкти [152] – непорожні точкові множини $A \subset R^3$, які задовольняють такі вимоги: A канонічно замкнута множина [203]; внутрішність і замикання множини A мають один і той же гомотопічний тип [204]; в будь-якій точці

$x \in \text{cl } A$ існує окіл $U_x \subset \text{cl } A$, такий, що $\text{int } U_x$ і $\text{cl } U_x$ мають один і той же гомотопічний тип.

Подання інформації про рhі-об'єкти, як про математичні моделі матеріальних об'єктів в задачах геометричного проектування пов'язано з поняттям геометричної інформації [150], введеної як сукупність трьох елементів: просторової форми рhі-об'єкта, що характеризує конфігурацію об'єкта; метричних характеристик, що визначають "розміри" рhі-об'єкта; параметрів розміщення, які задають місце розташування рhі-об'єкта у відповідному просторі.

Нехай $A \subset R^3$ і $B \subset R^3$ – замкнуті рhі-об'єкти. Вважаємо, що, принаймні, один з об'єктів обмежений. Відповідно місце розташування об'єкта A визначається вектором змінних параметрів розміщення (v_A, θ_A) , де: $v_A = (x_A, y_A, z_A)$ – вектор трансляції, $\theta_A = (\theta^1, \theta^2, \theta^3)$ – кути Ейлера. Позначимо $u_A = (v_A, \theta_A, \lambda_A)$ вектор змінних об'єкта A , $u_B = (v_B, \theta_B, \lambda_B)$ вектор змінних об'єкта B .

Надалі, об'єкт A , повернений на кути $\theta_1, \theta_2, \theta_3$, трансльований на вектор v позначимо як $A(u_A) = \{p \in R^3 : p = v_A + M(\theta_A) \cdot p^0, \forall p^0 \in A^0\}$, де A^0 позначає нетрансльований і неповернутий об'єкт A , $M(\theta_A) = M(\theta^1, \theta^2, \theta^3)$ – стандартна матриця повороту, що має вигляд:

$$\begin{pmatrix} \cos \theta^1 \cos \theta^3 - \sin \theta^1 \cos \theta^2 \sin \theta^3 & -\cos \theta^1 \sin \theta^3 - \sin \theta^1 \cos \theta^2 \cos \theta^3 & \sin \theta^1 \sin \theta^2 \\ \sin \theta^1 \cos \theta^3 + \cos \theta^1 \cos \theta^2 \sin \theta^3 & -\sin \theta^1 \sin \theta^3 + \cos \theta^1 \cos \theta^2 \cos \theta^3 & -\cos \theta^1 \sin \theta^2 \\ \sin \theta^2 \sin \theta^3 & \sin \theta^2 \cos \theta^3 & \cos \theta^2 \end{pmatrix},$$

Однією з найважливіших задач під час моделювання розміщення об'єктів є побудова такої функції, яка б описувала відношення між парою рhі-об'єктів $A(u_A)$ і $B(u_B)$.

Означена функція має враховувати такі ситуації:

- об'єкти $A(u_A)$ і $B(u_B)$ не мають спільних точок, тобто $clA(u_A) \cap clB(u_B) = \emptyset$, де $cl(\bullet)$ – замикання об'єкта (\bullet) ;
- об'єкти $A(u_A)$ і $B(u_B)$ перетинаються, тобто $int A(u_A) \cap int B(u_B) \neq \emptyset$, де $int(\bullet)$ – внутрішність об'єкта (\bullet) ;
- об'єкти $A(u_A)$ і $B(u_B)$ дотикаються, тобто $frA(u_A) \cap frB(u_B) \neq \emptyset$ та $int A(u_A) \cap int B(u_B) = \emptyset$, де $fr(\bullet)$ – границя об'єкта (\bullet) .

Наведені вище вимоги задовольняє ϕ -функція, введена в [152]. Визначення ϕ -функції та її властивості сформульовані в роботах [205, 206].

Визначення 2.1. Неперервна, усюди визначена функція $\Phi^{AB}(u_A, u_B)$ називається ϕ -функцією об'єктів $A(u_A)$ і $B(u_B)$, якщо вона задовольняє такі властивості:

$$\begin{aligned} \Phi^{AB}(u_A, u_B) < 0, \text{ якщо } int A(u_A) \cap int B(u_B) \neq \emptyset, \\ \Phi^{AB}(u_A, u_B) = 0, \text{ якщо } int A(u_A) \cap int B(u_B) = \emptyset \text{ і} \\ frA(u_A) \cap frB(u_B) \neq \emptyset, \\ \Phi^{AB}(u_A, u_B) > 0, \text{ якщо } A(u_A) \cap B(u_B) = \emptyset, \end{aligned} \quad (2.1)$$

Тут $fr A$ позначає границю, а $int A$ – внутрішність об'єкта A .

На рис. 2.1 зображені три випадки, які розрізняє ϕ -функція.

З метою моделювання обмежень на допустимі відстані між геометричними об'єктами введено поняття нормалізованої ϕ -функції [152, 205, 206].

Нехай задано обмеження на мінімально допустиму відстань ρ між об'єктами $A(u_A)$ і $B(u_B)$, тобто $\text{dist}(A(u_A), B(u_B)) \geq \rho$, де $\text{dist}(A(u_A), B(u_B)) = \min_{a \in A, b \in B} \text{dist}(a, b)$ – евклідова відстань між замкнутими множинами $A(u_A)$ і $B(u_B)$, $\text{dist}(a, b) = \sqrt{(x_a^2 - x_b^2) + (y_a^2 - y_b^2) + (z_a^2 - z_b^2)}$ – евклідова відстань між точками a і b в \mathbb{R}^3 .

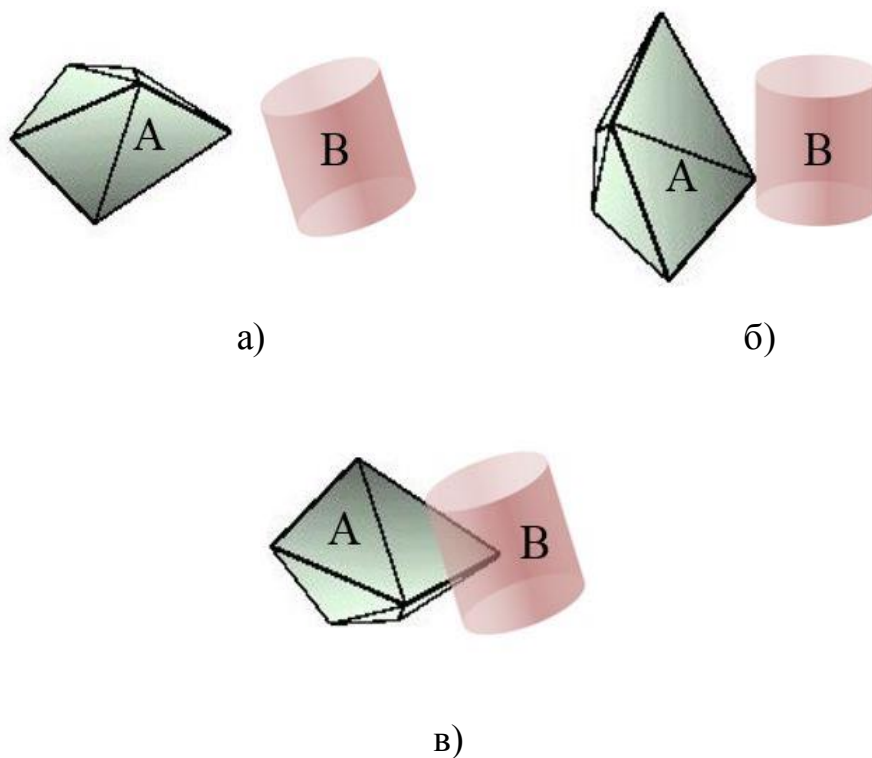


Рисунок 2.1 – Ілюстрація к визначенню ϕ -функції:

а) $\Phi^{AB}(u_A, u_B) > 0$; б) $\Phi^{AB}(u_A, u_B) = 0$; в) $\Phi^{AB}(u_A, u_B) < 0$.

Визначення 2.2. Φ -функція $\tilde{\Phi}^{AB}(u_A, u_B)$ називається нормалізованою, якщо її значення дорівнюють евклідовим відстаням між об'єктами $A(u_A)$ і $B(u_B)$, за умови $\text{int } A(u_A) \cap \text{int } B(u_B) = \emptyset$.

Тобто

$$\tilde{\Phi}^{AB}(u_A, u_B) = \text{dist}(A(u_A), B(u_B)) \Leftrightarrow \text{int } A(u_A) \cap \text{int } B(u_B) = \emptyset, \quad (2.2)$$

У термінах ρ -функцій обмеження $\text{dist}(A(u_A), B(u_B)) \geq \rho$ еквівалентне $\tilde{\Phi}^{AB}(u_A, u_B) \geq \rho$, де $\tilde{\Phi}^{AB}$ – нормалізована ρ -функція об'єктів $A(u_A)$ і $B(u_B)$, $u_A = (x_A, y_A, z_A, \theta_A, \lambda_A)$ і $u_B = (x_B, y_B, z_B, \theta_B, \lambda_B)$.

Побудова нормалізованих ρ -функцій для довільних ρ -об'єктів – досить складна процедура. Через це для моделювання мінімально допустимих відстаней між об'єктами пропонується використовувати вільні від радикалів псевдонормалізовані ρ -функції [205, 206].

Визначення 2.3. Неперервна всюди визначена функція $\hat{\Phi}^{AB}(u_A, u_B)$ називається псевдонормалізованою ρ -функцією об'єктів $A(u_A)$ і $B(u_B)$, якщо виконуються такі властивості:

$$\begin{aligned} \hat{\Phi}^{AB}(u_A, u_B) &> 0, \text{ якщо } \text{dist}(A(u_A), B(u_B)) > \rho, \\ \hat{\Phi}^{AB}(u_A, u_B) &= 0, \text{ якщо } \text{dist}(A(u_A), B(u_B)) = \rho, \\ \hat{\Phi}^{AB}(u_A, u_B) &< 0, \text{ якщо } \text{dist}(A(u_A), B(u_B)) < \rho. \end{aligned} \quad (2.3)$$

Зокрема, з (2.3) випливає

$$\text{dist}(A(u_A), B(u_B)) > \rho \Leftrightarrow \tilde{\Phi}^{AB}(u_A, u_B) \geq \rho \Leftrightarrow \hat{\Phi}^{AB}(u_A, u_B) \geq 0,$$

$$\tilde{\Phi}^{AB}(u_A, u_B) = \rho \Leftrightarrow \hat{\Phi}^{AB}(u_A, u_B) = 0.$$

На рис. 2.2 зображені три випадки, які розрізняє псевдонормалізована рхі-функція.

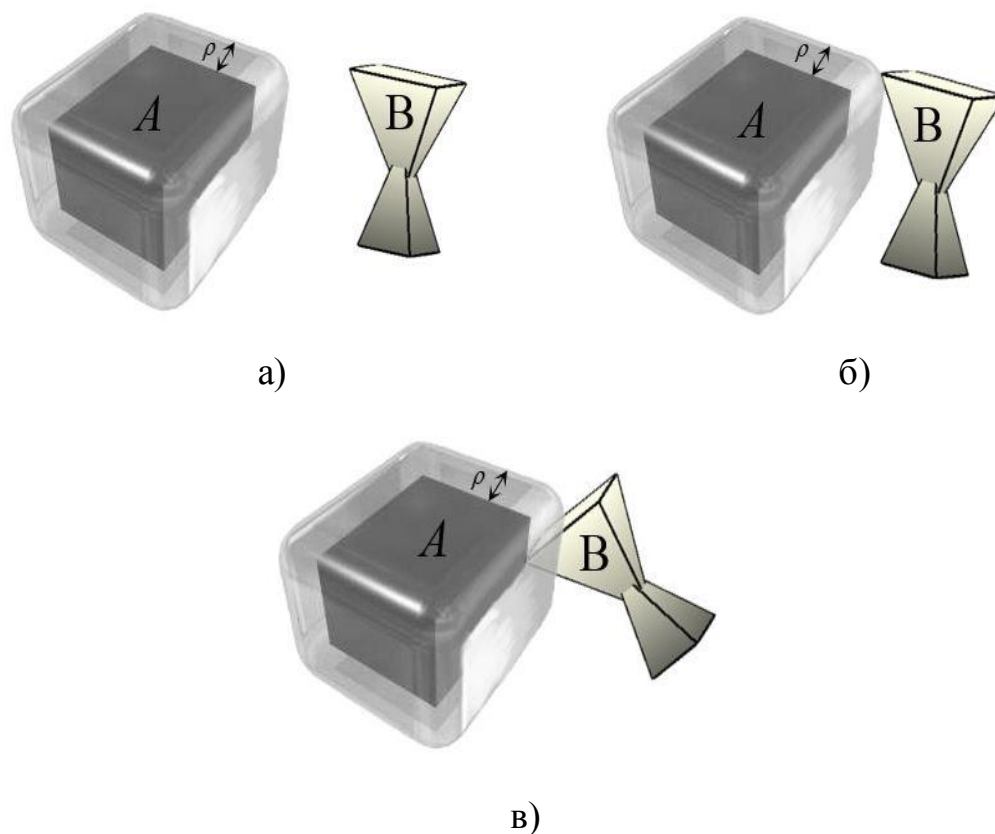


Рисунок 2.2 – Ілюстрація к визначенню псевдонормалізованої рхі-функції: а) $\widehat{\Phi}^{AB}(u_A, u_B) > 0$; б) $\widehat{\Phi}^{AB}(u_A, u_B) = 0$; в) $\widehat{\Phi}^{AB}(u_A, u_B) < 0$.

З (2.2) і (2.3) випливає, що функція $\tilde{\Phi}^{AB}(u_A, u_B) - \rho$ є псевдонормалізованою рхі-функцією.

Нехай $A(u_A) = \bigcup_{i=1}^{N_1} A_i(u_A)$, $B(u_B) = \bigcup_{j=1}^{N_2} B_j(u_B)$, а $\Phi^{A_i B_j}(u_A, u_B)$ є

рхі-функцією для об'єктів $A_i(u_A)$ і $B_j(u_B)$, $i = 1, 2, \dots, N_1$, $j = 1, 2, \dots, N_2$.

Тоді функція, подана у вигляді

$$\Phi^{AB}(u_A, u_B) = \min\{\Phi^{A_i B_j}(u_A, u_B), i = 1, 2, \dots, N_1, j = 1, 2, \dots, N_2\}, \quad (2.4)$$

є ϕ -функцією для об'єктів $A(u_A)$ і $B(u_B)$, а функція

$$\widehat{\Phi}^{AB}(u_A, u_B) = \min\{\widehat{\Phi}^{A_i B_j}(u_A, u_B), i=1, 2, \dots, N_1, j=1, 2, \dots, N_2\},$$

є псевдонормалізованою ϕ -функцією для об'єктів $A(u_A)$ і $B(u_B)$, або ϕ -функцією для об'єктів $\widehat{A}(u_A)$ і $B(u_B)$, де $\widehat{A}(0) = A(0) \oplus S(0)$, $S(0)$ – куля радіусу ρ .

2.1.2 Квазі ϕ -функція, нормалізована та псевдонормалізована квазі ϕ -функція.

Для математичного моделювання відношень ϕ -об'єктів використовуються також квазі ϕ -функції [137,207].

Визначення 2.4. Квазі ϕ -функцією для ϕ -об'єктів $A(u_A)$ і $B(u_B)$ називається всюди визначена, неперервна функція $\Phi'^{AB}(u_A, u_B, u')$, для якої функція $\max_{u' \in U} \Phi'^{AB}(u_A, u_B, u')$ є ϕ -функцією для $A(u_A)$ і $B(u_B)$, де вид множини $U \subset R^n$ і розмірність простору R^n залежать від форми об'єктів $A(u_A)$ і $B(u_B)$.

Квазі ϕ -функція $\Phi'^{AB}(u_A, u_B, u')$ має низку важливих властивостей:

1. Якщо $\Phi'^{AB}(u_A, u_B, u') \geq 0$ для деяких u' , то $\text{int } A(u_A) \cap \text{int } B(u_B) = \emptyset$.

2. Якщо $\Phi^{AP}(u_A, u_P)$ – ϕ -функція для $A(u_A)$ і $P(u_P)$, $\Phi^{BP^*}(u_B, u_P)$ – ϕ -функція для $B(u_B)$ і $P^*(u_P) = R^3 \setminus \text{int } P(u_P)$, де

$P(u_P) = \{(x, y, z) : \psi_P = \alpha \cdot x + \beta \cdot y + \gamma \cdot z + \mu_P \geq 0\}$, $u_P = (\theta_{xP}, \theta_{yP}, \mu_P)$,
 $\alpha = \sin \theta_{yP}$, $\beta = -\sin \theta_{xP} \cdot \cos \theta_{yP}$, $\gamma = \cos \theta_{xP} \cdot \cos \theta_{yP}$, то функція,
визначена у вигляді

$$\Phi'^{AB}(u_A, u_B, u_P) = \min\{\Phi^{AP}(u_A, u_P), \Phi^{BP^*}(u_B, u_P)\}, \quad (2.5)$$

є квазі рhі-функцією для пари обмежених $A(u_A)$ і $B(u_B)$. Тут $u' = u_P$.

3. Якщо $\Phi'^{AP}(u_A, u_P, u'_1)$ – квазі рhі-функція для $A(u_A)$ і $P(u_P)$, $\Phi'^{BP^*}(u_B, u_P, u'_2)$ – квазі рhі-функція для $B(u_B)$ і $P^*(u_P)$, то функція вигляду

$$\Phi'^{AB}(u_A, u_B, u') = \min\{\Phi'^{AP}(u_A, u_P, u'_1), \Phi'^{BP^*}(u_B, u_P, u'_2)\} \quad (2.6)$$

є квазі рhі-функцією для обмежених об'єктів $A(u_A)$ і $B(u_B)$. Тут $u' = (u_P, u'_1, u'_2)$.

Поняття квазі рhі-функції може бути використано також для моделювання обмежень на допустимі відстані між об'єктами.

З цією метою розглянемо визначення нормалізованої і псевдонормалізованої квазі рhі-функції, ґрунтуючись на аналогічних термінах для рhі-функцій.

Нехай $\rho > 0$ – задана мінімально допустима відстань між об'єктами $A(u_A)$ і $B(u_B)$.

Визначення 2.5. Квазі рhі-функція $\tilde{\Phi}'^{AB}(u_A, u_B, u')$ називається нормалізованою для об'єктів $A(u_A)$ і $B(u_B)$, якщо функція $\max_{u' \in U} \tilde{\Phi}'^{AB}(u_A, u_B, u')$ є нормалізованою рhі-функцією.

Таким чином,

$$\max_{u' \in U} \tilde{\Phi}'^{AB} \geq \rho \Leftrightarrow \text{dist}(A(u_A), B(u_B)) \geq \rho,$$

$$\tilde{\Phi}'^{AB} \geq \rho \Rightarrow \text{dist}(A(u_A), B(u_B)) \geq \rho.$$

Визначення 2.6. Функція $\hat{\Phi}'^{AB}(u_A, u_B, u')$ називається псевдонормалізованою квазі ρ -функцією для об'єктів $A(u_A)$ і $B(u_B)$, якщо функція $\max_{u' \in U} \hat{\Phi}'^{AB}(u_A, u_B, u')$ є псевдонормалізованою ρ -функцією.

Тоді

$$\max_{u' \in U} \hat{\Phi}'^{AB} \geq 0 \Leftrightarrow \text{dist}(A(u_A), B(u_B)) \geq \rho,$$

$$\hat{\Phi}'^{AB} \geq 0 \Rightarrow \text{dist}(A(u_A), B(u_B)) \geq \rho.$$

Нехай квазі ρ -функція має вигляд

$$\Phi'^{AB}(u_A, u_B, u_P) = \min\{\tilde{\Phi}^{AP}(u_A, u_P), \tilde{\Phi}^{BP^*}(u_B, u_P)\}, \quad (2.7)$$

де $\tilde{\Phi}^{AP}(u_A, u_P)$, $\tilde{\Phi}^{BP^*}(u_B, u_P)$ – нормалізовані ρ -функції.

Тоді квазі ρ -функція

$$\tilde{\Phi}'^{AB}(u_A, u_B, u_P) = 2\Phi'^{AB}(u_A, u_B, u_P)$$

є нормалізованою квазі ρ -функцією, а квазі ρ -функція

$$\hat{\Phi}'^{AB}(u_A, u_B, u_P) = \Phi'^{AB}(u_A, u_B, u_P) - 0.5\rho,$$

є псевдонормалізованою квазі рhі-функцією для об'єктів $A(u_A)$ і $B(u_B)$.

$$\text{Нехай} \quad A(u_A) = \bigcup_{i=1}^{N_1} A_i(u_A), \quad B(u_B) = \bigcup_{j=1}^{N_2} B_j(u_B), \quad \text{а}$$

$\tilde{\Phi}^{A_i B_j}(u_A, u_B, u'_{ij})$ є нормалізованою квазі рhі-функцією для об'єктів $A_i(u_A)$ і $B_j(u_B)$, $i=1, 2, \dots, N_1$, $j=1, 2, \dots, N_2$.

Тоді функція, подана у вигляді

$$\tilde{\Phi}^{AB}(u_A, u_B, u') = \min\{\tilde{\Phi}^{A_i B_j}(u_A, u_B, u'_{ij}), i=1, 2, \dots, N_1, j=1, 2, \dots, N_2\}, \quad (2.8)$$

є нормалізованою квазі рhі-функцією для об'єктів $A(u_A)$ і $B(u_B)$, а функція

$$\hat{\Phi}^{AB}(u_A, u_B, u') = \min\{\hat{\Phi}^{A_i B_j}(u_A, u_B, u'_{ij}), i=1, 2, \dots, N_1, j=1, 2, \dots, N_2\}$$

є псевдонормалізованою квазі рhі-функцією для об'єктів $A(u_A)$ і $B(u_B)$, або квазі рhі-функцією для об'єктів $\hat{A}(u_A)$ і $B(u_B)$, де $\hat{A}(0) = A(0) \oplus S(0)$, $S(0)$ – куля радіусу ρ , $u' = (u'_{ij}, i=1, 2, \dots, N_1, j=1, 2, \dots, N_2)$.

2.2 Постановка базової задачі оптимальної упаковки багатогранників (ОРР)

Розглядається задача упаковки в такій постановці.

Контейнери. Нехай $\Omega = \Omega(\rho)$ визначає опуклий контейнер зі змінними метричними характеристиками,

$$\Omega = \{(x, y, z, \rho) \in R^3 : \Psi(x, y, z, \rho) \geq 0\},$$

заданий в глобальній системі координат OXYZ, де

$$\Psi(x, y, z, p) = \min\{\Psi_t(x, y, z, p), t = 1, \dots, n_\Omega\},$$

а функції $\Psi_t(x, y, z, p)$ є диференційованими, $t = 1, \dots, n_\Omega$.

Зокрема, розглядаємо такі форми контейнерів (рис. 2.3):

– *кубоїд*

$$\mathbf{B} = \{(x, y, z, l, w, h) \in R^3 \mid \min\{\Psi_t(x, y, z, l, w, h), t = 1, 2, \dots, 6\} \geq 0\},$$

де

$$\begin{aligned} \Psi_1(x, l) &= x + l, & \Psi_2(x, l) &= -x + l, & \Psi_3(y, w) &= y + w, \\ \Psi_4(y, w) &= -y + w, & \Psi_5(z, h) &= z + h, & \Psi_6(z, h) &= -z + h, \end{aligned}$$

з метричними характеристиками (l, w, h) , де h – напіввисота; w – напівширина та l – напівдовжина кубоїда \mathbf{B} , $p = (l, w, h)$;

– *куля*

$$\mathbf{S} = \{(x, y, z, r) \in R^3 \mid r^2 - x^2 - y^2 - z^2 \geq 0\},$$

з метричною характеристикою (r) , де r – радіус кулі \mathbf{S} , $p = (r)$;

– *прямий круговий циліндр*

$$\mathbf{C} = \{(x, y, z, \lambda) \in R^3 \mid \min\{\Psi_t(x, y, z, \lambda), t=1, 2, 3\} \geq 0\},$$

де

$$\Psi_1(x, y, z, \lambda) = (\lambda r)^2 - x^2 - y^2, \quad \Psi_2(x, y, z, \lambda) = -z + \lambda h,$$

$$\Psi_3(x, y, z, \lambda) = z + \lambda h$$

з радіусом λr і висотою λh , де λ – змінний коефіцієнт гомотетії, $p = (\lambda)$, тут $\lambda = 1$ відповідає початковим розмірам контейнера;

– *еліпсоїд*

$$\mathbf{E} = \{(x, y, z, \lambda) \in R^3 \mid \lambda^2 - \frac{x^2}{\lambda a^2} - \frac{y^2}{\lambda b^2} - \frac{z^2}{\lambda c^2} \geq 0\},$$

зі змінними напівосями $\lambda a, \lambda b, \lambda c$, $p = (\lambda)$, тут $\lambda = 1$ відповідає початковим розмірам контейнера.

– *опуклий багатогранник*

$$\mathbf{P} = \{(x, y, z, \lambda) \in R^3 \mid \min\{\Psi_t(x, y, z, \lambda), t=1, 2, \dots, n_{\mathbf{P}}\} \geq 0\},$$

де $\Psi_t = A_t x + B_t y + C_t z + \lambda D_t = 0$ – рівняння t -ї площини, $p = (\lambda)$, тут $\lambda = 1$ відповідає початковим розмірам контейнера;

– *довільна опукла область*

$$\mathbf{\Lambda} = \{(x, y, z, \lambda) \in R^3 \mid \min\{\Psi_t(x, y, z, \lambda), t=1, 2, \dots, n_{\mathbf{\Lambda}}\} \geq 0\},$$

границя якої формується циліндричною, сферичною, еліптичною

поверхнями або площиною, $p = (\lambda)$, тут $\lambda = 1$ відповідає початковим розмірам контейнера;

Відзначимо, що кожна з метричних характеристик контейнера Ω може бути змінною сама по собі, тобто r і h – для циліндра, або a, b, c – для еліпсоїда.

Розміщені об'єкти. Задано набір багатогранників \mathbb{Q}_q , $q \in J_N$, $J_N = \{1, 2, \dots, N\}$.

Кожен багатогранник \mathbb{Q}_q може бути або опуклим, або не опуклим.

В межах цього дослідження вважаємо, що маса M_q кожного багатогранника \mathbb{Q}_q відома.

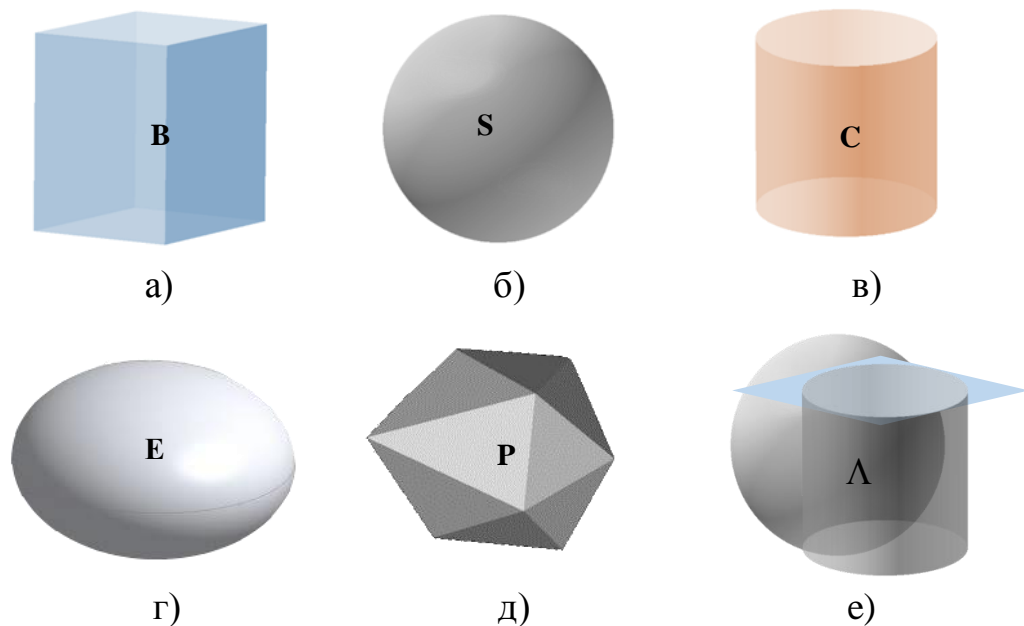


Рисунок 2.3 – Види контейнерів: а) – кубоїд; б) – куля; в) – прямий круговий циліндр; г) – еліпсоїд; д) – опуклий багатогранник; е) – опукла область, границя якої формується циліндричною, сферичною поверхнями та площиною

Із кожним з багатогранників \mathbb{Q}_q асоціюється його власна система координат з центром в точці v_q . Уявімо, що кожен неопуклий багатогранник \mathbb{Q}_q поданий у вигляді об'єднання опуклих багатогранників K_j^q , $j=1, \dots, n_q$, тобто

$$\mathbb{Q}_q(u_q) = \bigcup_{j=1}^{n_q} K_j^q(u_q).$$

З кожним опуклим багатогранником K_j^q асоціюється система координат багатогранника \mathbb{Q}_q (Рис. 2.4). Опуклий багатогранник K_j^q визначається його вершинами p_s^{qj} , $s=1, \dots, m_j^q$, в локальній системі координат неопуклого багатогранника \mathbb{Q}_q .

Вихідні дані, які використовуються для формування неопуклих багатогранників \mathbb{Q}_q , містяться в підрозділі 5.1.

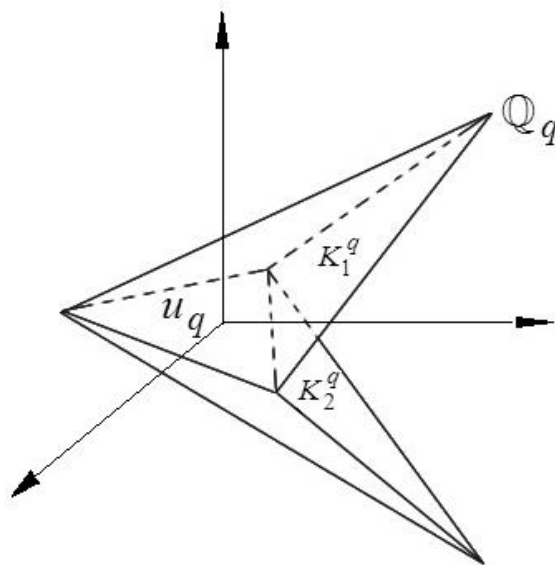


Рис. 2.4. Приклад неопуклого багатогранника $\mathbb{Q}_q = K_1^q \cup K_2^q$

В межах цього дослідження вважаємо, що представлення кожного багатогранника \mathbb{Q}_q у вигляді об'єднання опуклих багатогранників

$$\bigcup_{i=1}^{n_q} K_i^q \text{ відомо [208].}$$

Не втрачаючи загальності, вважаємо, що центр v_q багатогранника \mathbb{Q}_q збігається з центром описаної навколо нього сфери S_q радіуса r_q . Для побудови описаної навколо багатогранника сфери використовується алгоритм, який обчислює найменшу охоплюючу сферу для набору точок (<https://github.com/hbf/miniball>).

Положення та орієнтація кожного багатогранника \mathbb{Q}_q визначається вектором $u_q = (v_q, \theta_q)$ його змінних параметрів розміщення. Тут $v_q = (x_q, y_q, z_q)$ – вектор трансляції, $\theta_q = (\theta_q^1, \theta_q^2, \theta_q^3)$ – вектор параметрів обертання, де $\theta_q^1, \theta_q^2, \theta_q^3$ – кути Ейлера.

Багатогранник, повернутий на кути $\theta_q^1, \theta_q^2, \theta_q^3$ і трансльований на вектор v_q , позначається через

$$\mathbb{Q}_q(u_q) = \{p_q \in R^3 : p_q = v_q + M(\theta_q) \cdot p_q^0, p_q^0 \in \mathbb{Q}_q^0\}, \text{ де } u_q = (v_q, \theta_q),$$

\mathbb{Q}_q^0 визначає неповернутий і нетрансльований багатогранник $\mathbb{Q}_q(u_q)$,

$M(\theta_q)$ – матриця повороту вигляду

$$M(\theta_q^1, \theta_q^2, \theta_q^3) = \begin{pmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{pmatrix},$$

$$m_{11} = \cos \theta_q^1 \cos \theta_q^3 - \sin \theta_q^1 \cos \theta_q^2 \sin \theta_q^3,$$

$$\begin{aligned}
m_{12} &= -\cos \theta_q^1 \sin \theta_q^3 - \sin \theta_q^1 \cos \theta_q^2 \cos \theta_q^3, \\
m_{13} &= \sin \theta_q^1 \sin \theta_q^2, \quad m_{21} = \sin \theta_q^1 \cos \theta_q^3 + \cos \theta_q^1 \cos \theta_q^2 \sin \theta_q^3, \\
m_{22} &= -\sin \theta_q^1 \sin \theta_q^3 + \cos \theta_q^1 \cos \theta_q^2 \cos \theta_q^3, \quad m_{23} = -\cos \theta_q^1 \sin \theta_q^2, \\
m_{31} &= \sin \theta_q^2 \sin \theta_q^3, \quad m_{32} = \sin \theta_q^2 \cos \theta_q^3, \quad m_{33} = \cos \theta_q^2.
\end{aligned}$$

Кожна точка $p_q = (p_x^q, p_y^q, p_z^q)$ після обертання і трансляції багатогранника $\mathbb{Q}_q(u_q)$ матиме вигляд

$$\begin{aligned}
p_x^q &= x_q + (\sin \theta_q^1 \cos \theta_q^3 + \cos \theta_q^1 \cos \theta_q^2 \sin \theta_q^3) \cdot p_x^0 + \\
&+ (-\sin \theta_q^1 \sin \theta_q^3 + \cos \theta_q^1 \cos \theta_q^2 \cos \theta_q^3) \cdot p_y^0 + (-\cos \theta_q^1 \sin \theta_q^2) \cdot p_z^0, \\
p_y^q &= y_q + (\cos \theta_q^1 \cos \theta_q^3 - \sin \theta_q^1 \cos \theta_q^2 \sin \theta_q^3) \cdot p_x^0 + \\
&+ (-\cos \theta_q^1 \sin \theta_q^3 - \sin \theta_q^1 \cos \theta_q^2 \cos \theta_q^3) \cdot p_y^0 + (\sin \theta_q^1 \sin \theta_q^2) \cdot p_z^0, \\
p_z^q &= z_q + (\sin \theta_q^2 \sin \theta_q^3) \cdot p_x^0 + (\sin \theta_q^2 \cos \theta_q^3) \cdot p_y^0 + (\cos \theta_q^2) \cdot p_z^0
\end{aligned}$$

Зокрема, вершина p_s^q багатогранника $\mathbb{Q}_q(u_q)$ матиме вигляд

$$\begin{aligned}
p_{xs}^q &= x_q + (\sin \theta_q^1 \cos \theta_q^3 + \cos \theta_q^1 \cos \theta_q^2 \sin \theta_q^3) \cdot p_{xs}^0 + \\
&+ (-\sin \theta_q^1 \sin \theta_q^3 + \cos \theta_q^1 \cos \theta_q^2 \cos \theta_q^3) \cdot p_{ys}^0 + (-\cos \theta_q^1 \sin \theta_q^2) \cdot p_{zs}^0, \\
p_{ys}^q &= y_q + (\cos \theta_q^1 \cos \theta_q^3 - \sin \theta_q^1 \cos \theta_q^2 \sin \theta_q^3) \cdot p_{xs}^0 + \\
&+ (-\cos \theta_q^1 \sin \theta_q^3 - \sin \theta_q^1 \cos \theta_q^2 \cos \theta_q^3) \cdot p_{ys}^0 + (\sin \theta_q^1 \sin \theta_q^2) \cdot p_{zs}^0, \\
p_{zs}^q &= z_q + (\sin \theta_q^2 \sin \theta_q^3) \cdot p_{xs}^0 + (\sin \theta_q^2 \cos \theta_q^3) \cdot p_{ys}^0 + (\cos \theta_q^2) \cdot p_{zs}^0,
\end{aligned}$$

$$s = 1, \dots, m_i, i \in I_{n_q}. \quad (2.9)$$

Обмеження. Між кожною парою багатогранників $\mathbb{Q}_q(u_q)$ і $\mathbb{Q}_g(u_g)$, $q < g \in J_N$, також як і між багатогранником $\mathbb{Q}_q(u_q)$, $q \in J_N$, та границею контейнера Ω можуть бути задані мінімально допустимі відстані $\rho_{qg} > 0$ і $\rho_q > 0$.

Це означає, що кожен багатогранник $\mathbb{Q}_q(u_q)$ має бути розташований по відношенню до багатогранника $\mathbb{Q}_g(u_g)$ не ближче, ніж задана допустима відстань ρ_{qg} , а також, кожен багатогранник $\mathbb{Q}_q(u_q)$ має бути розташований всередині контейнера Ω і не ближче, ніж задана допустима відстань $\rho_q > 0$ до границі контейнера Ω , де

$$\text{dist}(\mathbb{Q}_q(u_q), \mathbb{Q}_g(u_g)) = \min_{a \in \mathbb{Q}_q, b \in \mathbb{Q}_g} \rho(a, b).$$

Відзначимо, що мінімально допустима відстань між кожною парою опуклих багатогранників $K_i^q(u_q) \subset \mathbb{Q}_q(u_q)$, $i = 1, \dots, n_q$ і $K_j^g(u_g) \subset \mathbb{Q}_g(u_g)$, $j = 1, \dots, n_g$, $q < g \in J_N$, збігається з заданою допустимою відстанню між вихідними багатогранниками $\mathbb{Q}_q(u_q)$ і $\mathbb{Q}_g(u_g)$. Більш того, мінімально допустима відстань між кожним багатогранником K_i^q , $i = 1, \dots, n_q$, $q \in J_N$, та границею контейнера Ω збігається з заданою допустимою відстанню між вихідним багатогранником $\mathbb{Q}_q(u_q)$, $q \in J_N$, та границею контейнера Ω .

Розглядаються такі види обмежень:

а) обмеження розміщення (*arrangement constraints*), які поділяються на

1) обмеження на мінімально допустимі відстані (*distance constraints*) між багатогранниками $\mathbb{Q}_q(u_q)$ та $\mathbb{Q}_g(u_g)$:

$$\text{dist}(\mathbb{Q}_q(u_q), \mathbb{Q}_g(u_g)) \leq \rho_{qg}, \quad q < g \in J_N,$$

або обмеження неперетину (*non-overlapping constraints*) (у разі якщо обмеження на мінімально допустимі відстані між багатогранниками не задані), у вигляді

$$\text{int } \mathbb{Q}_q(u_q) \cap \text{int } \mathbb{Q}_g(u_g) = \emptyset, \quad q < g \in J_N;$$

2) обмеження включення (*distance containment constraints*) з урахуванням допустимих відстаней

$$\text{dist}(\mathbb{Q}_q(u_q), \Omega^*) \leq \rho_q, \quad q \in J_N, \quad \Omega^* = \mathbb{R}^3 \setminus \text{int } \Omega,$$

або обмеження включення (*containment constraints*) (у разі якщо обмеження на мінімально допустимі відстані між багатогранником та границею контейнера не задані), у вигляді

$$\mathbb{Q}_q(u_q) \subset \Omega \Leftrightarrow \text{int } \mathbb{Q}_q(u_q) \cap \Omega^* = \emptyset, \quad q \in J_N;$$

б) обмеження балансу (*stability constraint*)

1) відхилення центра мас системи $\Omega_{\mathbb{Q}}$ (контейнер $\Omega(p)$) з упакованими в ньому багатогранниками $\mathbb{Q}_q(u_q)$, $q \in J_N$) від заданої точки $p_e = (x_e, y_e, z_e)$, що має не перевищувати заданого допустимого значення.

Обмеження балансу будемо визначати так:

$$\mu(p, u) = \min\{\mu_1(p, u), \mu_2(p, u), \mu_3(p, u)\} \geq 0, \quad (2.10)$$

де

$$\mu_1(p, u) = \min\{-(x_s - x_e) + \Delta x_e, (x_s - x_e) + \Delta x_e\},$$

$$\mu_2(p, u) = \min\{-(y_s - y_e) + \Delta y_e, (y_s - y_e) + \Delta y_e\},$$

$$\mu_3(p, u) = \min\{-(z_s - z_e) + \Delta z_e, (z_s - z_e) + \Delta z_e\},$$

$$x_s = \frac{\sum_{q=0}^N M_q \hat{x}_q}{\sum_{q=0}^N M_q}, \quad y_s = \frac{\sum_{q=0}^N M_q \hat{y}_q}{\sum_{q=0}^N M_q}, \quad z_s = \frac{\sum_{q=0}^N M_q \hat{z}_q}{\sum_{q=0}^N M_q},$$

де $u = (u_1, \dots, u_q, \dots, u_N)$

x_s, y_s, z_s – координати центра мас O_s системи $\Omega_{\mathbb{Q}}$;

$\Delta x_e, \Delta y_e, \Delta z_e$ – допустимі відхилення від координат деякої заданої точки p_e .

$\hat{x}_q, \hat{y}_q, \hat{z}_q$ – центр мас неопуклого багатогранника $\mathbb{Q}_q(u_q)$.

Точка $(\hat{x}_q, \hat{y}_q, \hat{z}_q)$ центру мас неопуклого багатогранника $\mathbb{Q}_q(u_q)$ розраховується як зважена сума точок центрів мас опуклих багатогранників, що формують розбиття неопуклого багатогранника.

Задаємо множину V вершин v_i опуклого багатогранника K . Нехай V_j – множина вершин з V , які інцидентні j -ій грані опуклого багатогранника K . Точку центру мас опуклого багатогранника $\mathbb{Q}_q(u_q)$

шукаємо, як зважену суму точок центрів мас тетраедрів, на які розбито опуклий багатогранник.

Для розбиття опуклого багатогранника K на тетраедри використовується наступний алгоритм:

а) Проводимо триангуляцію граней опуклого багатогранника. Для цього, для j -ї грані (за умови, що j -а грань не є трикутником) знаходимо внутрішню точку як

$$t_j = \frac{\sum_{v_i \in V_j} v_i}{\text{card}(V_j)}$$

та з'єднуємо її з усіма вершинами з V_j .

б) Формуємо множину T усіх трикутників, що утворюють триангуляцію поверхні K .

в) Будуємо точку $t = \frac{\sum_{v_i \in V} v_i}{\text{card}(V)}$ (якщо V – тетраедр, то ця точка є

точкою його центру мас).

г) Проводимо тетраедризацію опуклого багатогранника, для цього з'єднуємо точку t з усіма вершинами трикутників з T .

Не втрачаючи загальності, вважаємо, що $(x_e, y_e, z_e) = (x_0 = 0, y_0 = 0, z_0)$ – центр мас Ω , значення z_0 визначається для кожного виду контейнера Ω .

Функції цілі. Розглядаються різні типи функцій цілі, які залежать від параметрів розміщення багатогранників і змінних метричних характеристик контейнера Ω , зокрема, об'єм, коефіцієнт гомотетії, метрична характеристика контейнера. Відзначимо, що будь-яка метрична характеристика контейнера Ω може бути змінною або

фіксованою.

Базова задача оптимальної упаковки багатогранників (задача ОРР) може бути сформульована таким чином:

Упакувати заданий набір багатогранників $Q_q(u_q)$ $q \in J_N$, всередині опуклого контейнера $\Omega(p)$, з урахуванням мінімально допустимих відстаней і обмежень балансу, так, щоб функція цілі досягала екстремуму (Рис. 2.5).

Залежно від виду функції цілі (об'єм, метрична характеристика контейнеру, коефіцієнт гомотетії), форми контейнеру (куля, кубоїд, прямий циліндр, еліпсоїд, опуклий багатогранник, довільна опукла область), заданих обмежень (обмеження на мінімально допустимі відстані, обмеження балансу) розглядаються різні варіанти (реалізації) базової задачі оптимальної упаковки багатогранників.

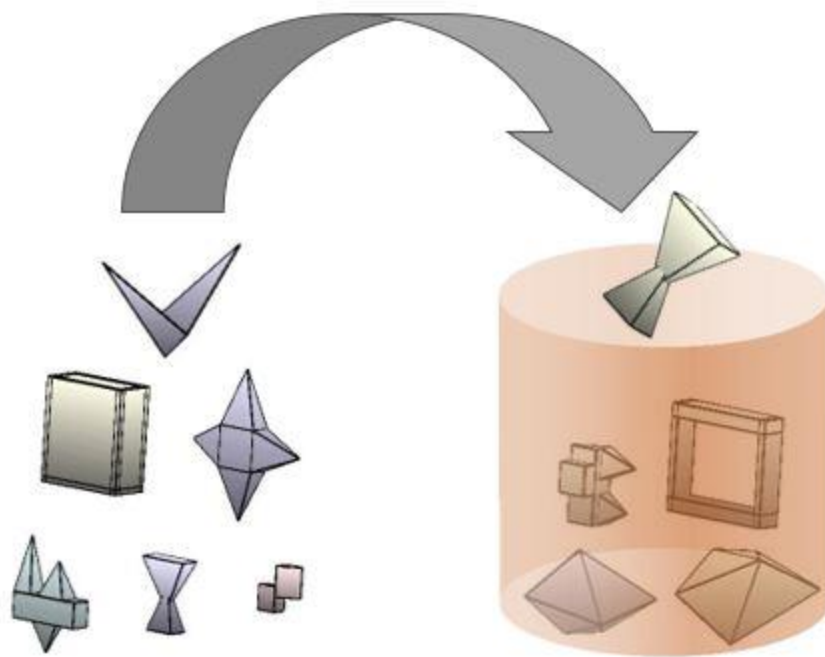


Рисунок 2.5 – Ілюстрація до постановки базової задачі упаковки довільних багатогранників в опуклий контейнер

Задача оптимального кластерингу багатогранників (задача ОРС) може бути сформульована, як окремий випадок задачі ОРР, таким чином:

Побудувати мінімальну охоплюючу оболонку, що має форму кубоїда, кулі або циліндра, для пари багатогранників $Q_1(u_1)$ і $Q_2(u_2)$, за умови $\text{int } Q_1(u_1) \cap \text{int } Q_2(u_2) = \emptyset$.

2.3 Висновки по розділу

У розділі 2 наведено визначення та основні властивості ρ -функції та квазі ρ -функції як засобів математичного моделювання відношень дотику, перетину, неперетину пари ρ -об'єктів. Визначено поняття нормалізованої та псевдонормалізованої ρ -функції, нормалізованої та псевдонормалізованої квазі ρ -функції як засобів моделювання обмежень на допустимі відстані між неорієнтованими довільними ρ -об'єктами. Визначено типи контейнерів, описано багатогранники як об'єкти розміщення, визначено обмеження розміщення та балансу, накладені на задачу. Сформульовано постановку базової задачі оптимальної упаковки довільних багатогранників та задачі оптимального кластерингу довільних багатогранників.

Наукові результати, наведені в розділі 2, опубліковані в таких роботах [33, 36, 42, 43].

Список джерел, які використано у даному розділі наведено у повному списку використаних джерел [137, 150, 152, 203-208].

3 МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ ОПТИМІЗАЦІЙНИХ ЗАДАЧ УПАКОВКИ БАГАТОГРАННИКІВ (ОРР)

В цьому розділі розглядаються засоби математичного моделювання обмежень розміщення (неперетину багатогранників, включення багатогранників у контейнер, обмежень на мінімально допустимі відстані визначені в розділі 2) в задачі ОРР. Для моделювання обмежень розміщення (неперетин об'єктів, включення об'єктів в опуклий контейнер, обмежень на мінімально допустимі відстані), будуються псевдонормалізовані ρ -функції і псевдонормалізовані квазі ρ -функції для неопуклих багатогранників, а також для неопуклих багатогранників і доповнень до контейнерів, що мають форму паралелепіпеда, кулі, прямого кругового циліндра, еліпсоїда, опуклого багатогранника і довільної опуклої області, обмеженої циліндричною, еліптичною, сферичною поверхнями або площинами. Будується математична модель базової задачі оптимальної упаковки довільних багатогранників в опуклому контейнері і наводяться її основні властивості. Будується математична модель задачі кластерингу неопуклих багатогранників. Наводиться приклад побудови математичної моделі для пари багатогранників.

Для опису обмежень неперетину і включення використовуються ρ -функції (див. розділ 2, визначення 2.1) і квазі ρ -функції (див. розділ 2, визначення 2.5), а для формалізації обмежень на мінімально допустимі відстані – псевдонормалізовані квазі ρ -функції (див. розділ 2, визначення 2.7) і псевдонормалізовані ρ -функції (див. розділ 2, визначення 2.3) [72, 209].

Нехай є два неопуклих багатогранники Q_1 і Q_2 з параметрами розміщення $u_1 = (v_1, \theta_1)$, $u_2 = (v_2, \theta_2)$, $v_1 = (x_1, y_1, z_1)$, $\theta_1 = (\theta_1^1, \theta_1^2, \theta_1^3)$,

$v_1 = (x_2, y_2, z_2)$, $\theta_2 = (\theta_2^1, \theta_2^2, \theta_2^3)$. Уявімо багатогранники $\mathbb{Q}_1(u_1)$ і $\mathbb{Q}_2(u_2)$ у вигляді об'єднання опуклих (Рис 3.1)

$$\mathbb{Q}_1(u_1) = \bigcup_{i=1}^{n_1} A_i(u_1) \text{ і } \mathbb{Q}_2(u_2) = \bigcup_{j=1}^{n_2} B_j(u_2), \quad (3.1)$$

де $A_i(u_1)$, $i \in \{1, 2, \dots, n_1\} = I_{n_1}$ і $B_j(u_2)$, $j \in \{1, 2, \dots, n_2\} = I_{n_2}$ – опуклі багатогранники. З кожним багатогранником $A_i(u_1) \subset \mathbb{Q}_1(u_1)$ асоціюється власна система координат, яка збігається з власною системою координат багатогранника $\mathbb{Q}_1(u_1)$.

Відзначимо, що кожен опуклий багатогранник $A_i(u_1) \subset \mathbb{Q}_1(u_1)$ задається своїми вершинами $p_s^{1i} = (p_{xs}^1, p_{ys}^1, p_{zs}^1)$ у власній системі координат багатогранника $\mathbb{Q}_1(u_1)$, де

$$\begin{aligned} p_{xs}^1 &= x_1 + (\sin \theta_1^1 \cos \theta_1^3 + \cos \theta_1^1 \cos \theta_1^2 \sin \theta_1^3) \cdot p_{xs}^0 + \\ &+ (-\sin \theta_1^1 \sin \theta_1^3 + \cos \theta_1^1 \cos \theta_1^2 \cos \theta_1^3) \cdot p_{ys}^0 + (-\cos \theta_1^1 \sin \theta_1^2) \cdot p_{zs}^0, \\ p_{ys}^1 &= y_1 + (\cos \theta_1^1 \cos \theta_1^3 - \sin \theta_1^1 \cos \theta_1^2 \sin \theta_1^3) \cdot p_{xs}^0 + \\ &+ (-\cos \theta_1^1 \sin \theta_1^3 - \sin \theta_1^1 \cos \theta_1^2 \cos \theta_1^3) \cdot p_{ys}^0 + (\sin \theta_1^1 \sin \theta_1^2) \cdot p_{zs}^0, \\ p_{zs}^1 &= z_1 + (\sin \theta_1^2 \sin \theta_1^3) \cdot p_{xs}^0 + (\sin \theta_1^2 \cos \theta_1^3) \cdot p_{ys}^0 + (\cos \theta_1^2) \cdot p_{zs}^0, \\ &s = 1, \dots, m_i^1, i \in I_{n_1}. \end{aligned}$$

Власно як і опуклий багатогранник $A_i(u_1) \subset \mathbb{Q}_1(u_1)$, кожен опуклий багатогранник $B_j(u_2) \subset \mathbb{Q}_2(u_2)$ задається своїми вершинами $p_s^{2j} = (p_{xs}^2, p_{ys}^2, p_{zs}^2)$ у власній системі координат багатогранника

$\mathbb{Q}_2(u_2)$, де

$$\begin{aligned}
 p_{xs}^2 &= x_2 + (\sin \theta_2^1 \cos \theta_2^3 + \cos \theta_2^1 \cos \theta_2^2 \sin \theta_2^3) \cdot p_{xs}^0 + \\
 &+ (-\sin \theta_2^1 \sin \theta_2^3 + \cos \theta_2^1 \cos \theta_2^2 \cos \theta_2^3) \cdot p_{ys}^0 + (-\cos \theta_2^1 \sin \theta_2^2) \cdot p_{zs}^0, \\
 p_{ys}^2 &= y_2 + (\cos \theta_2^1 \cos \theta_2^3 - \sin \theta_2^1 \cos \theta_2^2 \sin \theta_2^3) \cdot p_{xs}^0 + \\
 &+ (-\cos \theta_2^1 \sin \theta_2^3 - \sin \theta_2^1 \cos \theta_2^2 \cos \theta_2^3) \cdot p_{ys}^0 + (\sin \theta_2^1 \sin \theta_2^2) \cdot p_{zs}^0, \\
 p_{zs}^2 &= z_2 + (\sin \theta_2^2 \sin \theta_2^3) \cdot p_{xs}^0 + (\sin \theta_2^2 \cos \theta_2^3) \cdot p_{ys}^0 + (\cos \theta_2^2) \cdot p_{zs}^0, \\
 &s = 1, \dots, m_j^2, j \in I_{n_2}.
 \end{aligned}$$

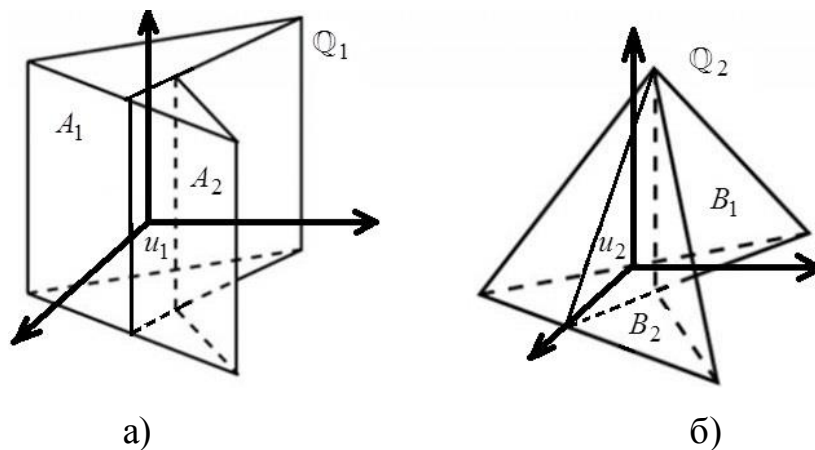


Рис. 3.1. Приклад декомпозиції неопуклих багатогранників $\mathbb{Q}_1(u_1)$ і

$\mathbb{Q}_2(u_2)$ на опуклі багатогранники: а) $\mathbb{Q}_1(u_1) = A_1(u_1) \cup A_2(u_1)$, б)

$$\mathbb{Q}_2(u_2) = B_1(u_2) \cup B_2(u_2)$$

3.1 Засоби математичного моделювання обмежень розміщення в задачі ОРР із застосуванням ϕ -функцій

3.1.1 ϕ -функції для моделювання обмежень включення.

Для моделювання обмеження включення вигляду

$$\mathbb{Q}_1(u_1) \subset \Omega(p) \Leftrightarrow \text{int } \mathbb{Q}_1(u_1) \cap \Omega^*(p) = \emptyset$$

використовується ϕ -функція $\Phi^{\mathbb{Q}_1\Omega^*}(u_1, p)$ для неопуклого багатогранника $\mathbb{Q}_1(u_1)$ і об'єкта $\Omega^*(p) = R^3 \setminus \text{int } \Omega(p)$, яка залежно від виду контейнера визначається таким чином:

$$\Phi^{\mathbb{Q}_1\Omega^*}(u_1, p) = \begin{cases} \Phi^{\mathbb{Q}_1\mathbf{S}^*}(u_1, p), & \Omega = \mathbf{S} \\ \Phi^{\mathbb{Q}_1\mathbf{C}^*}(u_1, p), & \Omega = \mathbf{C} \\ \Phi^{\mathbb{Q}_1\mathbf{B}^*}(u_1, p), & \Omega = \mathbf{B} \\ \Phi^{\mathbb{Q}_1\mathbf{P}^*}(u_1, p), & \Omega = \mathbf{P} \\ \Phi^{\mathbb{Q}_1\mathbf{E}^*}(u_1, p), & \Omega = \mathbf{E} \\ \Phi^{\mathbb{Q}_1\mathbf{\Lambda}^*}(u_1, p), & \Omega = \mathbf{\Lambda} \end{cases}$$

ϕ -функція для неопуклого багатогранника $\mathbb{Q}_1(u_1)$ і об'єкта $\Omega^*(p)$ має вигляд

$$\Phi^{\mathbb{Q}_1\Omega^*}(u_1, p) = \min\{\Phi^{A_1\Omega^*}(u_1, p), \dots, \Phi^{A_{n_1}\Omega^*}(u_1, p)\}, \quad (3.2)$$

де $\Phi^{A_i\Omega^*}(u_1, p)$ – ϕ -функція для опуклого багатогранника $A_i(u_1)$, що формує неопуклий багатогранник $\mathbb{Q}_1(u_1)$ в (3.1), і об'єкта $\Omega^*(p)$.

Зауважимо, що, як ϕ -функцію для неопуклого багатогранника $\mathbb{Q}_1(u_1)$ і об'єкта $\Omega^*(p)$ можна використовувати ϕ -функцію для опуклої оболонки неопуклого багатогранника $\mathbb{Q}_1(u_1)$ і об'єкта $\Omega^*(p)$.

Для побудови опуклої оболонки для скінченної множини точок існує велика кількість алгоритмів, наприклад алгоритм, описаний в [210].

Розглянемо вигляд ϕ -функцій $\Phi^{A\Omega^*}(u_1, p)$ для опуклого багатогранника і об'єкта $\Omega^*(p)$.

Перш за все, визначимо функцію

$$\begin{aligned}\varphi_A(u_1, r) &= \min\{\varphi_s(u_1, r), s = 1, \dots, m_A\}, \\ \varphi_s(u_1, r) &= r^2 - (p_{xs}^A)^2 - (p_{ys}^A)^2,\end{aligned}\tag{3.3}$$

та ϕ -функцію

$$\begin{aligned}\Phi^{AP}(u_1, \lambda) &= \min\{\phi_s(u_1, \lambda), s = 1, \dots, m_A\}, \\ \phi_s(u_1, \lambda) &= Ap_{xs}^A + Bp_{ys}^A + Cp_{zs}^A + \lambda D,\end{aligned}\tag{3.4}$$

для напівпростору $P = \{(x, y, z, \lambda) : Ax + By + Cz + \lambda D \geq 0\}$ та багатогранника $A(u_1)$, де $u_1 = (x_1, y_1, z_1, \theta_1^1, \theta_1^2, \theta_1^3)$.

ϕ -функція для $A(u_1)$ і об'єкта $S^*(r)$. ϕ -функція для $A(u_1)$ і об'єкта $S^*(r) = R^3 / \text{int } S(r)$ може бути визначена так:

$$\begin{aligned}\Phi^{AS^*}(u_1, r) &= \min\{\varphi_s(u_1, r), s = 1, \dots, m_A\}, \\ \varphi_s(u_1, r) &= r^2 - (p_{xs}^A)^2 - (p_{ys}^A)^2 - (p_{zs}^A)^2\end{aligned}\tag{3.5}$$

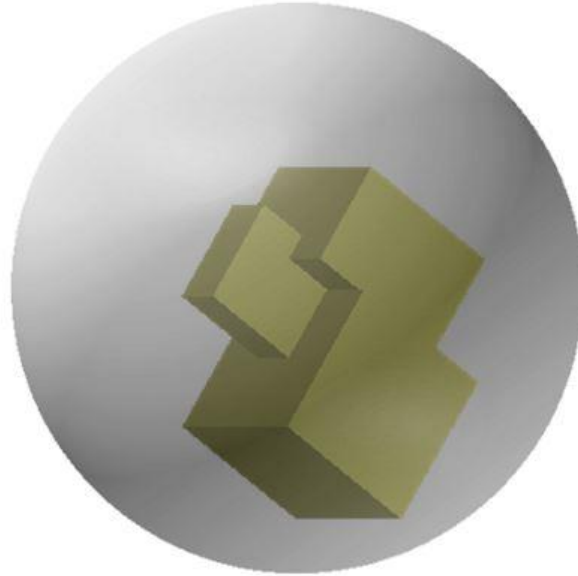


Рисунок 3.2 – Взаємне розміщення об'єктів $\mathbb{Q}_1(u_1)$ і $\mathbf{S}(r)$

На рис. 3.2 наведено взаємне розміщення об'єктів $\mathbb{Q}_1(u_1)$ і $\mathbf{S}(r)$, за якого $\mathbb{Q}_1(u_1) \subset \mathbf{S}(r) \Leftrightarrow \Phi^{\mathbb{Q}_1 \mathbf{S}^*}(u_1, r) > 0$.

Phi-функція для $A(u_1)$ і об'єкта $\mathbf{C}^(\lambda)$. Phi-функція для $A(u_1)$ і об'єкта $\mathbf{C}^*(\lambda) = R^3 / \text{int } \mathbf{C}(\lambda)$ може бути визначена так:*

$$\Phi^{A\mathbf{C}^*}(u_1, \lambda) = \min\{\Psi_t(u_1, \lambda), t = 1, 2, 3\}, \quad (3.6)$$

$$\Psi_1(u_1, \lambda) = \min\{\varphi_s(u_1, \lambda), s = 1, \dots, m_A\},$$

$$\varphi_s(u_1, \lambda) = (\lambda r)^2 - (p_{xs}^A)^2 - (p_{ys}^A)^2,$$

$$\Psi_2(u_1, \lambda) = \min\{\psi_s(u_1, \lambda), s = 1, \dots, m_A\},$$

$$\psi_s(u_1, \lambda) = p_{zs}^A + \lambda h,$$

$$\Psi_3(u_1, \lambda) = \min\{\omega_s(u_1, \lambda), s = 1, \dots, m_A\},$$

$$\omega_s(u_1, \lambda) = -p_{zs}^A + \lambda h,$$

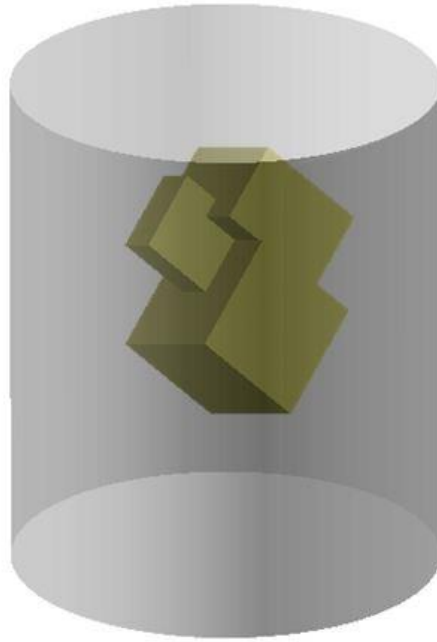


Рисунок 3.3 – Взаємне розміщення об'єктів $\mathbb{Q}_1(u_1)$ і $\mathbf{C}(\lambda)$

На рис. 3.3 наведено взаємне розміщення $\mathbb{Q}_1(u_1)$ і $\mathbf{C}(\lambda)$, за якого

$$\mathbb{Q}_1(u_1) \subset \mathbf{C}(\lambda) \Leftrightarrow \Phi^{\mathbb{Q}_1 \mathbf{C}^*}(u_1, \lambda) > 0.$$

Phi-функція для $A(u_1)$ і об'єкта $\mathbf{B}^(l, w, h)$. Phi-функція для $A(u_1)$ і об'єкта $\mathbf{B}^*(l, w, h) = R^3 / \text{int } \mathbf{B}(l, w, h)$ може бути визначена так:*

$$\Phi^{A\mathbf{B}^*}(u_1, l, w, h) = \min\left\{ \min_{1 \leq s \leq m_A} \Psi_{st}(u_1, l, w, h), t = 1, \dots, 6 \right\}, \quad (3.7)$$

$$\Psi_{s1}(u_1, l) = p_{xs}^A + l,$$

$$\Psi_{s2}(u_1, l) = -p_{xs}^A + l,$$

$$\Psi_{s3}(u_1, w) = p_{ys}^A + w,$$

$$\Psi_{s4}(u_1, w) = -p_{ys}^A + w,$$

$$\Psi_{s5}(u_1, h) = p_{zs}^A + h,$$

$$\Psi_{s6}(u_1, h) = -p_{zs}^A + h.$$

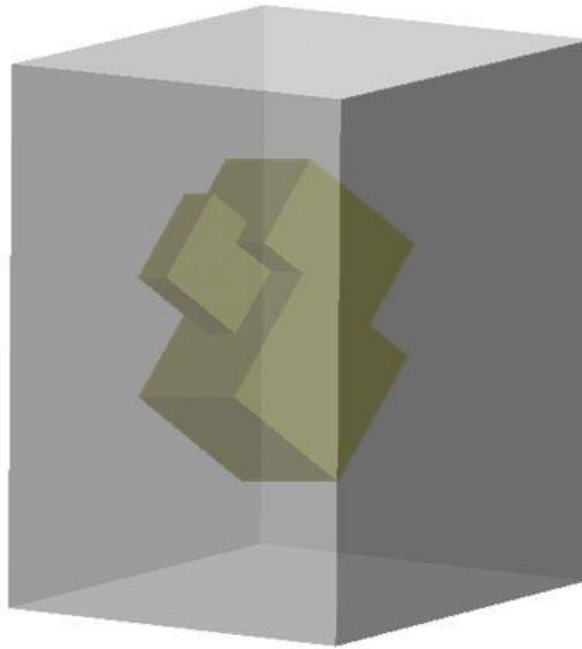


Рисунок 3.4 – Взаємне розміщення об'єктів $Q_1(u_1)$ і $\mathbf{B}(l, w, h)$

На рис. 3.4 наведено взаємне розміщення об'єктів $Q_1(u_1)$ і $\mathbf{B}(l, w, h)$, за якого $Q_1(u_1) \subset \mathbf{B}(l, w, h) \Leftrightarrow \Phi^{Q_1 \mathbf{B}^*}(u_1, l, w, h) > 0$.

Phi-функція для $A(u_1)$ і об'єкта $\mathbf{P}^(\lambda)$.* Phi-функція для $A(u_1)$ і об'єкта $\mathbf{P}^*(\lambda) = R^3 / \text{int } \mathbf{P}(\lambda)$ може бути визначена так:

$$\Phi^{A \mathbf{P}^*}(u_1, \lambda) = \min \left\{ \min_{1 \leq s \leq m_A} \Psi_{st}(u_1, \lambda), t = 1, \dots, m_{\mathbf{P}} \right\},$$

де $\Psi_{st}(u_1, \lambda) = A_t p_{xs}^A + B_t p_{ys}^A + C_t p_{zs}^A + \lambda D_t$, $\Psi_{st}(u_1, \lambda) = 0$ – рівняння площини, що відповідає t -ій грані багатогранного контейнеру $\mathbf{P}(\lambda)$.

На рис. 3.5 наведено взаємне розміщення об'єктів $Q_1(u_1)$ і $\mathbf{P}(\lambda)$, за якого $Q_1(u_1) \subset \mathbf{P}(\lambda) \Leftrightarrow \Phi^{Q_1 \mathbf{P}^*}(u_1, \lambda) > 0$.

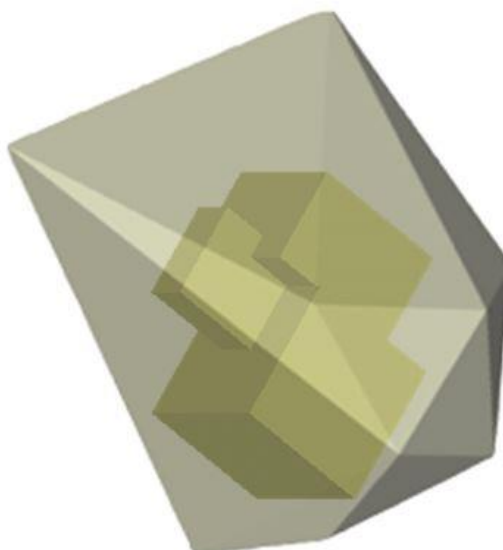


Рисунок 3.5 – Взаємне розміщення об'єктів $\mathbb{Q}_1(u_1)$ і $\mathbf{P}(\lambda)$

Phi-функція для $A(u_1)$ і об'єкта $\mathbf{E}^*(\lambda)$. *Phi*-функція для $A(u_1)$ і об'єкта $\mathbf{E}^*(\lambda) = R^3 / \text{int } \mathbf{E}(\lambda)$ може бути визначена так:

$$\Phi^{A\mathbf{E}^*}(u_1, \lambda) = \min\{\psi_s(u_1, \lambda), s = 1, \dots, m_A\}, \quad (3.8)$$

$$\psi_s(u_1, \lambda) = (\lambda)^2 - \frac{(p_{xs}^A)^2}{a^2} - \frac{(p_{ys}^A)^2}{b^2} - \frac{(p_{zs}^A)^2}{c^2}.$$

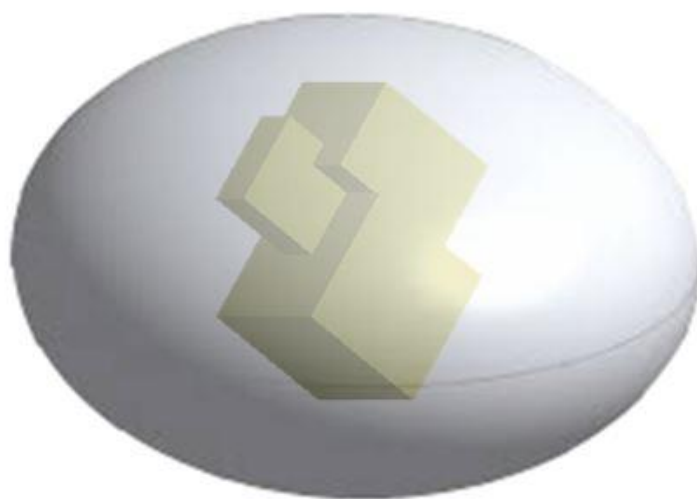


Рисунок 3.6 – Взаємне розміщення об'єктів $\mathbb{Q}_1(u_1)$ і $\mathbf{E}(\lambda)$

На рис. 3.6 наведено взаємне розміщення об'єктів $\mathbb{Q}_1(u_1)$ і $\mathbf{E}(\lambda)$, за якого $\mathbb{Q}_1(u_1) \subset \mathbf{E}(\lambda) \Leftrightarrow \Phi^{\mathbb{Q}_1 \mathbf{E}^*}(u_1, \lambda) > 0$.

Phi-функція для $A(u_1)$ і об'єкта $\Lambda^(\lambda)$.* Phi-функція для $A(u_1)$ і об'єкта $\Lambda^*(\lambda) = R^3 / \text{int } \Lambda(\lambda)$ може бути визначена так:

$$\Phi^{A\Lambda^*}(u_1, \lambda) = \min\{\Psi_t(u_1, \lambda), t = 1, 2, \dots, n_\Lambda\},$$

де $\Psi_t(u_1, \lambda)$ – функція вигляду (3.3), (3.4), (3.5), (3.8).

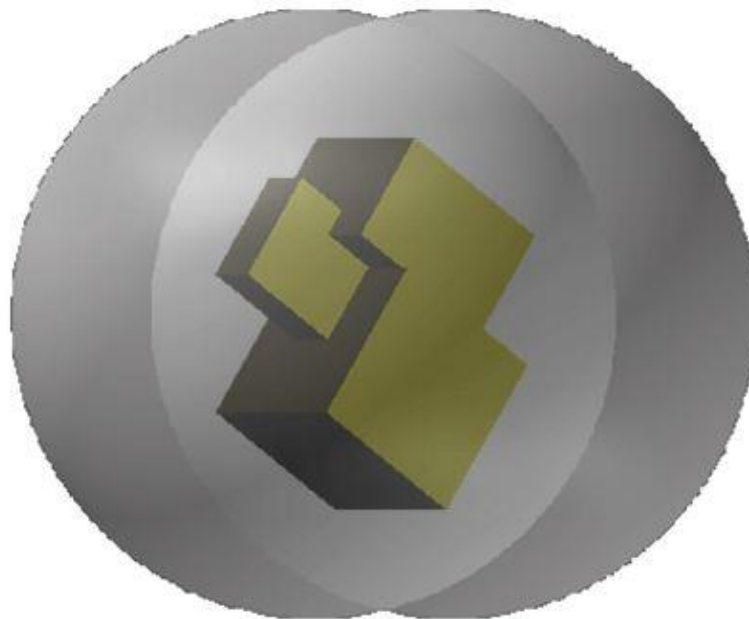


Рисунок 3.7 – Взаємне розміщення об'єктів $\mathbb{Q}_1(u_1)$ і $\Lambda(\lambda)$

На рис. 3.7 наведено взаємне розміщення об'єктів $\mathbb{Q}_1(u_1)$ і $\Lambda(\lambda) = \mathbf{S}_1(\lambda) \cap \mathbf{S}_2(\lambda)$, за якого $\mathbb{Q}_1(u_1) \subset \Lambda(\lambda) \Leftrightarrow \Phi^{\mathbb{Q}_1 \Lambda^*}(u_1, \lambda) > 0$.

3.1.2 Псевдонормалізовані та нормалізовані ϕ -функції для моделювання обмежень включення з урахуванням допустимих відстаней.

Нехай задано мінімально допустима відстань ρ між неопуклим багатогранником $\mathbb{Q}_1(u_1)$ і об'єктом $\Omega^*(p)$. Зазначемо, що $dist(\mathbb{Q}_1(u_1), \Omega^*(p)) \geq \rho \Leftrightarrow dist(A_j(u_1), \Omega^*(p)) \geq \rho, j=1, \dots, n_1$.

Для опису обмеження $dist(\mathbb{Q}_1(u_1), \Omega^*(p)) \geq \rho$ будемо використовувати псевдонормалізовану ϕ -функцію вигляду

$$\hat{\Phi}^{\mathbb{Q}_1\Omega^*}(u_1, p) = \min\{\hat{\Phi}^{A_1\Omega^*}(u_1, p), \dots, \hat{\Phi}^{A_{n_1}\Omega^*}(u_1, p)\}, \quad (3.9)$$

або нормалізовану ϕ -функцію вигляду

$$\tilde{\Phi}^{\mathbb{Q}_1\Omega^*}(u_1, p) = \min\{\tilde{\Phi}^{A_1\Omega^*}(u_1, p), \dots, \tilde{\Phi}^{A_{n_1}\Omega^*}(u_1, p)\}.$$

$$\hat{\Phi}^{A_j\Omega^*}(u_1, p) = \tilde{\Phi}^{A_j\Omega^*}(u_1, p) - \rho,$$

У свою чергу, для опису обмеження $dist(A_j(u_1), \Omega^*(p)) \geq \rho$ використовується псевдонормалізована або нормалізована ϕ -функція для об'єктів $A_j(u_1)$ і $\Omega^*(p)$.

Якщо контейнер має вигляд кулі $\mathbf{S}(r)$, тоді:

– нормалізована ϕ -функція для опуклого багатогранника $A(u_1)$ і об'єкта $\mathbf{S}^*(r)$ виглядає так:

$$\tilde{\Phi}^{AS^*}(u_1, r) = \min\{\tilde{\Phi}_s(u_1, r), s = 1, \dots, m_A\}, \quad (3.10)$$

$$\tilde{\Phi}_s(u_1, r) = r - \sqrt{(p_{xs}^A)^2 - (p_{ys}^A)^2 - (p_{zs}^A)^2}.$$

– псевдонормалізована *phi*-функція для опуклого багатогранника $A(u_1)$ і об'єкта $S^*(r)$ виглядає так:

$$\hat{\Phi}^{AS^*}(u_1, r) = \min\{\hat{\Phi}_s(u_1, r), s = 1, \dots, m_A\}, \quad (3.11)$$

$$\hat{\Phi}_s(u_1, r) = (r - \rho)^2 - (p_{xs}^A)^2 - (p_{ys}^A)^2 - (p_{zs}^A)^2.$$

Якщо контейнер має вигляд циліндра $C(\lambda)$, тоді:

– нормалізована *phi*-функція для опуклого багатогранника $A(u_1)$ і об'єкта $C^*(\lambda)$ виглядає так:

$$\tilde{\Phi}^{AC^*}(u_1, \lambda) = \min\{\tilde{\Psi}_t(u_1, \lambda), t = 1, 2, 3\}, \quad (3.12)$$

$$\tilde{\Psi}_1(u_1, \lambda) = \min\{\tilde{\Phi}_s(u_1, \lambda), s = 1, \dots, m_A\},$$

$$\tilde{\Phi}_s(u_1, \lambda) = \lambda r - \sqrt{(p_{xs}^A)^2 + (p_{ys}^A)^2},$$

$$\tilde{\Psi}_2(u_1, \lambda) = \min\{\psi_s(u_1, \lambda), s = 1, \dots, m_A\},$$

$$\psi_s(u_1, \lambda) = p_{zs}^A + \lambda h,$$

$$\tilde{\Psi}_3(u_1, \lambda) = \min\{\omega_s(u_1, \lambda), s = 1, \dots, m_A\},$$

$$\omega_s(u_1, \lambda) = -p_{zs}^A + \lambda h.$$

– псевдонормалізована *phi*-функція для опуклого багатогранника $A(u_1)$ і об'єкта $C^*(\lambda)$ виглядає так:

$$\widehat{\Phi}^{AC^*}(u_1, \lambda) = \min\{\Psi_t(u_1, \lambda), t = 1, 2, 3\}, \quad (3.13)$$

$$\Psi_1(u_1, \lambda) = \min\{\widehat{\phi}_s(u_1, \lambda), s = 1, \dots, m_A\},$$

$$\widehat{\phi}_s(u_1, \lambda) = (\lambda(r - \rho))^2 - (p_{xs}^A)^2 - (p_{ys}^A)^2,$$

$$\Psi_2(u_1, \lambda) = \min\{\widehat{\psi}_s(u_1, \lambda), s = 1, \dots, m_A\},$$

$$\widehat{\psi}_s(u_1, \lambda) = p_{zs}^A + \lambda(h - \rho),$$

$$\Psi_3(u_1, \lambda) = \min\{\widehat{\omega}_s(u_1, \lambda), s = 1, \dots, m_A\},$$

$$\widehat{\omega}_s(u_1, \lambda) = -p_{zs}^A + \lambda(h - \rho),$$

Якщо контейнер має вигляд паралелепіпеда $\mathbf{B}(l, w, h)$, тоді:

– нормалізована *phi*-функція $\tilde{\Phi}^{AB^*}(u_1, l, w, h)$ для опуклого багатогранника $A(u_1)$ і об'єкта $\mathbf{B}^*(l, w, h)$ визначається формулою (3.7).

– псевдонормалізована *phi*-функція $\tilde{\Phi}^{AB^*}(u_1, l, w, h)$ для опуклого багатогранника $A(u_1)$ і об'єкта $\mathbf{B}^*(l, w, h)$ виглядає так:

$$\widehat{\Phi}^{AB^*}(u_1, l, w, h) = \min\left\{\min_{1 \leq s \leq m_A} \Psi_{st}(u_1, l, w, h), t = 1, \dots, 6\right\}, \quad (3.14)$$

$$\Psi_{s1}(u_1, l) = p_{xs}^A + l - \rho,$$

$$\Psi_{s2}(u_1, l) = -p_{xs}^A + l - \rho,$$

$$\Psi_{s3}(u_1, w) = p_{ys}^A + w - \rho,$$

$$\Psi_{s4}(u_1, w) = -p_{ys}^A + w - \rho,$$

$$\Psi_{s5}(u_1, h) = p_{zs}^A + h - \rho,$$

$$\Psi_{s6}(u_1, h) = -p_{zs}^A + h - \rho.$$

Якщо контейнер має вигляд опуклого багатогранника $\mathbf{P}(\lambda)$, тоді:

– нормалізована *phi*-функція для опуклого багатогранника $A(u_1)$ і об'єкта $\mathbf{P}^*(\lambda)$ виглядає так:

$$\tilde{\Phi}^{A\mathbf{P}^*}(u_1, \lambda) = \min\left\{ \min_{1 \leq s \leq m_A} \tilde{\Psi}_{st}(u_1, \lambda), t = 1, \dots, m_{\mathbf{P}} \right\}, \quad (3, 15)$$

де $\tilde{\Psi}_{st}(u_1, \lambda) = \tilde{A}_t p_{xs}^A + \tilde{B}_t p_{ys}^A + \tilde{C}_t p_{zs}^A + \lambda \tilde{D}_t$,

$$\tilde{A}_t = \frac{A_t}{\sqrt{A_t^2 + B_t^2 + C_t^2}}, \quad \tilde{B}_t = \frac{B_t}{\sqrt{A_t^2 + B_t^2 + C_t^2}}, \quad \tilde{C}_t = \frac{C_t}{\sqrt{A_t^2 + B_t^2 + C_t^2}},$$

$\tilde{D}_t = \frac{D_t}{\sqrt{A_t^2 + B_t^2 + C_t^2}}$, $\tilde{\Psi}_{st}(u_1, \lambda) = 0$ – нормальне рівняння площини, що

відповідає t -ій грані багатогранного контейнеру $\mathbf{P}(\lambda)$.

– псевдонормалізована *phi*-функція для опуклого багатогранника $A(u_1)$ і об'єкта $\mathbf{P}^*(\lambda)$ виглядає так:

$$\hat{\Phi}^{A\mathbf{P}^*}(u_1, \lambda) = \min\left\{ \min_{1 \leq s \leq m_A} \Psi_{st}(u_1, \lambda), t = 1, \dots, m_{\mathbf{P}} \right\}, \quad (3.16)$$

де $\Psi_{st}(u_1, \lambda) = A_t p_{xs}^A + B_t p_{ys}^A + C_t p_{zs}^A + \lambda(D_t - \rho)$,

$\Psi_{st}(u_1, \lambda) = 0$ – рівняння площини, що відповідає t -ій грані багатогранного контейнеру $\mathbf{P}(\lambda)$.

Якщо контейнер має вигляд еліпсоїда $\mathbf{E}(\lambda)$, тоді використовується

ρ -функція для опуклого багатогранника $A(u_1)$ і об'єкта $E_\rho^*(\lambda)$ яка має вигляд

$$\Phi^{AE_\rho^*}(u_1, \lambda) = \min\{\varphi_s(u_1, \lambda), s = 1, \dots, m_A\}, \quad (3.17)$$

$$\varphi_s(u_1, \lambda) = \lambda^2 - \frac{(p_{xs}^A)^2}{(a-\rho)^2} - \frac{(p_{ys}^A)^2}{(b-\rho)^2} - \frac{(p_{zs}^A)^2}{(c-\rho)^2}.$$

де $E_\rho^*(x, y, z, \lambda) = \{(x, y, z, \lambda) \mid \lambda^2 - \frac{x^2}{(a-\rho)^2} - \frac{y^2}{(b-\rho)^2} - \frac{z^2}{(c-\rho)^2}\},$

$a-\rho > 0, b-\rho > 0, c-\rho > 0.$

3.2 Засоби математичного моделювання обмежень розміщення в задачі ОРР із застосуванням квазі ρ -функцій

3.2.1 Квазі ρ -функції для моделювання обмежень неперетину.

Нехай $Q_q(u_q)$ і $Q_g(u_g)$ – неопуклі багатогранники.

Твердження 3.1. Функція вигляду

$$\Phi'_{qg}(u_q, u_g, u'_{qg}) = \min\{\Phi'_{ij}(u_q, u_g, u'_{ij}), i = 1, \dots, n_q, j = 1, \dots, n_g\} \quad (3.18)$$

є квазі ρ -функцією для неопуклих багатогранників $Q_q(u_q)$ і $Q_g(u_g)$,

де $u'_{qg} = (u'_{ij}, i = 1, \dots, n_q, j = 1, \dots, n_g), u'_{ij} = (\theta_{ij}^1, \theta_{ij}^2, \xi_{ij}), \theta_{ij}^1, \theta_{ij}^2, \xi_{ij}$ –

параметри напівпростору $P_{ij} = \{(x, y, z) : \psi_{ij}(x, y, z) =$

$$= \alpha_{ij} \cdot x + \beta_{ij} \cdot y + \gamma_{ij} \cdot z + \xi_{ij} \geq 0\}, \quad \alpha = \sin \theta_{ij}^2, \quad \beta = -\sin \theta_{ij}^1 \cdot \cos \theta_{ij}^2, \\ \gamma = \cos \theta_{ij}^1 \cdot \cos \theta_{ij}^2.$$

Доведення. Покажемо, що $\max_{u'_{qg}} \Phi'_{qg}(u_q, u_g, u'_{qg})$ є phi-функцією

для неопуклих багатогранників $\mathbb{Q}_q(u_q)$ і $\mathbb{Q}_g(u_g)$, тобто

$$1) \max_{u'_{qg}} \Phi'_{qg}(u_q, u_g, u'_{qg}) < 0, \text{ якщо } \text{int } \mathbb{Q}_q(u_q) \cap \text{int } \mathbb{Q}_g(u_g) \neq \emptyset,$$

$$2) \max_{u'_{qg}} \Phi'_{qg}(u_q, u_g, u'_{qg}) = 0, \text{ якщо } \text{int } \mathbb{Q}_q(u_q) \cap \text{int } \mathbb{Q}_g(u_g) = \emptyset \text{ і}$$

$$\text{fr} \mathbb{Q}_q(u_q) \cap \text{fr} \mathbb{Q}_g(u_g) \neq \emptyset,$$

$$3) \max_{u'_{qg}} \Phi'_{qg}(u_q, u_g, u'_{qg}) > 0, \text{ якщо } \mathbb{Q}_q(u_q) \cap \mathbb{Q}_g(u_g) = \emptyset.$$

Оскільки в кожній з функцій вигляду $\Phi'_{ij}(u_q, u_g, u'_{ij})$, додаткові змінні u'_{ij} є незалежними для кожної пари $(i, j), i = 1, \dots, n_q, j = 1, \dots, n_g$, то, отже,

$$\begin{aligned} & \max_{u'_{qg}} \{ \min \{ \Phi'_{ij}(u_q, u_g, u'_{ij}), i = 1, \dots, n_q, j = 1, \dots, n_g \} \} = \\ & = \min \{ \max \{ \Phi'_{ij}(u_i, u_j, u'_{ij}), i = 1, \dots, n_q, j = 1, \dots, n_g \} \} = \\ & = \min \{ \Phi_{ij}(u_i, u_j), i = 1, \dots, n_q, j = 1, \dots, n_g \}, \end{aligned}$$

де $\min \{ \Phi_{ij}(u_i, u_j), i = 1, \dots, n_q, j = 1, \dots, n_g \}$ – phi-функція для неопуклих багатогранників $\mathbb{Q}_q(u_q)$ і $\mathbb{Q}_g(u_g)$.

Звідси випливає, що

$$\max_{u'_{qg}} \Phi'_{qg}(u_q, u_g, u'_{qg}) =$$

$$= \min \{ \Phi_{ij}(u_i, u_j), i = 1, \dots, n_q, j = 1, \dots, n_g \} = \Phi_{qg}(u_q, u_g).$$

Таким чином, функція $\Phi_{qg}(u_q, u_g)$ є ϕ -функцією для неопуклих багатогранників $\mathbb{Q}_q(u_q)$ і $\mathbb{Q}_g(u_g)$, а отже, виконуються властивості 1–3. ►

Використовуючи властивості квазі ϕ -функцій [209], умову неперетину багатогранників $\mathbb{Q}_q(u_q)$ і $\mathbb{Q}_g(u_g)$ можна описати у вигляді нерівності $\Phi'_{qg}(u_q, u_g, u'_{qg}) \geq 0 \Rightarrow \text{int } \mathbb{Q}_q(u_q) \cap \text{int } \mathbb{Q}_g(u_g) = \emptyset$, де $\Phi'_{qg}(u_q, u_g, u'_{qg})$ визначається формулою (3.18).

3.2.2 Псевдонормалізовані та нормалізовані квазі ϕ -функції для моделювання обмежень на допустимі відстані.

У разі якщо між багатогранниками $\mathbb{Q}_q(u_q)$ і $\mathbb{Q}_g(u_g)$ задана мінімально допустима відстань $\rho_{qg} > 0$, то обмеження $\text{dist}(\mathbb{Q}_q(u_q), \mathbb{Q}_g(u_g)) \geq \rho_{qg}$, можна описати за допомогою псевдонормалізованої квазі ϕ -функції.

Нехай $\widehat{\Phi}'_{ij}(u_q, u_g, u'_{ij})$ – псевдонормалізована квазі ϕ -функція для опуклих багатогранників $K_i(u_q) \subset \mathbb{Q}_q(u_q)$ і $K_j(u_g) \subset \mathbb{Q}_g(u_g)$ вигляду (Рис. 3.8):

$$\widehat{\Phi}'_{ij}(u_q, u_g, u'_{ij}) = \min \{ \Phi^{K_i P_{ij}}(u_q, u'_{ij}), \Phi^{K_j P_{ij}^*}(u_g, u'_{ij}) \} - 0.5\rho_{qg}$$

де

$$\Phi^{K_i P_{ij}}(u_q, u'_{ij}) = \min_{1 \leq s \leq m_i} \psi_{ij}(p_s^i) \text{ і } \Phi^{K_j P_{ij}^*}(u_q, u'_{ij}) = \min_{1 \leq s \leq m_j} (-\psi_{ij}(p_s^j)) \quad [74],$$

$$P_{ij}^* = R^3 \setminus \text{int } P_{ij}.$$

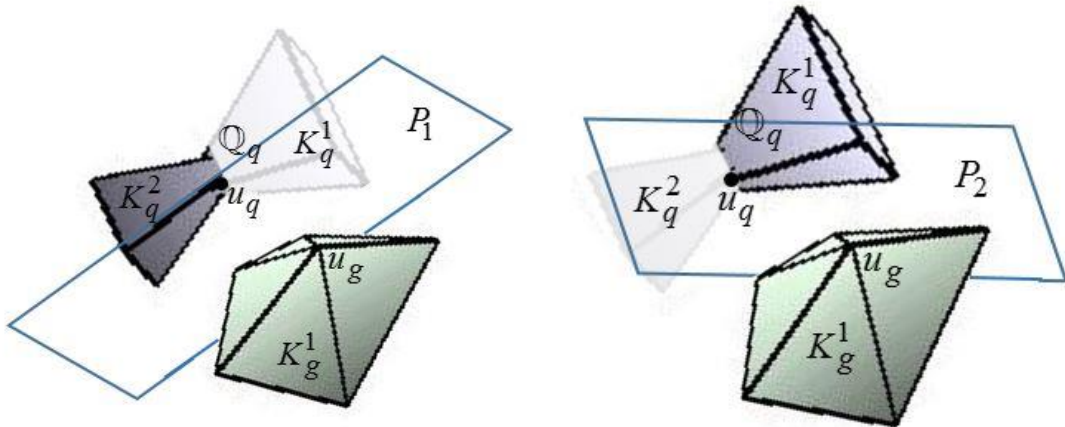


Рисунок 3.8. – Багатогранники $K_i^q, i=1, 2, K_1^g$ і площини P_1 і P_2 , що їх розділяють

Таким чином $\Phi'_{ij}(u_q, u_g, u'_{ij}) \geq 0 \Rightarrow \text{dist}(K_i(u_q), K_j(u_g)) \geq \rho_{qg}$.

Твердження 3.2. Функція

$$\widehat{\Phi}'_{qg}(u_q, u_g, u'_{qg}) = \min \{ \widehat{\Phi}'_{ij}(u_q, u_g, u'_{ij}), i=1, \dots, n_q, j=1, \dots, n_g \} \quad (3.19)$$

є псевдонормалізованою квазі рhі-функцією для неопуклих багатогранників $\mathbb{Q}_q(u_q)$ і $\mathbb{Q}_g(u_g)$.

Доведення. Покажемо, що $\max_{u'_{qg}} \widehat{\Phi}'_{qg}(u_q, u_g, u'_{qg})$ є

псевдонормалізованою рhі-функцією для неопуклих багатогранників $\mathbb{Q}_q(u_q)$ і $\mathbb{Q}_g(u_g)$ [72], тобто

- 1) $\max_{u'_{qg}} \widehat{\Phi}'_{qg}(u_q, u_g, u'_{qg}) > 0$, якщо $dist(\mathbb{Q}_q(u_q), \mathbb{Q}_g(u_g)) > \rho_{qg}$,
- 2) $\max_{u'_{qg}} \widehat{\Phi}'_{qg}(u_q, u_g, u'_{qg}) = 0$, якщо $dist(\mathbb{Q}_q(u_q), \mathbb{Q}_g(u_g)) = \rho_{qg}$,
- 3) $\max_{u'_{qg}} \widehat{\Phi}'_{qg}(u_q, u_g, u'_{qg}) < 0$, якщо

$$dist(\mathbb{Q}_q(u_q), \mathbb{Q}_g(u_g)) < \rho_{qg}.$$

Оскільки в кожній з функцій вигляду $\widehat{\Phi}'_{ij}(u_q, u_g, u'_{ij})$, додаткові змінні u'_{ij} є незалежними для кожної пари (i, j) , $i = 1, \dots, n_q$, $j = 1, \dots, n_g$, то, отже,

$$\begin{aligned} & \max_{u'_{qg}} \{ \min \{ \widehat{\Phi}'_{ij}(u_q, u_g, u'_{ij}), i = 1, \dots, n_q, j = 1, \dots, n_g \} \} = \\ & = \min_{u'_{ij}} \{ \max \{ \widehat{\Phi}'_{ij}(u_q, u_g, u'_{ij}), i = 1, \dots, n_q, j = 1, \dots, n_g \} \} = \\ & = \min \{ \widehat{\Phi}_{ij}(u_q, u_g), i = 1, \dots, n_q, j = 1, \dots, n_g \}, \end{aligned}$$

де $\min \{ \widehat{\Phi}_{ij}(u_q, u_g), i = 1, \dots, n_q, j = 1, \dots, n_g \}$ – псевдонормалізована рhi-функція для неопуклих багатогранників $\mathbb{Q}_q(u_q)$ і $\mathbb{Q}_g(u_g)$. Звідси випливає, що

$$\begin{aligned} & \max_{u'_{qg}} \widehat{\Phi}'_{qg}(u_q, u_g, u'_{qg}) = \\ & = \min \{ \widehat{\Phi}_{ij}(u_q, u_g), i = 1, \dots, n_q, j = 1, \dots, n_g \} = \\ & = \widehat{\Phi}_{qg}(u_q, u_g). \end{aligned}$$

Таким чином, функція $\widehat{\Phi}_{qg}(u_q, u_g)$ є псевдонормалізованою рhi-функцією для неопуклих багатогранників $\mathbb{Q}_q(u_q)$ і $\mathbb{Q}_g(u_g)$, а отже, виконуються властивості 1–3. ►

Використовуючи властивості псевдонормалізованих квазі рhi-функцій [209], обмеження на мінімально допустиму відстань між неопуклими багатогранниками $\mathbb{Q}_q(u_q)$ і $\mathbb{Q}_g(u_g)$ можна описати у вигляді нерівності $\widehat{\Phi}'_{qg}(u_q, u_g, u'_{qg}) \geq 0 \Rightarrow \text{dist}(\mathbb{Q}_q(u_q), \mathbb{Q}_g(u_g)) \geq \rho_{qg}$, де $\widehat{\Phi}'_{qg}(u_q, u_g, u'_{qg})$ визначається формулою (3.19).

3.3 Математична модель базової задачі оптимальної упаковки багатогранників та її властивості

Вектор $u \in R^\sigma$ всіх змінних може бути описаний таким чином: $u = (\zeta, \tau) \in R^\sigma$, де $\zeta = (p, u_1, u_2, \dots, u_N)$, p – вектор змінних метричних характеристик заданого контейнера Ω і $u_{a_i} = (v_{a_i}, \theta_{a_i}) = (x_{a_i}, y_{a_i}, z_{a_i}, \theta_{a_i}^1, \theta_{a_i}^2, \theta_{a_i}^3)$ – вектор змінних параметрів розміщення $K_i(u_{a_i}) \subset \mathbb{Q}_q(u_q)$, $i \in I_n$, індекс $a_i \in \{1, 2, \dots, N\}$ – компонента "склеювального" вектора $\mathbf{a} = (a_1, \dots, a_n)$, $a_i \in J_N$, де $a_i = q$, якщо K_i бере участь у формуванні неопуклого багатогранника, $\mathbb{Q}_q(u_q)$, $q \in J_N$.

Тут $\tau = (u'^1, \dots, u'^m)$ позначає вектор всіх додаткових змінних, де $u'^s = (\theta^{1s}, \theta^{2s}, \xi^s)$ – вектор додаткових змінних для s -ї пари опуклих багатогранників, визначених у (3.1) $s = 1, \dots, m$, $m = \text{card}(\Xi)$,

$$\Xi = \{(i, j), a_i \neq a_j, i < j = 1, \dots, n\}. \quad (3.20)$$

Число змінних задачі виводиться як $\sigma = |p| + 6N + 3m$.

Таким чином, математична модель задачі упаковки багатогранників може бути сформульована як

$$\min_{u \in W \subset R^\sigma} F(u), \quad (3.21)$$

$$W = \{u \in R^\sigma : \hat{\Phi}'_{ij}(u_{a_i}, u_{a_j}, u'_{a_i a_j}) \geq 0, (i, j) \in \Xi, , \\ \hat{\Phi}_i(u_{a_i}, p) \geq 0, i \in I_n, \mu(p, u) \geq 0, \delta(u) \geq 0\} \quad (3.22)$$

де $F(u) = p$, $\hat{\Phi}'_{ij}(u_{a_i}, u_{a_j}, u'_{a_i a_j})$ – псевдонормалізована квазі ρ -функція, визначена в (3.19);

$$a_i, a_j \in J_N, \text{ де } (i, j) \in \Xi, u'_{a_i a_j} = u'^s, s = 1, \dots, m;$$

Ξ наведено в (3.20), для пари багатогранників $K_i(u_{a_i}) \subset \mathbb{Q}_q(u_q)$ і $K_j(u_{a_j}) \subset \mathbb{Q}_g(u_g)$, беручи до уваги мінімально допустимі відстані ρ_{qg} ;

$\hat{\Phi}_i(u_{a_i}, p)$ – псевдонормалізована ρ -функція для багатогранника $K_i(u_{a_i}) \subset \mathbb{Q}_q(u_q)$ і об'єкта $\Omega^*(p)$, враховуючи мінімально допустимі відстані ρ_q ;

$$\mu(p, u) \geq 0 \text{ – система обмежень балансу визначена в (2.10);}$$

$\delta(u) \geq 0$ – система додаткових обмежень, що можуть бути накладені на метричні характеристики контейнера та /або на параметри розміщення багатогранників.

Якщо ρ_{qg} і ρ_q не задані, тоді замінюємо псевдонормалізовані квазі ϕ -функції $\widehat{\Phi}'_{ij}(u_{a_i}, u_{a_j}, u'_{a_i a_j})$ на квазі ϕ -функції $\Phi'_{ij}(u_{a_i}, u_{a_j}, u'_{a_i a_j})$, визначені в (3.18), для забезпечення дотримання обмежень неперетину, і псевдонормалізовані ϕ -функції $\widehat{\Phi}_i(u_{a_i}, p)$ на ϕ -функції $\Phi_i(u_{a_i}, p)$ для забезпечення дотримання обмежень включення.

Слід зазначити: для того щоб уникнути надмірності нерівностей в обмеженнях включення, можна використовувати псевдонормалізовану ϕ -функцію $\widehat{\Phi}_q^h(u_q, p)$ для опуклої оболонки кожного неопуклого багатогранника $Q_q(u_q)$, $q=1, \dots, N$, замість відповідних псевдонормалізованих ϕ -функцій $\widehat{\Phi}_i(u_{a_i}, p)$, $i=1, \dots, n$, для опуклих багатогранників K_i , $i=1, \dots, n$.

Математична модель задачі оптимального кластерингу (ОРС) багатогранників $Q_1(u_1)$ і $Q_2(u_2)$, що не перетинаються, яка полягає в пошуку мінімально охоплюючої оболонки може бути подана у вигляді наступної задачі нелінійного програмування:

$$\min_{u \in W \subset R^\sigma} F(u), \quad (3.23)$$

$$W = \{u \in R^\sigma : \Phi^{\Omega^* Q_1}(u_1, p) \geq 0, \Phi^{\Omega^* Q_2}(u_2, p) \geq 0, \widehat{\Phi}'^{Q_1 Q_2}(u_1, u_2) \geq 0\}, \quad (3.24)$$

де $F(u)$ – функція цілі (об'єм, метрична характеристика або коефіцієнт гомотетії контейнера);

W – область допустимих розв'язків;

$u = (p, u_1, u_2) \in R^\sigma$ – вектор змінних;

R^σ – евклідів простір розмірності σ ($\sigma=15$, якщо контейнер має форму кубоїда, $\sigma=13$, якщо контейнер має форму кулі, еліпсоїда, циліндра, опуклого багатогранника або довільної опуклої області);

p – вектор змінних метричних характеристик контейнера Ω , наприклад $p = (l, w, h)$, якщо контейнер має форму кубоїда, $p = r$, якщо контейнер має форму кулі, $p = \lambda$, якщо контейнер має форму еліпсоїда, циліндра, опуклого багатогранника, довільної опуклої області;

$u_1 = (x_{Q_1}, y_{Q_1}, z_{Q_1}, \theta_{Q_1}^1, \theta_{Q_1}^2, \theta_{Q_1}^3)$ – вектор змінних параметрів розміщення неопуклого багатогранника $Q_1(u_1)$;

$u_2 = (x_{Q_2}, y_{Q_2}, z_{Q_2}, \theta_{Q_2}^1, \theta_{Q_2}^2, \theta_{Q_2}^3)$ – вектор змінних параметрів розміщення неопуклого багатогранника $Q_2(u_2)$;

$\widehat{\Phi}'^{Q_1 Q_2}(u_1, u_2, p)$ – псевдонормалізована квазі ϕ -функція вигляду (3.19) для неопуклих багатогранників $Q_1(u_1)$ і $Q_2(u_2)$, що враховує задану мінімально допустиму відстані ρ між багатогранниками;

$\Phi^{Q_1 \Omega^*}(u_1, p)$ – ϕ -функція вигляду (3.2) для багатогранника $Q_1(u_1)$ і $\Omega^*(p)$;

$\Phi^{Q_2 \Omega^*}(u_2, p)$ – ϕ -функція вигляду (3.2) для багатогранника $Q_2(u_2)$ і $\Omega^*(p)$.

Слід зазначити, якщо мінімально допустимі відстані не задані, тоді замість псевдонормалізованих ϕ -функцій і квазі ϕ -функцій використовуються ϕ -функції і квазі ϕ -функції.

Приклад побудови області допустимих розв'язків для задачі оптимального кластерингу наведено в додатку Г.

Розглянемо основні особливості математичної моделі (3.21) – (3.22). Математична модель (3.21) – (3.22) є неопуклою неперервною задачею нелінійного програмування, що включає всі глобально оптимальні розв'язки задачі оптимальної упаковки довільних багатогранників. Область допустимих розв'язків (3.22) описується системою нерівностей з гладкими функціями. Розмірність задачі і вигляд функцій $\Phi_q, \widehat{\Phi}_q, \Phi'_{qg}, \widehat{\Phi}'_{qg}$, що описують область допустимих розв'язків (3.22), залежить від: форми контейнера (паралелепіпед, куля, циліндр, опуклий багатогранник, еліпсоїд, довільна опукла область), наявності обмежень на мінімально допустимі відстані та обмеження балансу.

Існує можливість, хоча б теоретично, використання глобального NLP-солвера для розв'язання задачі (3.21)– (3.22).

Приклад 3.1. Як приклад розглянемо математичну модель задачі упаковки двох багатогранників: $\mathbb{Q}_1(u_1) = K_1(u_1)$ і $\mathbb{Q}_2(u_2) = K_2(u_2) \cup K_3(u_2)$ (рис. 3.9) в кубоїд $\Omega = \{(x, y, z) \in R^3 : -l \leq x \leq l, -w \leq y \leq w, -h \leq z \leq h\}$ мінімального об'єму.

Тут $n = 3$ – число опуклих багатогранників, $\mathbf{a} = (a_1, a_2, a_3) = (1, 2, 2)$ – "склеювальний" вектор, $\Xi = \{(i, j), a_i \neq a_j, i < j = 1, 2, 3\} = \{(1, 2), (1, 3)\}$, $u_{a_i} = (v_{a_i}, \theta_{a_i}) = (x_{a_i}, y_{a_i}, z_{a_i}, \theta_{a_i}^1, \theta_{a_i}^2, \theta_{a_i}^3)$ – вектор параметрів розміщення $K_i(u_{a_i})$, $i = 1, 2, 3$. Відповідно до "склеювального" вектора, $m = 2$ – число пар опуклих багатогранників, згідно з Ξ , $\tau = (u'^1, u'^2) = (u'_{12}, u'_{13})$ – вектор додаткових змінних, $|\tau| = 3m = 6$, $u = (l, w, h, u_1, u_2, u'_{12}, u'_{13})$ – вектор змінних задачі. Число змінних задачі $\sigma = 3 + 6N + 3m = 21$.

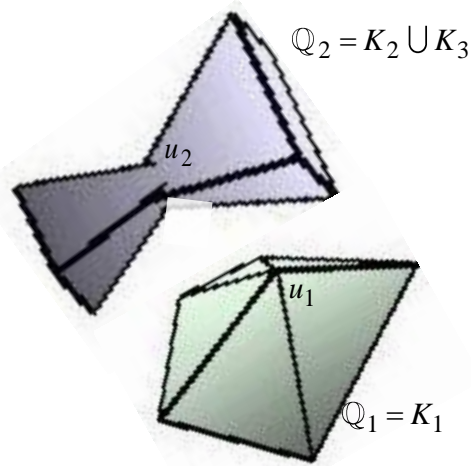


Рисунок 3.9 – Багатогранники $Q_1(u_1) = K_1(u_1)$ і

$$Q_2(u_2) = K_2(u_2) \cup K_3(u_2)$$

Математична модель (3.21)–(3.22) задачі упаковки набуває вигляду

$$\min_{u \in W \subset R^{21}} F(u),$$

$$W = \{u \in R^{21} : \Phi'_{12}(u_1, u_2, u'_{12}) \geq 0, \Phi'_{13}(u_1, u_2, u'_{13}) \geq 0,$$

$$\Phi_1(u_1, p) \geq 0, \Phi_2(u_2, p) \geq 0, \Phi_3(u_2, p) \geq 0\},$$

де $\Phi'_{12}(u_1, u_2, u'_{12})$ – квазі ρ -функція для $K_1(u_1)$ і $K_2(u_2)$,

$\Phi'_{13}(u_1, u_2, u'_{13})$ – квазі ρ -функція для $K_1(u_1)$ і $K_3(u_2)$,

$\Phi_1(u_1, p)$ – ρ -функція для $K_1(u_1)$ і об'єкта Ω^* ,

$\Phi_2(u_2, p)$ – ρ -функція для $K_2(u_2)$ і об'єкта Ω^* ,

$\Phi_3(u_2, p)$ – ρ -функція для $K_3(u_2)$ і об'єкта Ω^* .

Відзначимо, що в моделі можна використовувати дві ρ -функції: ρ -функцію $\Phi_1(u_1)$ і ρ -функцію для опуклої оболонки неопуклого

багатогранника $\mathbb{Q}_2(u_2)$ і об'єкта $\Omega^*(p)$ замість ρ -функцій $\widehat{\Phi}_i(u_{a_i}, p)$, $i=1, 2, 3$, для опуклих багатогранників $K_i(u_{a_i})$, $i=1, 2, 3$ і об'єкта $\Omega^*(p)$. ►

На практиці математична модель (3.21) – (3.22) містить велику кількість змінних і величезне число нерівностей, а саме: $O(n^2)$ нелінійних нерівностей і $O(n^2)$ змінних (з урахуванням додаткових змінних, необхідних для побудови квазі ρ -функцій), де n – число опуклих багатогранників. В результаті пошук локально оптимального розв'язку стає практично нереальною задачею для безпосереднього застосування існуючих сучасних NLP-солверів для $N > 15$, стартуючи з випадкової точки, і для $N > 30$, стартуючи з допустимої точки.

Для пошуку локально оптимальних розв'язків задачі ОРР за розумний час пропонується ефективний метод, який описується в розділі 4.

3.4 Висновки по розділу

Розділ 3 присвячений засобам математичного моделювання обмежень розміщення в задачі ОРР з використанням методу ρ -функцій.

Для моделювання обмежень розміщення (неперетин об'єктів, включення об'єктів в опуклий контейнер, обмежень на мінімально допустимі відстані), побудовані нормалізовані та псевдонормалізовані ρ -функції і псевдонормалізовані квазі ρ -функції для неопуклих багатогранників, а також для неопуклих багатогранників і доповнень до контейнерів, що мають форму кубоїда, кулі, прямого кругового циліндра, еліпсоїда, опуклого багатогранника і довільної опуклої

області, обмеженою циліндричною, еліптичною, сферичною поверхнями або площинами.

Побудовано математичну модель базової задачі оптимальної упаковки довільних багатогранників у опуклому контейнері, границя якого формується циліндричною, сферичною, еліптичною поверхнями та площинами. Наведено її основні властивості. Побудовано математичну модель задачі кластерингу неопуклих багатогранників. Наведено приклад побудови математичної моделі задачі *OPP*.

Результати досліджень, наведені в розділі 3, опубліковані в роботах [31, 36, 38, 40, 41, 42, 43].

Список джерел, які використано у даному розділі наведено у повному списку використаних джерел [72, 74, 209, 210]

4 МЕТОДИ РОЗВ'ЯЗАННЯ ОПТИМІЗАЦІЙНИХ ЗАДАЧ УПАКОВКИ БАГАТОГРАННИКІВ (ОРР)

В цьому розділі розглядається стратегія розв'язання задачі ОРР, яка ґрунтується на методі мультистарту. Пропонується метод побудови стартових точок для задачі оптимальної упаковки довільних багатогранників в різні опуклі контейнери. Цей метод включає в себе алгоритм побудови допустимих параметрів розміщення багатогранників і алгоритм генерації додаткових змінних, які використовуються для побудови квазі ρ -функцій. Наводиться опис методу декомпозиції POLYDEC для пошуку локальних екстремумів, що дозволяє звести задачу ОРР, яка має оцінку числа нерівностей і розмірності $O(n^2)$, до послідовності підзадач нелінійного програмування з оцінкою числа нелінійних нерівностей і розмірності $O(n)$, де n – число опуклих багатогранників, що формують неопуклі багатогранники.

4.1 Стратегія розв'язання задачі ОРР

Стратегія розв'язання задачі (3.21) – (3.22) ґрунтується на методі мультистарту і включає в себе наступні етапи.

Етап 1. Будуємо множину $\{\zeta^0\}$ векторів $\zeta^0 = (p^0, u_1^0, u_2^0, \dots, u_N^0)$ допустимих параметрів розміщення $(u_1^0, u_2^0, \dots, u_N^0)$ багатогранників, розміщених в контейнері $\Omega^0(p^0)$ в задачі (3.21) – (3.22). Для отримання допустимого розв'язку задачі ОРР, наприклад, у роботі [71] застосовано метод апроксимації багатогранників набором δ -паралелепіпедів. В цьому дослідженні запропоновано алгоритм, який детально описаний в підрозділі 4.2.

Етап 2. Шукаємо локальний мінімум функції цілі $F(u)$ в задачі (3.21) – (3.22), стартуючи з кожної точки множини $\{\zeta^0\}$, отриманої на етапі 1. Для пошуку локального мінімуму задачі (3.21) – (3.22) розроблено метод декомпозиції POLYDEC задачі *OPP*, що описаний в підрозділі 4.3.

Етап 3. Вибираємо кращий зі знайдених на етапі 2 локальних мінімумів, як локально оптимальний розв'язок задачі (3.21) – (3.22).

Для розв'язання задач нелінійного програмування (за реалізації етапів 1-2) використовується бібліотека IPOPT [211], яка знаходиться у відкритому доступі на некомерційних репозиторіях програмного забезпечення (див. <https://projects.coin-or.org/Ipopt>).

4.2 Метод побудови допустимих стартових точок (FSPA)

Крок 1. Вибираємо досить великі початкові розміри p^0 контейнера Ω^0 , для того, щоб здійснити розміщення куль S_q^p , $q \in J_N$, всередині контейнера Ω^0 , де $S_q^p = S_q \oplus S^p$ – сума Мінковського для кулі S_q радіуса r_q (рис. 4.1) і кулі S^p радіуса $\rho = 0.5 \max\{\max_{q,g \in J_N} \rho_{qg}, \max_{q \in J_N} \rho_q\}$, за умови, що S_q і S^p мають однакові центри. Зокрема, $p^0 = (l^0, w^0, h^0)$ – для кубоїда; $p^0 = (r^0)$ – для кулі; $p^0 = (\lambda^0)$ – для еліпсоїда, опуклої, багатогранної та довільної областей; $p^0 = (r^0, h^0)$ – для циліндра.

Наприклад, початкове значення кожної змінної вектору p^0 можна встановити $2 \sum_{q=1}^N r_q + (n+1)\rho$.

Крок 2. Генеруємо множину N випадково вибраних центрів (x_q^0, y_q^0, z_q^0) куль S_q^p , $q=1, \dots, N, (q \in J_N)$, із масою M_q , всередині контейнера Ω^0 , за умови $\mu(v) \geq 0$.

Крок 3. Збільшуємо радіус $\beta(r_q + \rho)$ куль S_q^p , $q=1, 2, \dots, N$, починаючи з $\beta=0$ до повного розміру ($\beta=1$). Змінними в цій задачі є центри куль S_q^p і коефіцієнти гомотетії (параметри масштабування) β , де $0 \leq \beta \leq 1$ (рис. 4.2).



Рисунок 4.1 – Неопуклі багатогранники \mathbb{Q}_q і відповідні кулі S_q

Для реалізації кроку 3 розміри контейнера фіксуються p^0 і, стартуючи з точки $v^0 = (x_1^0, y_1^0, z_1^0, \dots, x_N^0, y_N^0, z_N^0, \beta^0 = 0)$, розв'язується така NLP-задача:

$$\max_{v \in W_\beta} \beta, \quad (4.1)$$

$$W_\beta = \{v \in \mathbb{R}^{3N+1} : \hat{\Phi}^{S_q S_g}(v) \geq 0, q < g \in J_N,$$

$$\widehat{\Phi}^{S_q \Omega^*}(v) \geq 0, q \in J_N, 1 - \beta \geq 0, \beta \geq 0, \mu(v) \geq 0 \} \quad (4.2)$$

де $v = (x_1, y_1, z_1, \dots, x_N, y_N, z_N, \beta)$,

$$\widehat{\Phi}^{S_q S_g}(v) = (x_q - x_g)^2 + (y_q - y_g)^2 + (z_q - z_g)^2 - \beta^2 (r_q + 2\rho + r_g)^2, \quad (4.3)$$

де (4.3) – псевдонормалізована рhi-функція для кулі S_q радіуса βr_q і кулі S_g радіуса βr_g ;

$\widehat{\Phi}^{S_q \Omega^*}(v)$ – псевдонормалізована рhi-функція для кулі S_q радіуса βr_q і об'єкта Ω^* ,

$$\widehat{\Phi}^{S_q \Omega^*}(v) = \begin{cases} \widehat{\Phi}^{S_q \mathbf{S}^*}(v), & \Omega = \mathbf{S} \\ \widehat{\Phi}^{S_q \mathbf{C}^*}(v), & \Omega = \mathbf{C} \\ \widehat{\Phi}^{S_q \mathbf{P}^*}(v), & \Omega = \mathbf{P} \\ \widehat{\Phi}^{S_q \mathbf{\Lambda}^*}(v), & \Omega = \mathbf{\Lambda} \\ \widehat{\Phi}^{S_q \mathbf{B}^*}(v), & \Omega = \mathbf{B} \\ \widehat{\Phi}^{S_q \mathbf{E}^*}(v), & \Omega = \mathbf{E} \end{cases}, \quad (4.4)$$

$$\widehat{\Phi}^{S_q \mathbf{S}^*}(v) = (r^0 - \beta(r_q + \rho_q))^2 - (x_q)^2 - (y_q)^2 - (z_q)^2;$$

$$\widehat{\Phi}^{S_q \mathbf{C}^*}(v) = \min\{\Psi_{qt}(v), t = 1, 2, 3\},$$

$$\Psi_{q1}(v) = (r^0 - \beta(r_q + \rho_q))^2 - (x_q)^2 - (y_q)^2,$$

$$\Psi_{q2}(v) = z_q + h^0 - \beta(r_q + \rho_q),$$

$$\Psi_{q3}(v) = -z_q + h^0 - \beta(r_q + \rho_q);$$

$$\widehat{\Phi}^{S_q \mathbf{B}^*}(v) = \min\{\Psi_{qt}(v), t=1, \dots, 6\},$$

$$\Psi_{q1}(v) = -x_q + l^0 - \beta(r_q + \rho_q),$$

$$\Psi_{q2}(v) = x_q + l^0 - \beta(r_q + 2\rho),$$

$$\Psi_{q3}(v) = -y_q + w^0 - \beta(r_q + \rho_q),$$

$$\Psi_{q4}(v) = y_q + w^0 - \beta(r_q + 2\rho),$$

$$\Psi_{q5}(v) = -z_q + h^0 - \beta(r_q + \rho_q),$$

$$\Psi_{q6}(v) = z_q + h^0 - \beta(r_q + 2\rho).$$

$$\widehat{\Phi}^{S_q \mathbf{P}^*}(v) = \min\{\Psi_{qt}(v), t=1, \dots, n_{\mathbf{P}}\}$$

$$\Psi_{qt}(v) = \frac{1}{\sqrt{(A_t^0)^2 + (B_t^0)^2 + (C_t^0)^2}} (A_t^0 x_q + B_t^0 y_q + C_t^0 z_q + D_t^0) - \beta(r_q + \rho_q),$$

$$\widehat{\Phi}^{S_q \mathbf{E}^*}(v) = \min\{\Psi_{qt}(v), t=1, \dots, 6\},$$

$$\Psi_{q1}(v) = -x_q + \widehat{a}^0 - \beta(r_q + \rho_q),$$

$$\Psi_{q2}(v) = x_q + \widehat{a}^0 - \beta(r_q + 2\rho),$$

$$\Psi_{q3}(v) = -y_q + \widehat{b}^0 - \beta(r_q + \rho_q),$$

$$\Psi_{q4}(v) = y_q + \widehat{b}^0 - \beta(r_q + 2\rho),$$

$$\Psi_{q5}(v) = -z_q + \widehat{c}^0 - \beta(r_q + \rho_q),$$

$$\Psi_{q6}(v) = z_q + \hat{c}^0 - \beta(r_q + 2\rho),$$

де $\hat{a}^0, \hat{b}^0, \hat{c}^0$ – параметри паралелепіпеда, вписаного в еліпсоїд $\mathbf{E}^0(a^0 = \lambda^0 a, b^0 = \lambda^0 b, c^0 = \lambda^0 c)$.

Позначимо точку глобального максимуму задачі (4.1) – (4.2) через $v^* = (x_1^*, y_1^*, z_1^*, \dots, x_N^*, y_N^*, z_N^*, \beta^* = 1)$.

Крок 4. Формується вектор допустимих параметрів розміщення $\zeta^0 = (p^0, u_1^0, \dots, u_q^0, \dots, u_N^0)$, припускаючи, що $u_q^0 = (x_q^0, y_q^0, z_q^0, \theta_q^0)$, $(x_q^0, y_q^0, z_q^0) = (x_q^*, y_q^*, z_q^*)$, а θ_q^0 – вектор випадково згенерованих кутів повороту багатогранників $\mathbb{Q}_q, q \in J_N$.

Відзначимо, що глобальний розв'язок задачі (4.1) – (4.2) завжди може бути знайдено (оскільки стартові розміри p^0 контейнера на кроці 1 вибираються досить великими). Розв'язок автоматично задовольняє всі обмеження неперетину, включення і обмеження на допустимі відстані для неопуклих багатогранників.

На рис. 4.2 наведена ілюстрація кроків оптимізаційної процедури для пошуку допустимих параметрів розміщення багатогранників в кубоїд, використовуючи гомотетичне перетворення куль.

Крок 5. Формуємо допустиму стартову точку $u^0 = (\zeta^0, \tau^0)$ для подальшого пошуку локального мінімуму задачі (3.21) – (3.22). Для пошуку вектора τ^0 використовується спеціальна оптимізаційна процедура, описана в підрозділі 4.3.

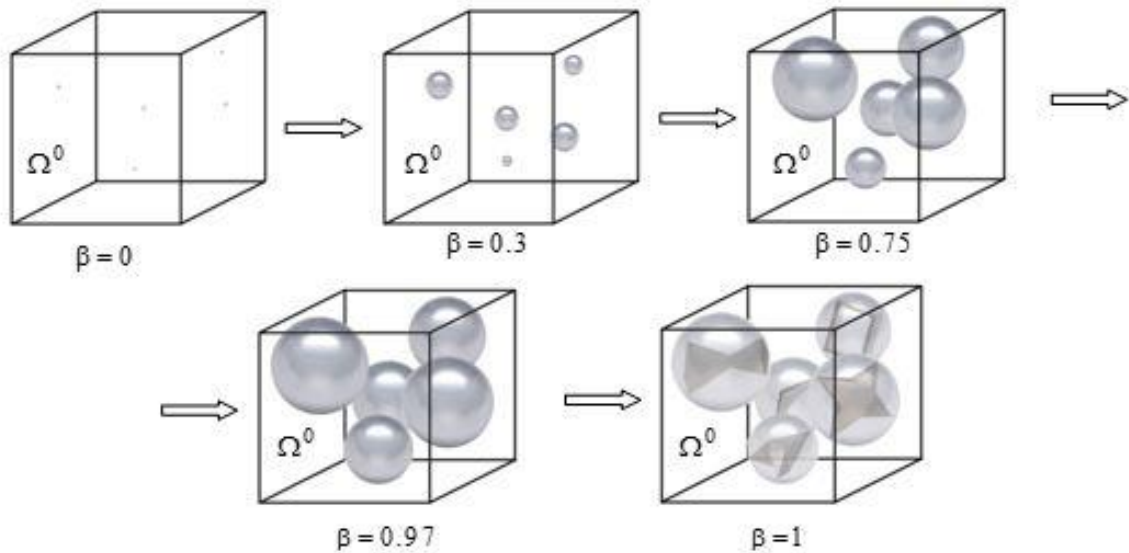


Рисунок 4.2 – Ілюстрація оптимізаційної процедури для пошуку допустимих параметрів розміщення багатогранників

4.3 Метод пошуку локальних екстремумів (POLYDEC)

В основі методу використовується ідея LOFRT процедури, запропонованої в статті [209] для задачі упаковки еліпсів.

Якщо розв'язується задача оптимального кластерингу неопуклих багатогранників і потужність m індексної множини (3.16) менше або дорівнює 5, тоді описаний далі метод декомпозиції не використовується.

Оскільки задача (3.21) – (3.22) не може бути розв'язана за прийнятний час для $N > 30$ за безпосереднього використання сучасних NLP-solvers, навіть стартуючи з допустимої точки, пропонується метод пошуку локального оптимального розв'язку задачі ОРР, метод декомпозиції POLYDEC.

Метод POLYDEC зводить задачу (3.21) – (3.22), яка має велике число нерівностей ($O(n^2)$) і велику розмірність ($O(n^2)$) множини допустимих розв'язків (3.22), до послідовності підзадач нелінійного програмування зі значно меншою кількістю нелінійних нерівностей

($O(n)$) і меншою розмірністю ($O(n)$).

Ключова ідея цього методу полягає в такому.

Для кожної сфери, описаної навколо відповідного опуклого багатогранника будується фіксований індивідуальний кубічний контейнер з центром у точці, яка співпадає з відповідними значеннями вектора допустимих параметрів розміщення багатогранників, отриманих за допомогою алгоритму FSPA. Далі кожній із сфер дозволяється здійснювати вільний рух всередині індивідуального контейнеру. Рух кожної із сфер описується системою з шести лінійних ε -нерівностей. Потім формується підобласть допустимих розв'язків таким чином: додається $O(n)$ ε -нерівностей (для кожної сфери) до системи нерівностей (3.22), що дозволяє видалити $O(n^2)$ ρ -нерівностей для тих пар багатогранників, чії індивідуальні контейнери не перетинаються, та видалити надлишкові обмеження включення багатогранників в контейнер.

Далі здійснюється пошук локального мінімуму на підобласті розмірності $O(n)$. Ця підобласть описується $O(n)$ нелінійними нерівностями. Далі отриманий локальний мінімум використовується як стартова точка для наступної ітерації. На останній ітерації алгоритму знаходиться локальний мінімум задачі (3.21) – (3.22).

Розглянемо метод декомпозиції POLYDEC більш детально.

Припускаємо, що побудовані кулі $S_q^0 \equiv S_q(0)$ радіуса r_q і центром $v_q = (x_q, y_q, z_q)$, описані навколо кожного неопуклого багатогранника \mathbb{Q}_q , $q \in J_N$, також як і кулі $S_i^0 \equiv S_i(0)$ радіуса r_i і центром $v_{ci} = (x_{ci}, y_{ci}, z_{ci})$, описані навколо кожного опуклого багатогранника K_i^0 , $i = 1, \dots, n$.

Алгоритм, який реалізує метод POLYDEC – це ітераційна

процедура, що включає такі кроки.

Крок 1. Припускаємо $k = 1$ і задаємо вектор $\zeta^{k-1} = (p^{k-1}, u_1^{k-1}, \dots, u_N^{k-1})$ допустимих параметрів розміщення багатогранників \mathbb{Q}_q , що розміщуються всередині контейнера Ω^{k-1} .

Крок 2. Формуємо відповідний вектор $(v_{c1}^{(k-1)}, \dots, v_{cn}^{(k-1)})$ центрів куль $S_i(u_{a_i}^{(k-1)})$, $i \in I_n$. Відповідно до вектора \mathbf{a} , центр v_{ci} кулі $S_i \supset K_i$ після трансляції та повороту вихідного опуклого багатогранника K_i^0 набуває вигляду

$$v_{ci}^{(k-1)} = v_{ci}(u_{a_i}^{(k-1)}) = v_{a_i}^{(k-1)} + M(\theta_{a_i}^{(k-1)}) \cdot v_{ci}.$$

На рис. 4.3 зображений неопуклий багатогранник $\mathbb{Q}(0) = K_1(0) \cup K_2(0)$ з вектором трансляції $(0, 0, 0)$ і кутами повороту $\theta = (0, 0, 0)$. Кулі $S_1(0)$ і $S_2(0)$, описані навколо $K_1(0)$ і $K_2(0)$, мають центри v_{c1} і v_{c2} . Багатогранник $\mathbb{Q}(0)$, що транслюється на вектор v^0 і повернутий на кут $\theta^0 = (\theta_1^0, \theta_2^0, \theta_3^0)$, позначається як $\mathbb{Q}(u^0) = K_1(u^0) \cup K_2(u^0)$, де $u^0 = (v^0, \theta^0)$. Центри куль $S_1(u^0)$ і $S_2(u^0)$ позначаються як v_{c1}^0 і v_{c2}^0 .

Крок 3. Для кожної кулі $S_i(u_{a_i}^{(k-1)})$ будується фіксований індивідуальний контейнер $\Omega_i^k \supset S_i \supset K_i$ з рівними напівсторонами довжини $r_i + \varepsilon$, $i \in I_n$, і центром симетрії в точці $v_{ci}^{(k-1)}$, припускаючи,

$$\text{що } \varepsilon = \sum_{i=1}^n r_i / n.$$

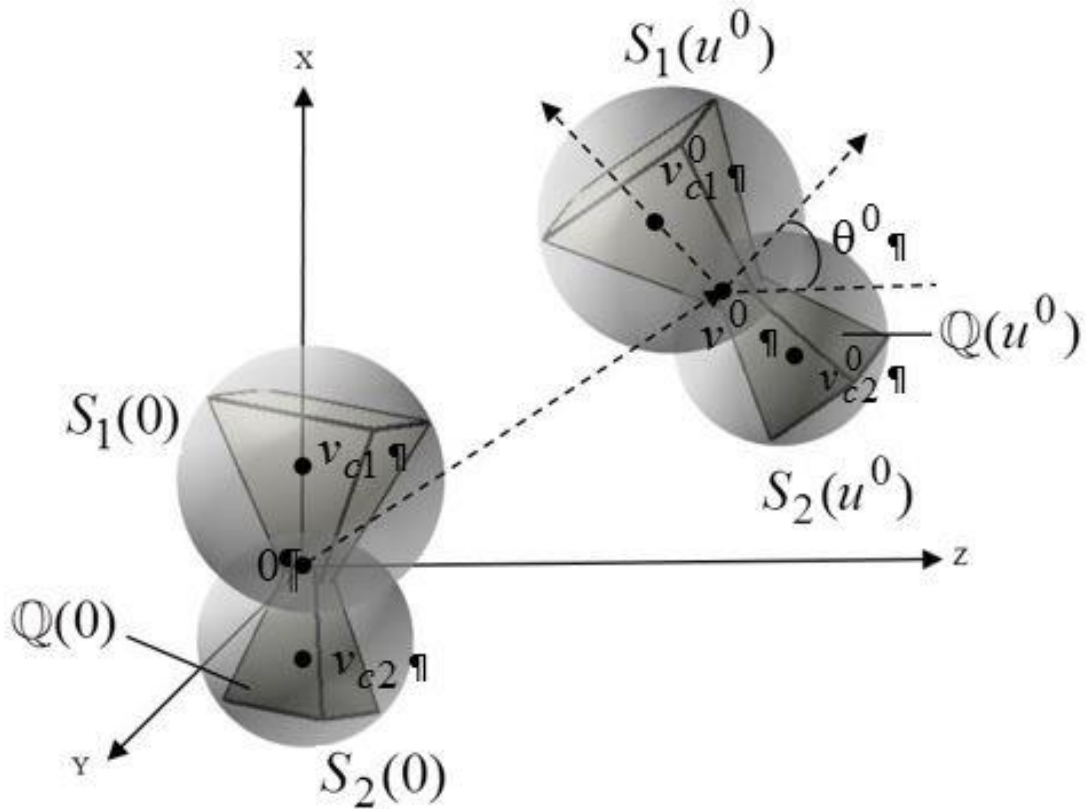


Рисунок 4.3 – Параметри розміщення $S_1(u^0)$ і $S_2(u^0)$ неопуклого багатогранника $Q(0) = K_1(0) \cup K_2(0)$, що транслюється на вектор v^0 і повернутий на кут θ^0 .

Вершини індивідуального контейнера Ω_i^k задаються так:

$$\begin{aligned}
 v_{\Omega_i}^{1(k)} &= (x_i^{(k-1)} + r_i + \varepsilon, y_i^{(k-1)} + r_i + \varepsilon, z_i^{(k-1)} + r_i + \varepsilon), \\
 v_{\Omega_i}^{2(k)} &= (x_i^{(k-1)} - (r_i + \varepsilon), y_i^{(k-1)} + r_i + \varepsilon, z_i^{(k-1)} + r_i + \varepsilon), \\
 v_{\Omega_i}^{3(k)} &= (x_i^{(k-1)} - (r_i + \varepsilon), y_i^{(k-1)} + r_i + \varepsilon, z_i^{(k-1)} - (r_i + \varepsilon)), \\
 v_{\Omega_i}^{4(k)} &= (x_i^{(k-1)} + r_i + \varepsilon, y_i^{(k-1)} + r_i + \varepsilon, z_i^{(k-1)} - (r_i + \varepsilon)), \\
 v_{\Omega_i}^{5(k)} &= (x_i^{(k-1)} + r_i + \varepsilon, y_i^{(k-1)} - (r_i + \varepsilon), z_i^{(k-1)} + r_i + \varepsilon),
 \end{aligned}$$

$$v_{\Omega_i}^{6(k)} = (x_i^{(k-1)} - (r_i + \varepsilon), y_i^{(k-1)} - (r_i + \varepsilon), z_i^{(k-1)} + r_i + \varepsilon),$$

$$v_{\Omega_i}^{7(k)} = (x_i^{(k-1)} - (r_i + \varepsilon), y_i^{(k-1)} - (r_i + \varepsilon), z_i^{(k-1)} - (r_i + \varepsilon)),$$

$$v_{\Omega_i}^{8(k)} = (x_i^{(k-1)} + r_i + \varepsilon, y_i^{(k-1)} - (r_i + \varepsilon), z_i^{(k-1)} - (r_i + \varepsilon)).$$

На рис. 4.4 зображені індивідуальні контейнери $\Omega_1(u_{a_1}^0)$ і $\Omega_2(u_{a_2}^0)$ для куль $S_1(u_{a_1}^0)$ і $S_2(u_{a_2}^0)$, розглянутих у прикладі вище. Відзначимо, що тут $a_1 = a_2$.

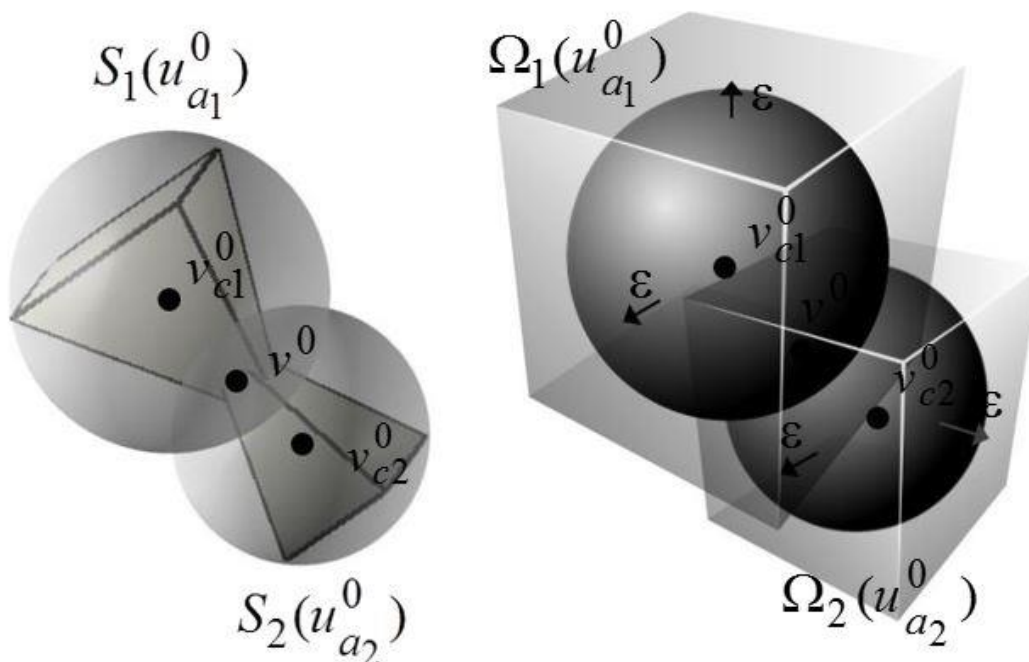


Рисунок 4.4 – Індивідуальні контейнери $\Omega_1(u_{a_1}^0)$ і $\Omega_2(u_{a_2}^0)$ для куль

$S_1(u_{a_1}^0)$ і $S_2(u_{a_2}^0)$.

Крок 4. Кожна куля S_i , пов'язана з опуклим багатогранником K_i , може здійснювати рух тільки всередині відповідного фіксованого

індивідуального контейнера Ω_i^k (побудованого на кроці 3). Отже, будемо phi-функцію $\Phi^{S_i\Omega_i^*}$ для кожної кулі S_i і об'єкта $\Omega_i^* = R^3 \setminus \text{int } \Omega_i$:

$$\begin{aligned} \Phi^{S_i\Omega_i^*}(v_{a_i}, v_{a_i}^{(k-1)}) = \min\{-x_{ci}(u_{a_i}^{(k-1)}) + x_{a_i} + \varepsilon, \\ -y_{ci}(u_{a_i}^{(k-1)}) + y_{a_i} + \varepsilon, -z_{ci}(u_{a_i}^{(k-1)}) + z_{a_i} + \varepsilon, \\ x_{ci}(u_{a_i}^{(k-1)}) - x_{a_i} + \varepsilon, y_{ci}(u_{a_i}^{(k-1)}) - y_{a_i} + \varepsilon, z_{ci}(u_{a_i}^{(k-1)}) - z_{a_i} + \varepsilon\}. \end{aligned}$$

Нерівність $\Phi^{S_i\Omega_i^*}(v_{a_i}, v_{a_i}^{(k-1)}) \geq 0$ забезпечує $S_i \subset \Omega_i^k$ і може бути описана системою, що складається з шести лінійних "ε-обмежень"

$$\begin{aligned} -x_{ci}(u_{a_i}^{(k-1)}) + x_{a_i} + \varepsilon \geq 0, -y_{ci}(u_{a_i}^{(k-1)}) + y_{a_i} + \varepsilon \geq 0, \\ -z_{ci}(u_{a_i}^{(k-1)}) + z_{a_i} + \varepsilon \geq 0, x_{ci}(u_{a_i}^{(k-1)}) - x_{a_i} + \varepsilon \geq 0, \\ y_{ci}(u_{a_i}^{(k-1)}) - y_{a_i} + \varepsilon \geq 0, z_{ci}(u_{a_i}^{(k-1)}) - z_{a_i} + \varepsilon \geq 0. \end{aligned}$$

Вводимо додаткову (штучну) підмножину Λ_k^ε додаткових "ε-обмежень" на вектори трансляції $v_{a_i} = (x_{a_i}, y_{a_i}, z_{a_i})$, $i \in I_n$ опуклих багатогранників K_i , $i \in I_n$

$$\begin{aligned} \Lambda_k^\varepsilon = \{u \in R^\sigma : -x_{ci}(u_{a_i}^{(k-1)}) + x_{a_i} + \varepsilon \geq 0, -y_{ci}(u_{a_i}^{(k-1)}) + y_{a_i} + \varepsilon \geq 0, \\ -z_{ci}(u_{a_i}^{(k-1)}) + z_{a_i} + \varepsilon \geq 0, x_{ci}(u_{a_i}^{(k-1)}) - x_{a_i} + \varepsilon \geq 0, \end{aligned}$$

$$y_{ci}(u_{a_i}^{(k-1)}) - y_{a_i} + \varepsilon \geq 0, z_{ci}(u_{a_i}^{(k-1)}) - z_{a_i} + \varepsilon \geq 0, i \in I_n \}.$$

Далі додаємо систему нерівностей з bn додаткових лінійних " ε -обмежень", які описують підмножину Λ_k^ε до системи нерівностей, що визначає множину допустимих розв'язків W і отримуємо k -у підобласть вигляду $W_k = W \cap \Lambda_k^\varepsilon$.

Слід зазначити, що система нерівностей, яка описує підмножину допустимих розв'язків W_k , в більшості випадків включає в себе $O(n^2)$ надлишкових рhі-нерівностей.

Крок 5. Для того щоб уникнути надмірних рhі-нерівностей, які описують W_k , формуються спеціальні індексні множини Ξ_1^k і Ξ_2^k , що включають в себе індекси тих пар об'єктів, які пов'язані з відповідними "ненадлишковими" обмеженнями включення і неперетину.

Для формування індексної множини Ξ_1^k необхідно виключити з Ξ (3.16) індекси тих пар опуклих багатогранників, чий індивідуальні контейнери не перетинаються, тобто

$$\Xi_1^k = \{(i, j) \in \Xi_1^{kS} : \varphi^{\Omega_i^k \Omega_j^k}(v_{a_i}^{(k-1)}, v_{a_j}^{(k-1)}) < 0\}, \text{ де}$$

$$\Xi_1^{kS} = \{(i, j) \in \Xi : \hat{\Phi}^{S_{a_i} S_{a_j}}(v_{a_i}^{(k-1)}, v_{a_j}^{(k-1)}) < 0\},$$

$$\varphi^{\Omega_i^k \Omega_j^k}(v_{a_i}^{(k-1)}, v_{a_j}^{(k-1)}) = \max\{\varphi_{ij}^s(v_{a_i}^{(k-1)}, v_{a_j}^{(k-1)}), s = 1, \dots, 6\},$$

$$\varphi_{ij}^1(v_{a_i}^{(k-1)}, v_{a_j}^{(k-1)}) = (x_i^{(k-1)} - x_j^{(k-1)}) - R_{ij},$$

$$\varphi_{ij}^2(v_{a_i}^{(k-1)}, v_{a_j}^{(k-1)}) = (y_i^{(k-1)} - y_j^{(k-1)}) - R_{ij},$$

$$\varphi_{ij}^3(v_{a_i}^{(k-1)}, v_{a_j}^{(k-1)}) = (z_i^{(k-1)} - z_j^{(k-1)}) - R_{ij},$$

$$\varphi_{ij}^4(v_{a_i}^{(k-1)}, v_{a_j}^{(k-1)}) = -(x_i^{(k-1)} - x_j^{(k-1)}) - R_{ij},$$

$$\varphi_{ij}^5(v_{a_i}^{(k-1)}, v_{a_j}^{(k-1)}) = -(y_i^{(k-1)} - y_j^{(k-1)}) - R_{ij},$$

$$\varphi_{ij}^6(v_{a_i}^{(k-1)}, v_{a_j}^{(k-1)}) = -(z_i^{(k-1)} - z_j^{(k-1)}) - R_{ij},$$

$$R_{ij} = (r_i + r_j) + \rho_{ij} + 2\varepsilon,$$

$\widehat{\Phi}^{S_{a_i} S_{a_j}}(v_{a_i}^{(k-1)}, v_{a_j}^{(k-1)})$ – псевдонормалізована рхі-функція (4.3)

для пари куль S_q і S_g ($a_i = q$, $a_j = g$), описаних навколо неопуклих багатогранників $\mathbb{Q}_q(u_q^{(k-1)}) \supset K_i(u_q^{(k-1)})$ і $\mathbb{Q}_g(u_g^{(k-1)}) \supset K_j(u_g^{(k-1)})$.

Відзначимо, що, якщо $(i, j) \notin \Xi_1^k$, то немає необхідності перевіряти обмеження на допустимі відстані (або обмеження неперетину) для відповідних пар багатогранників $K_i(u_{a_i}^{(k-1)})$ і $K_j(u_{a_j}^{(k-1)})$. Якщо мінімально допустимі відстані не задані, тоді функція $\varphi^{\Omega_i^k \Omega_j^k}(v_{a_i}^{(k-1)}, v_{a_j}^{(k-1)})$ стає рхі-функцією для двох орієнтованих паралелепіпедів Ω_i і Ω_j .

Далі наведено приклад формування індексної множини Ξ_1^k в методі POLYDEC.

Приклад 4.1 Нехай $N = 3$ – неопуклі багатогранники, розміщені всередині контейнера Ω^k на k -й ітерації процедури, що реалізує метод POLYDEC (рис. 4.5)

Тоді індексна множина Ξ , визначена в (3.16), має вигляд:

$$\Xi = \{(1, 3), (1, 4), (1, 5), (1, 6), (2, 3), (2, 4), (2, 5), (2, 6), (3, 5), (3, 6), (4, 5), (4, 6)\}.$$

Для того щоб визначити множину Ξ_1^k , перш за все, побудуємо індексну множину Ξ_1^{kS} (рис 4.5. а)

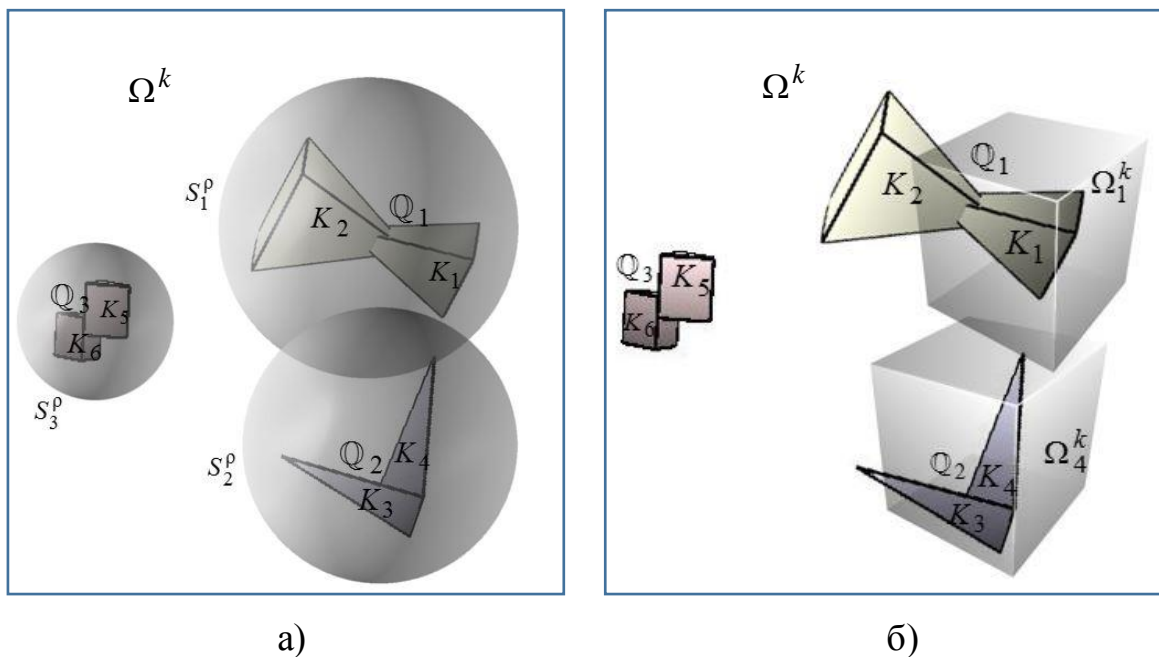


Рисунок 4.5 – Побудова індексної множини: а) Ξ_1^{kS} , б) Ξ_1^k на k -й ітерації процедури, що реалізує метод POLYDEC

$$\Xi_1^{kS} = \{(i, j) \in \Xi : \hat{\Phi}^{S_{a_i} S_{a_j}}(v_1^{(k-1)}, v_2^{(k-1)}) < 0\} = \{(1, 3), (1, 4), (2, 3), (2, 4)\}.$$

Це означає, що тільки S_1^p і S_2^p (для неопуклих багатогранників Q_1 і Q_2) перетинаються, тобто $\hat{\Phi}^{S_1 S_2}(v_1^{(k-1)}, v_2^{(k-1)}) < 0$, а отже, має сенс розглядати тільки можливі перетини таких опуклих багатогранників: K_1 і K_3 , K_1 і K_4 , K_2 і K_3 , K_2 і K_4 .

Далі формуємо індексну множину Ξ_1^k (рис 4.5 б)

$$\Xi_1^k = \{(i, j) \in \Xi_1^{kS} : \varphi^{\Omega_i^k \Omega_j^k}(v_{a_i}^{(k-1)}, v_{a_j}^{(k-1)}) < 0\} = \{(1, 4)\}.$$

Це означає, що тільки індивідуальні контейнери Ω_1^k і Ω_4^k опуклих багатогранників K_1 і K_4 мають непорожній перетин, тобто $\varphi^{\Omega_1^k \Omega_4^k}(v_1^{(k-1)}, v_2^{(k-1)}) < 0$, а отже, в k -ту підзадачу необхідно включити лише квазі ρ -функції для багатогранників K_1 і K_4 . ►

Для того, щоб сформулювати індексну множину Ξ_2^k , необхідно виключити з (3.22) всі ρ -нерівності обмежень включення для опуклих багатогранників, чий індивідуальні контейнери не перетинаються з множиною $\Omega_\varepsilon^{k*} = R^3 \setminus \text{int } \Omega_\varepsilon^k$, зокрема:

– якщо контейнер Ω^k має форму кубоїда, то

$$\Omega_\varepsilon^k = \{(x, y, z) : \varepsilon \leq x \leq l^{(k-1)} - \varepsilon, \varepsilon \leq y \leq w^{(k-1)} - \varepsilon, \varepsilon \leq z \leq h^{(k-1)} - \varepsilon\};$$

– якщо контейнер Ω^k має форму кулі, то

$$\Omega_\varepsilon^k = \{(x, y, z) : x^2 - y^2 - z^2 + (r^{(k-1)} - \varepsilon)^2 \geq 0\}.$$

– якщо контейнер Ω^k має форму циліндра, то

$$\Omega_\varepsilon^k = \{(x, y, z) : \min\{(r - \varepsilon)^2 - x^2 - y^2, -z + h - \varepsilon, z + h - \varepsilon\} \geq 0\},$$

– якщо контейнер Ω^k має форму еліпсоїда, то

$$\Omega_\varepsilon^k = \{(x, y, z) : 1 - \frac{x^2}{(a-\varepsilon)^2} - \frac{y^2}{(b-\varepsilon)^2} - \frac{z^2}{(c-\varepsilon)^2} \geq 0\},$$

Таким чином, $\Xi_2^k = \{i \in \Xi_2^{kS} : \widehat{\Phi}^{\Omega_i^k \Omega_\varepsilon^*}(v_{a_i}^{(k-1)}) < 0\}$, де

$\widehat{\Phi}^{\Omega_i^k \Omega_\varepsilon^*}(v_{a_i}^{(k-1)})$ – псевдонормалізована ρ -функція для багатогранника

$K_i(u_{a_i}^{(k-1)})$ і об'єкта Ω_ε^{k*} , $\Xi_2^{kS} = \{i \in I_n : \widehat{\Phi}^{S_{a_i} \Omega_\varepsilon^{k*}}(v_{a_i}^{(k-1)}) < 0\}$, $\widehat{\Phi}^{S_{a_i} \Omega_\varepsilon^{k*}}$ –

псевдонормалізована ρ -функція (4.4) для кулі S_q , пов'язаної з

неопуклим багатогранником $\mathbb{Q}_q(u_q^{(k-1)}) \supset K_i(u_q^{(k-1)})$, і об'єктом Ω_ε^* ,

$a_i = q$.

Зокрема:

– якщо Ω_ε^{k*} – кубоїд, то псевдонормалізована ρ -функція для об'єктів Ω_ε^{k*} та Ω_i^k

$$\widehat{\Phi}^{\Omega_i^k \Omega_\varepsilon^{k*}}(v_{a_i}^{(k-1)}) = \min\{\Psi_{it}(v_{a_i}^{(k-1)}), t = 1, \dots, 6\},$$

$$\Psi_{i1}(v_{a_i}^{(k-1)}) = x_i^{(k-1)} + l^{(k-1)} - R_i,$$

$$\Psi_{i2}(v_{a_i}^{(k-1)}) = y_i^{(k-1)} + w^{(k-1)} - R_i,$$

$$\Psi_{i3}(v_{a_i}^{(k-1)}) = z_i^{(k-1)} + h^{(k-1)} - R_i,$$

$$\Psi_{i4}(v_{a_i}^{(k-1)}) = -x_i^{(k-1)} + l^{(k-1)} - R_i,$$

$$\Psi_{i5}(v_{a_i}^{(k-1)}) = -y_i^{(k-1)} + w^{(k-1)} - R_i,$$

$$\Psi_{i6}(v_{a_i}^{(k-1)}) = -z_i^{(k-1)} + h^{(k-1)} - R_i,$$

$$R_i = r_i + \rho_q + 2\varepsilon.$$

– якщо Ω_ε^{k*} – куля, то псевдонормалізована рхі-функція для об'єктів Ω_ε^{k*} та Ω_i^k

$$\widehat{\Phi}^{\Omega_i^k \Omega_\varepsilon^{k*}}(v_{a_i}^{(k-1)}) = \min_{s=1, \dots, 8} ((x_{is}^{(k-1)})^2 + (y_{is}^{(k-1)})^2 + (z_{is}^{(k-1)})^2 - (r^{(k-1)} - R_i)^2),$$

де $(x_{is}^{(k-1)}, y_{is}^{(k-1)}, z_{is}^{(k-1)})$ – координати s -ої вершини індивідуального контейнера з центром в точці $v_{a_i}^{(k-1)}$

– якщо Ω_ε^{k*} – циліндр, то псевдонормалізована рхі-функція для об'єктів Ω_ε^{k*} та Ω_i^k

$$\widehat{\Phi}^{\Omega_i^k \Omega_\varepsilon^{k*}}(v_{a_i}^{(k-1)}) = \min\{\Psi_{it}(v_{a_i}^{(k-1)}), t = 1, 2, 3\},$$

$$\Psi_{i1}(v_{a_i}^{(k-1)}) = \min_{s=1, \dots, 8} ((r^{(k-1)} - R_i)^2 - (x_{is}^{(k-1)})^2 - (y_{is}^{(k-1)})^2),$$

$$\Psi_{i2}(v_{a_i}^{(k-1)}) = z_i^{(k-1)} + h^{(k-1)} - R_i,$$

$$\Psi_{i3}(v_{a_i}^{(k-1)}) = -z_i^{(k-1)} + h^{(k-1)} - R_i.$$

– якщо Ω_ε^{k*} – еліпсоїд, то псевдонормалізована рхі-функція для об'єктів Ω_ε^{k*} та Ω_i^k

$$\widehat{\Phi}^{\Omega_i^k \Omega_\varepsilon^{k*}}(v_{a_i}^{(k-1)}) = \min_{s=1, \dots, 8} \left(1 - \frac{(x_{is}^{(k-1)})^2}{(a-R_i)^2} + \frac{(y_{is}^{(k-1)})^2}{(b-R_i)^2} + \frac{(z_{is}^{(k-1)})^2}{(c-R_i)^2}\right).$$

Відзначимо, що, якщо $i \notin \Xi_2^k$, то немає необхідності перевіряти обмеження на допустимі відстані (або обмеження включення) для багатогранника $K_i(u_{a_i}^{(k-1)})$ і об'єкта Ω_ε^k .

Крок 6. Генеруємо k -у підзадачу на підмножині розв'язків $W_k = W \cap \Lambda_k^\varepsilon$ з видаленими надлишковими ρ -нерівностями і меншої розмірності ($O(n)$)

$$\min_{u_{w_k} \in W_k \subset R^{\sigma - \sigma_k}} F(u_{w_k}), \quad (4.5)$$

$$W_k = \{u_{w_k} = (\zeta, \tau_{w_k}) \in R^{\sigma - \sigma_k} : \widehat{\Phi}'_{ij}(u_{a_i}, u_{a_j}) \geq 0, (i, j) \in \Xi_1^k,$$

$$\widehat{\Phi}_i(u_{a_i}, p) \geq 0, i \in \Xi_2^k, \Phi^{S_i \Omega_i^{k*}}(u_{a_i}) \geq 0, i \in I_n,$$

$$f(p^{(k-1)}, \varepsilon) \geq 0, \mu(u_{w_k}, p^{(k-1)}) \geq 0\},$$

де Ξ_1^k і Ξ_2^k визначені на кроці 5;

$\sigma_k = 3(m - \text{card}(\Xi_1^k))$ – число всіх видалених додаткових змінних, які зустрічаються у відповідних надлишкових ρ -нерівностях, $\sigma - \sigma_k = |p| + 6N + \text{card}(\Xi_1^k)$, $\text{card}(\Xi_1^k)$ порядку ($O(n)$);

$f(p^{(k-1)}, \varepsilon) \geq 0$ – обмеження на розміри контейнера Ω^k на k -й ітерації.

Крок 7. Генеруємо допустиму стартову точку $u^{(k-1)} = (\zeta^{(k-1)}, \tau_{w_k}^{(k-1)})$ для задачі (4.5). Оскільки вектор $\zeta^{(k-1)}$ вже визначено, необхідно знайти значення вектора додаткових змінних $\tau_{w_k}^{(k-1)} = (u'^{(k-1)1}, \dots, u'^{(k-1)s}, \dots, u'^{(k-1)m})$ для таких $s \in \{1, \dots, m\}$, що

$(i, j) \in \Xi_1^k$.

Для визначення вектора $u'^{(k-1)s}$ використовуємо наступний алгоритм.

Основна ідея алгоритму полягає в такому: визначаємо вектор $u'^{(k-1)s}$ як вектор допустимих параметрів, що розділяє площини для двох куль $S_i(u_{a_i}^{(k-1)})$ і $S_j(u_{a_j}^{(k-1)})$, якщо $\widehat{\Phi}^{S_i S_j} \geq 0$, за допомогою простих геометричних обчислень, в іншому випадку знаходимо вектор $u'^{(k-1)s}$, розв'язуючи таку додаткову підзадачу:

$$\max_{(u'^s, \alpha) \in W'_\alpha} \alpha, \quad (4.6)$$

де

$$W'_\alpha = \{(u'^s, \alpha) \in R^4 : \widehat{\Phi}'_{ij}(u_{a_i}^{(k-1)}, u_{a_j}^{(k-1)}, u'^s) - \alpha \geq 0\},$$

$\alpha \in R^1, u'^s = (\theta^{1s}, \theta^{2s}, \xi^s)$, за фіксованих параметрів $(u_{a_i}^{(k-1)}, u_{a_j}^{(k-1)})$, які містяться у відповідній псевдонормалізованій квазі phi-функції $\widehat{\Phi}'_{ij}(u_{a_i}^{(k-1)}, u_{a_j}^{(k-1)}, u'^s), \forall (i, j) \in \Xi$.

На рис. 4.6 зображені два випадки для отримання вектора допустимих параметрів u' , що розділяє площини: а) для двох куль S_i і S_j , якщо $\text{int } S_i \cap \text{int } S_j = \emptyset$; б) для двох опуклих багатогранників K_i і K_j , якщо $\text{int } S_i \cap \text{int } S_j \neq \emptyset$.

Для випадку а) використовуються тривіальні геометричні обчислення з метою знайти u' ; для випадку б) розв'язується NLP

підзадача (4.6) з метою знайти невід'ємні значення, що відповідає задачі пошуку невід'ємних значень квазі ρ -функції для двох опуклих багатогранників K_i і K_j .

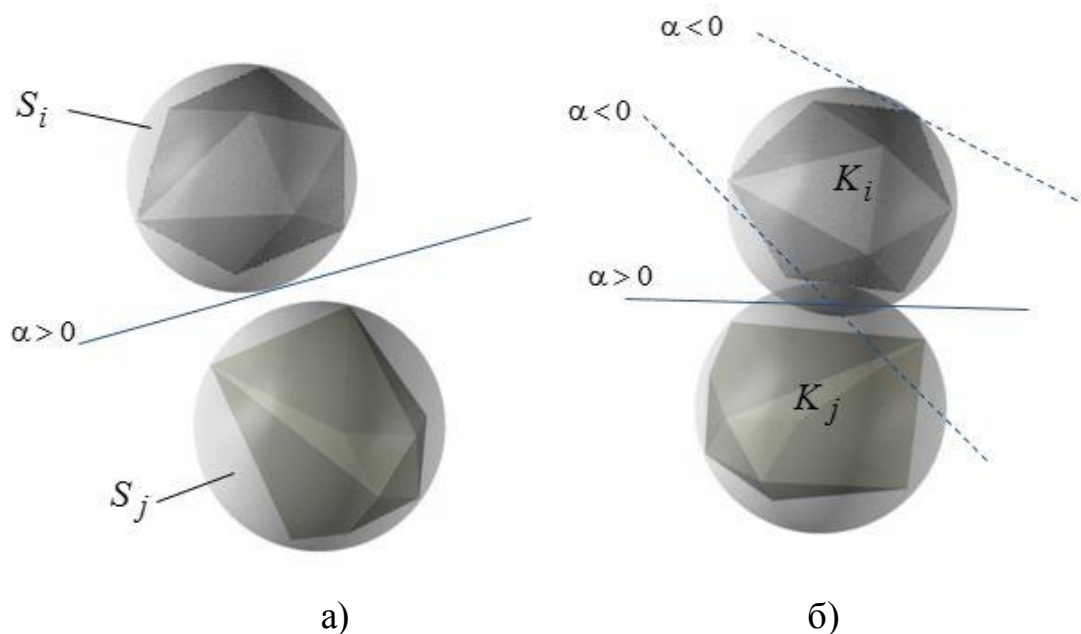


Рисунок 4.6 – Зображення 7-го кроку на k -й ітерації процедури:

а) $S_i \cap S_j = \emptyset$; б) $\text{int } S_i \cap \text{int } S_j \neq \emptyset$.

Слід зазначити, що будь-яке позитивне значення α в (4.6) забезпечує допустимі значення u^S .

Таким чином, всі псевдонормалізовані квазі ρ -функції і ρ -функції в (4.5) в точці $u^{(k-1)}$ набувають невід'ємних значень.

Крок 8. Розв'язуємо підзадачу (4.5), стартуючи з допустимої точки $u^{(k-1)}$

$$\min_{u_{w_k} \in W_k \subset R^{\sigma-\sigma_k}} F(u_{w_k}), \quad (4.7)$$

і знаходимо точку локального мінімуму $u_{w_k}^* = (\zeta^{*k}, \tau_{w_k}^{*k})$.

Якщо точка $u_{w_k}^*$ локального мінімуму підзадачі (4.7) належить границі додаткової підмножини Λ_k^ε , тобто $u_{w_k}^* \in fr\Lambda_k^\varepsilon$, то приймаємо ζ^{*k} як стартовий вектор ζ^k для наступної ітерації процедури (встановлюємо $k = k + 1$ і переходимо до кроку 2), в іншому випадку зупиняємо оптимізаційну процедуру.

Точка $u^* = u^{*k} = (\zeta^{*k}, \tau^{*k}) \in R^\sigma$ є точкою локального мінімуму задачі (3.21) – (3.22). Тут τ^{*k} включає в себе вектор $\tau_{w_k}^{*k}$, а також вектор σ_k додаткових змінних, видалених на k -й ітерації. Зауважимо, що видалені раніше додаткові змінні не визначаються, оскільки на останньому кроці алгоритму значення додаткових змінних не впливають на значення цільової функції, тобто $F(u_{w_k}^*) = F(u^{*k})$.

На рис. 4.7 зображена діаграма ітераційної процедури методу декомпозиції POLYDEC під час розв'язання задачі (3.21) – (3.22). Проілюстрована процедура розв'язання послідовності підзадач, заданих в (4.7) для $k = 2, 3, 4$. Нехай $u^{(0)}$ – допустима стартова точка. Проекція кожної додаткова (штучна) множина Λ_k^ε , описана на кроці 4 процедури, та зображена у вигляді квадрата з центром в точці $u^{(k-1)}$, $k = 1, 2, 3, 4$.

Приймаємо допустиму точку $u^{(0)}$, формуємо множину Λ_1^ε з центром $u^{(0)}$, розв'язуємо підзадачу (4.7) на підобласті $W_1 = \Lambda_1^\varepsilon \cap W$ і отримуємо точку локального мінімуму $u_{w_1}^*$. Точка $u_{w_1}^*$ належить границі множини Λ_1^ε , отже, формуємо наступну множину Λ_2^ε з центром $u^{(1)} = u_{w_1}^*$ і шукаємо точку локального мінімуму $u_{w_2}^*$ підзадачі (4.7) на

підобласті $W_2 = \Lambda_2^\varepsilon \cap W$. Точка $u_{w_2}^*$ належить границі множини Λ_2^ε , отже, формуємо наступну множину Λ_3^ε з центром $u^{(2)} = u_{w_2}^*$ і шукаємо точку локального мінімуму $u_{w_3}^*$ підзадачі (4.7) на підобласті $W_3 = \Lambda_3^\varepsilon \cap W$. Точка $u_{w_3}^*$ належить внутрішності множини Λ_3^ε , тобто $u_{w_3}^* \in \text{int } \Lambda_3^\varepsilon$, отже, зупиняємо процедуру оптимізації. Точка $u_{w_3}^* = u^*$ є точкою локального мінімуму задачі (3.21) – (3.22).

На рис. 4.8 наведена ілюстрація ітеративної процедури упаковки неопуклих багатогранників в циліндрі, яка відповідає діаграмі рис. 4.7.

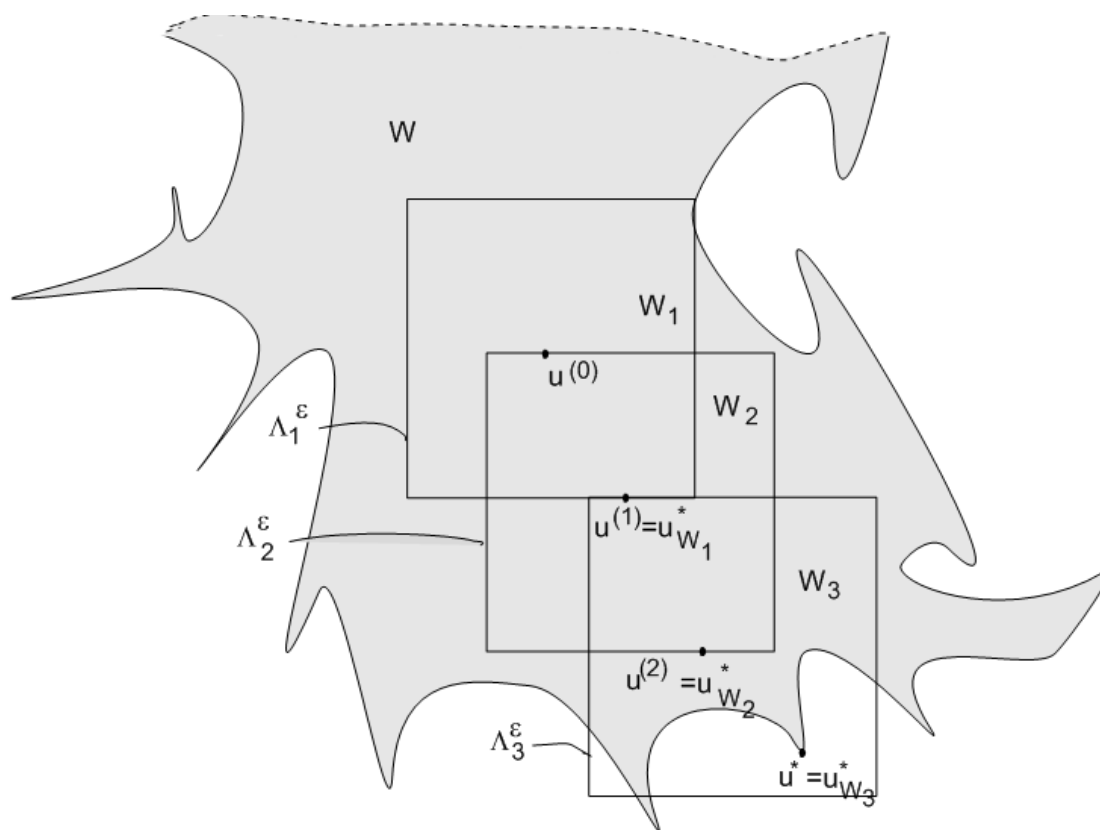


Рисунок 4.7 – Діаграма методу декомпозиції POLYDEC

Відзначимо, що $\text{dist}(u_{w_k}^*, u_{w_{k+1}}^*) \geq \varepsilon$, якщо $u_{w_{k+1}}^* \in \text{fr } \Lambda_k^\varepsilon$. Крім того, прийняте значення ε істотно більшим, ніж точність IPOPT (10^{-8}).

Таким чином, можна зробити висновок, що умова зупинки процедури завжди досягається за кінцеве число ітерацій.

Якщо IPOPT не «справляється» зі знаходженням локального мінімуму підзадачі (4.5) за прийнятний час, ми ділимо значення ε на два і запускаємо алгоритм, що реалізує метод POLYDEC. Якщо локальний мінімум знайдений за половини значення ε , тоді відновлюємо початкове значення ε і продовжуємо ітераційну процедуру для нової допустимої стартової точки, в іншому випадку зупиняємо процедуру.

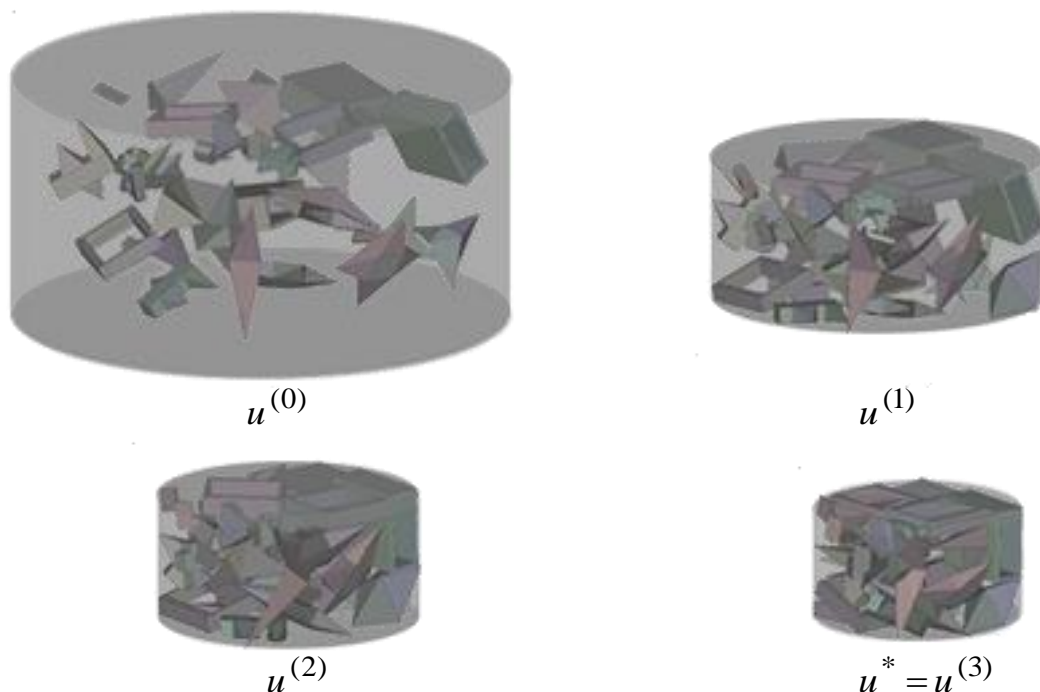


Рисунок 4.8 – Розташування неопуклих багатогранників в кубоді, що відповідає послідовності допустимих точок $u^{(0)}$, $u^{(1)}$, $u^{(2)}$, $u^* = u^{(3)}$

Запропонований алгоритм, в більшості випадків, здійснює перевірку умов неперетину для значно меншої кількості пар багатогранників, ніж m (тут m – число всіх пар опуклих багатогранників, що розглядаються в задачі (3.21) – (3.22)), оскільки для кожного багатогранника необхідно враховувати відношення неперетину тільки з " ε -сусідами". Слід зазначити, що алгоритм не є ефективним

для окремих випадків, коли всі об'єкти є " ε -сусідами".

Параметр ε забезпечує баланс між числом нерівностей в кожній підзадачі нелінійного програмування (4.5) і числом підзадач (4.3), які необхідно згенерувати і розв'язати для того, щоб отримати локально-оптимальний розв'язок задачі (3.21) – (3.22).

Таким чином, метод POLYDEC дозволяє звести задачу (3.21) – (3.22) з великим числом нерівностей і великої розмірності $O(n^2)$ області допустимих розв'язків W до послідовності NLP підзадач (4.5) з меншим числом нелінійних нерівностей і меншої розмірності $O(n)$ підмножини розв'язків W_k .

4.4 Висновки по розділу

Розділ 4 присвячений методам розв'язання задач оптимальної упаковки багатогранників в опуклих контейнерах. Наведено опис стратегії розв'язання, метод пошуку допустимих стартових точок і метод декомпозиції POLYDEC для пошуку локальних екстремумів задачі OPP.

Достоїнство методу POLYDEC полягає в тому, що в більшості випадків аналізується значно менша кількість пар багатогранників, ніж m , оскільки для кожного багатогранника здійснюється перевірка тільки його « ε -околу».

Таким чином, метод декомпозиції POLYDEC дозволяє звести задачу (3.21) – (3.22) з великою кількістю нерівностей і великою розмірністю області допустимих розв'язків W вигляду (3.22) до послідовності підзадач вигляду (4.5) меншої розмірності зі значно меншим числом нелінійних нерівностей.

Результати досліджень, наведені в розділі 4, опубліковані в роботах [32, 40, 41, 44].

Список джерел, які використано у даному розділі наведено у повному списку використаних джерел [71, 209, 211].

5 РЕЗУЛЬТАТИ ОБЧИСЛЮВАЛЬНИХ ЕКСПЕРИМЕНТІВ

У розділі наводяться результати обчислювальних експериментів задачі *OPP* для різних опуклих контейнів та задачі *OPC* для різних охоплюючих оболонок, отриманих в ході тестування розроблених засобів математичного моделювання, побудованих математичних моделей, та запропонованих методів розв'язання. Проведено порівняння результатів розв'язання різноманітних задач *OPP* з кращими опублікованими результатами (benchmark instances). Наведено результати експериментів, що підтверджують ефективність методу POLYDEC.

Експерименти проводилися на комп'ютері AMD Athlon 64 X2 5200+. Для пошуку локальних мінімумів задач нелінійного програмування використовувався спеціальний програмний засіб для обчислення матриць Якобі и Гессе (розроблений у відділі математичного моделювання та оптимального проектування ІПМаш НАН України), та NLP-солвер IPOPT, який застосовує модифікацію методу внутрішньої точки [211].

Коди програм, точки локальних екстремумів для прикладів, наведених у цьому розділі, опис інтерфейсу системи розв'язання задач кластерингу наведено у додатках В і Г.

5.1 Вихідні дані для неопуклих багатогранників

Далі представлені вихідні дані десяти типів неопуклих багатогранників (рис. 5.1), які використовуються при розв'язання задач *OPP* та *OPC*, розглянутих в підрозділах 5.2 та 5.3.

Кожен тип неопуклого багатогранника представлений у вигляді об'єднання опуклих багатогранників, заданих відповідним набором

вершин у власній системі координат відповідного неопуклого багатогранника.

Далі наведені два списки вихідних даних (координати) вершин опуклих багатогранників, які формують неопуклі багатогранники: Список 1 – координати вершин; Список 2 – номери вершин (відповідно до списку 1), що визначають набір вершин опуклого багатогранника, що формує неопуклий багатогранник.

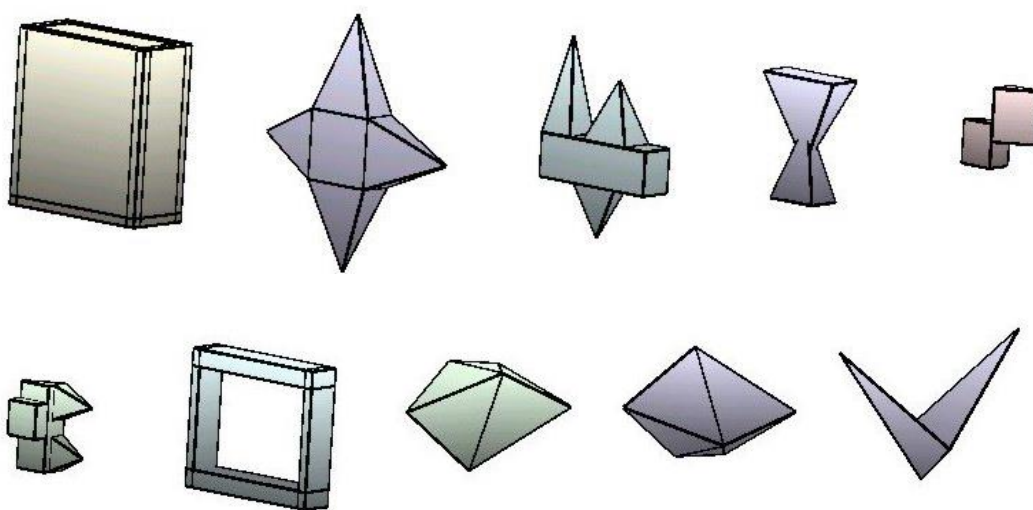


Рисунок 5.1 – Типи неопуклих багатогранників Q_j $j = 1, 2, \dots, 10$

Зауваження. Список 1 включає координати вершин неопуклого багатогранника i , в загальному випадку, координати додаткових вершин, які виникають при побудові декомпозиції неопуклого багатогранника на опуклі багатогранники

Список 1. Координати $(x_q, y_q, z_q), q = 1, 2, \dots, n$ вершин, що використовуються для опису неопуклих багатогранників:

$Q_1: \{ (x_q, y_q, z_q), q = 1, 2, \dots, 28 \} = \{ (0, 0, 0), (8, 0, 0), (8, 0, 20), (0, 0, 20), (0, 1, 0), (0, 1, 20), (8, 1, 20), (8, 1, 0), (8, 18, 0), (8, 18, 20), (7, 18, 0), (7, 18, 20), (7, 0, 20), (8, 17, 0), (0, 17, 0), (0, 17, 20), (8, 17, 20), (1, 18, 20), (0,$

18, 20), (1, 0, 20), (1, 0, 0), (1,18, 0), (0,18, 0), (0,18, 1), (0, 0, 1), (8, 0, 1), (8, 18, 1), (7, 0, 0)};

$Q_2: \{(x_q, y_q, z_q), q=1,2,\dots,12\} = \{(0, 0, 0), (4, 0, 0), (4, 8, 0), (0, 8, 0), (4, 0, 8), (4, 8, 8), (0, 8, 8), (0, 0, 8), (2, -8, 4), (2, 4, -10), (2, 18, 4), (2, 4, 19)\};$

$Q_3: \{(x_q, y_q, z_q), q=1,2,\dots,21\} = \{(0, 0, 0), (4, 0, 0), (4, 15, 0), (0, 15, 0), (4, 0, 5), (4, 15, 5), (0, 15, 5), (0, 0, 5), (3, 4, 0), (1, 4, 0), (1, 10, 0), (3, 10, 0), (2, 7, -6), (4, 4, 5), (0, 4, 5), (2, 2, 16), (4, 6, 5), (0, 6, 5), (0, 12, 5), (4, 12, 5), (2, 9, 12)\};$

$Q_4: \{(x_q, y_q, z_q), q=1,2,\dots,10\} = \{(2, -3, 0), (-2, -3, 0), (-2, 3, 0), (2, 3, 0), (0, 0, 9), (0, 0, 4), (2, -5, 14), (2, 5, 14), (-2, 5, 14), (-2, -5, 14)\};$

$Q_5: \{(x_q, y_q, z_q), q=1,2,\dots,12\} = \{(0, 0, 0), (3, 0, 0), (3, -4, 0), (3, -4, 5), (0, 0, 5), (3, 0, 5), (3, 0, 3), (0, 4, 3), (0, 4, 9), (3, 0, 9), (0, 0, 9), (0, 0, 3)\};$

$Q_6: \{(x_q, y_q, z_q), q=1,2,\dots,24\} = \{(0, 0, 0), (4, 0, 0), (4, 0, 16), (0, 0, 16), (0, 1, 0), (4, 1, 0), (4, 1, 16), (0, 1, 16), (0, 18, 16), (4, 18, 16), (4, 18, 0), (0, 18, 0), (0, 17, 16), (4, 17, 16), (4, 17, 0), (0, 17, 0), (4, 0, 2), (4, 18, 2), (0, 18, 2), (0, 0, 2), (4, 0, 14), (4, 18, 14), (0, 18, 14), (0, 0, 14)\};$

$Q_7: \{(x_q, y_q, z_q), q=1,2,\dots,22\} = \{(3, 0, 0), (3, 4, 0), (0, 4, 0), (3, 0, 10), (3, 4, 10), (0, 4, 10), (5, 0, 8), (5, 4, 8), (5, 4, 4), (5, 0, 4), (3, 0, 8), (3, 4, 8), (3, 4, 4), (3, 0, 4), (2, 4, 6), (2, 4, 10), (1, 9, 8), (0, 4, 6), (2, 4, 4), (0, 4, 4), (2, 4, 0), (1, 8, 2)\};$

$Q_8: \{(x_q, y_q, z_q), q=1,2,\dots,7\} = \{(0, 0, 0), (12, 0, 8), (-8, 8, 8), (-8, -8, 8), (0, -4, 12), (0, 4, 12), (-4, 0, 12)\};$

$Q_9: \{(x_q, y_q, z_q), q=1,2,\dots,7\} = \{(0, 0, 0), (12, 0, -8), (-8, 8, -8), (-8, -8, -8), (0, -4, -12), (0, 4, -12), (-4, 0, -12)\};$

$Q_{10}: \{(x_q, y_q, z_q), q=1,2,\dots,5\} = \{(0, 0, 0), (0, -4, 4), (0, 4, 4), (16, 0, 16), (-16, 0, 16)\}.$

Список 2. Номери вершин відповідних опуклих багатогранників K_i , що формують неопуклі багатогранники $Q_j, j=1,2,\dots,10$:

$$Q_1 = K_1 \cup K_2 \cup K_3 \cup K_4 \cup K_5$$

Q_1 : K_1 : {3, 10, 9, 2, 13, 12, 11, 28}, K_2 : {2, 9, 10, 19, 23, 14, 15, 16, 17}, K_3 : {18, 20, 21, 22, 1, 23, 19, 4}, K_4 : {1, 2, 3, 4, 5, 6, 7, 8}, K_5 : {1, 2, 9, 23, 24, 25, 26, 27};

$$Q_2 = K_1 \cup K_2 \cup K_3 \cup K_4 \cup K_5$$

Q_2 : K_1 : {1, 2, 5, 8, 9}, K_2 : {5, 6, 7, 8, 12}, K_3 : {3, 4, 6, 7, 11}, K_4 : {4, 3, 2, 1, 10}, K_5 : {1, 2, 3, 4, 5, 6, 7, 8};

$$Q_3 = K_1 \cup K_2 \cup K_3 \cup K_4$$

Q_3 : K_1 : {1, 2, 3, 4, 5, 6, 7, 8}, K_2 : {5, 8, 15, 14, 16}, K_3 : {17, 18, 19, 20, 21}, K_4 : {9, 10, 11, 12, 13};

$$Q_4 = K_1 \cup K_2$$

Q_4 : K_1 : {1, 2, 3, 4, 5}, K_2 : {6, 7, 8, 9, 10};

$$Q_5 = K_1 \cup K_2$$

Q_5 : K_1 : {1, 2, 3, 4, 5, 6}, K_2 : {7, 8, 9, 10, 11, 12};

$$Q_6 = K_1 \cup K_2 \cup K_3 \cup K_4$$

Q_6 : K_1 : {1, 2, 3, 4, 5, 6, 7, 8}, K_2 : {9, 10, 11, 12, 13, 14, 15, 16}, K_3 : {17, 18, 19, 20, 1, 2, 11, 12}, K_4 : {21, 22, 23, 24, 3, 4, 9, 10};

$$Q_7 = K_1 \cup K_2 \cup K_3 \cup K_4$$

Q_7 : K_1 : {1, 2, 3, 4, 5, 6}, K_2 : {7, 8, 9, 10, 11, 12, 13, 14}, K_3 : {15, 16, 17, 18, 6}, K_4 : {19, 20, 21, 22, 3};

$$Q_8 = K_1 \cup K_2 \cup K_3 \cup K_4$$

Q_8 : K_1 : {1, 5, 6, 7}, K_2 : {1, 4, 5, 7}, K_3 : {1, 2, 5, 6}, K_4 : {1, 3, 6, 7};

$$Q_9 = K_1 \cup K_2 \cup K_3 \cup K_4$$

Q_9 : K_1 : {1, 5, 6, 7}, K_2 : {1, 4, 5, 7}, K_3 : {1, 2, 5, 6}, K_4 : {1, 3, 6, 7};

$$Q_{10} = K_1 \cup K_2$$

$$Q_{10}: K_1: \{1, 2, 3, 4\}, K_2: \{1, 2, 3, 5\}.$$

5.2 Результати обчислювальних експериментів для *ОРС*

Припускаємо, що $A = \bigcup_{i=1}^{n_A} A_i$ і $B = \bigcup_{j=1}^{n_B} B_j$ – довільні неопуклі

багатогранники, де A_i, B_j є опуклими багатогранниками.

Як контейнери розглядаються наступні області:

– кубоїд (задача 1)

$$\mathbf{B} = \{(x, y, z) \in R^3 \mid \min\{x+l, -x+l, y+w, -y+w, z+h, -z+h\} \geq 0\},$$

де l, w і h є змінними метричними характеристиками кубоїда (довжина, ширина і висота);

– шар зі змінним радіусом r (задача 2)

$$\mathbf{S} = \{(x, y, z) \in R^3 \mid r^2 - x^2 - y^2 - z^2 \geq 0\}$$

– циліндр з радіусом λr і висотою λh , λ – змінний коефіцієнт гомотетії (задача 3)

$$\mathbf{C} = \{(x, y, z) \in R^3 \mid \min\{(\lambda r)^2 - x^2 - y^2, -z + \lambda h, z + \lambda h\} \geq 0\}$$

вихідні розміри контейнера відповідають $\lambda = 1$.

Приклад 5.1. Розглядається задача 1. Контейнер – кубоїд **B**, багатогранники:

$A: \{ p_s(x_A, y_A, z_A), s=1, 2, \dots, 10 \} = \{(2, -3, 0), (-2, -3, 0), (-2, 3, 0), (2, 3, 0), (0, 0, 9), (0, 0, 4), (2, -5, 14), (2, 5, 14), (-2, 5, 14), (-2, -5, 14)\};$

$B: \{ p_s(x_B, y_B, z_B), s=1, 2, \dots, 10 \} = \{(2, -3, 0), (-2, -3, 0), (-2, 3, 0), (2, 3, 0), (0, 0, 9), (0, 0, 4), (2, -5, 14), (2, 5, 14), (-2, 5, 14), (-2, -5, 14)\};$

Кластерінг багатогранників A і B всередині мінімально охоплюючого кубоїда **B** наведена на рис.

5.2. $F(u^*) = l^* \cdot w^* \cdot h^* = 1502.0771$, де $l^* = 16.4725$, $w^* = 4.000$, $h^* = 22.7966$. Час розв'язання 3.21 с.

Приклад 5.2. Розглядається задача 1. Контейнер – кубоїд **B**, багатогранники:

$A: \{ p_s(x_A, y_A, z_A), s=1, 2, \dots, 10 \} = \{(2, -3, 0), (-2, -3, 0), (-2, 3, 0), (2, 3, 0), (0, 0, 9), (0, 0, 4), (2, -5, 14), (2, 5, 14), (-2, 5, 14), (-2, -5, 14)\};$

$B: \{ p_s(x_B, y_B, z_B), s=1, 2, \dots, 24 \} = \{(0, 0, 0), (4, 0, 0), (4, 0, 16), (0, 0, 16), (0, 1, 0), (4, 1, 0), (4, 1, 16), (0, 1, 16), (0, 18, 16), (4, 18, 16), (4, 18, 0), (0, 18, 0), (0, 17, 16), (4, 17, 16), (4, 17, 0), (0, 17, 0), (4, 0, 2), (4, 18, 2), (0, 18, 2), (0, 0, 2), (4, 0, 14), (4, 18, 14), (0, 18, 14), (0, 0, 14)\};$

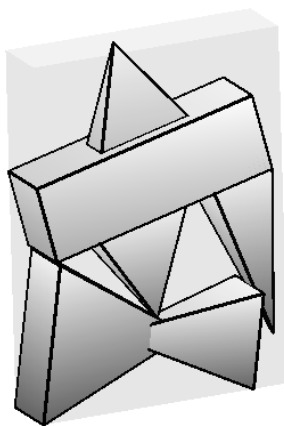


Рисунок 5.2 – Локально оптимальне розміщення багатогранників A і B для прикладу 5.1

Кластерінг багатогранників A і B всередині мінімально охоплюючого кубоїда B наведена на рис. 5.3

$F(u^*) = l^* \cdot w^* \cdot h^* = 1399.9999$, де $l^* = 10.000$, $w^* = 10.000$, $h^* = 14.000$. Час розв'язання 8.674 с.

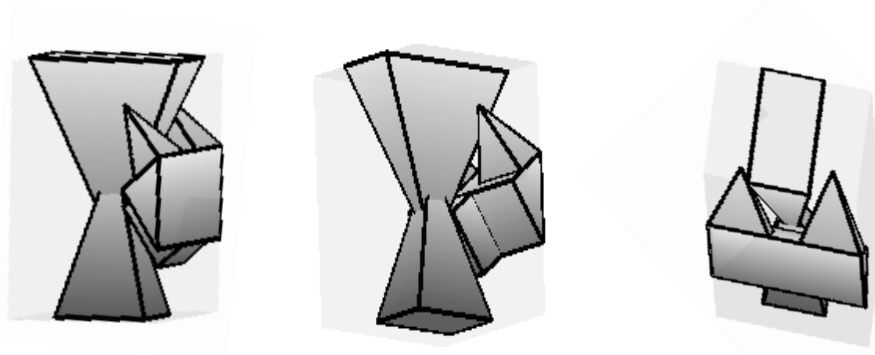


Рисунок 5.3 – Локально оптимальне розміщення багатогранників A і B для прикладу 5.2

Приклад 5.3. Розглядається задача 1. Контейнер – кубоїд B , багатогранники:

$A: \{ p_s(x_A, y_A, z_A), s=1, 2, \dots, 10 \} = \{(2, -3, 0), (-2, -3, 0), (-2, 3, 0), (2, 3, 0), (0, 0, 9), (0, 0, 4), (2, -5, 14), (2, 5, 14), (-2, 5, 14), (-2, -5, 14)\};$

$B: \{ p_s(x_B, y_B, z_B), s=1, 2, \dots, 22 \} = \{(3, 0, 0), (3, 4, 0), (0, 4, 0), (3, 0, 10), (3, 4, 10), (0, 4, 10), (5, 0, 8), (5, 4, 8), (5, 4, 4), (5, 0, 4), (3, 0, 8), (3, 4, 8), (3, 4, 4), (3, 0, 4), (2, 4, 6), (2, 4, 10), (1, 9, 8), (0, 4, 6), (2, 4, 4), (0, 4, 4), (2, 4, 0), (1, 8, 2)\};$

Кластерінг багатогранників A і B всередині мінімально охоплюючого кубоїда B наведена на рис. 5.4

$F(u^*) = l^* \cdot w^* \cdot h^* = 1152.000$, де $l^* = 4.000$, $w^* = 16.000$, $h^* = 18.000$. Час розв'язання 4.93 с.

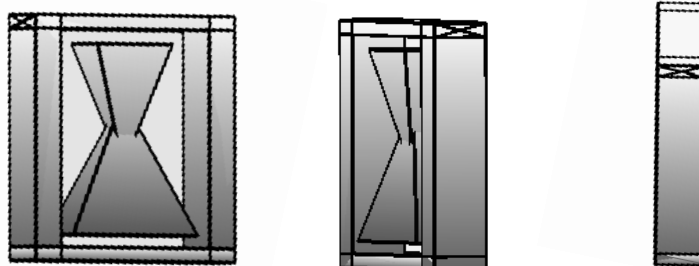


Рисунок 5.4 – Локально оптимальне розміщення багатогранників A і B для прикладу 5.3

Приклад 5.4. Розглядається задача 2. Контейнер – шар S , багатогранники:

$A: \{ p_s(x_A, y_A, z_A), s=1, 2, \dots, 7 \} = \{(0, 0, 0), (12, 0, 8), (-8, 8, 8), (-8, -8, 8), (0, -4, 12), (0, 4, 12), (-4, 0, 12)\};$

$B: \{ p_s(x_B, y_B, z_B), s=1, 2, \dots, 7 \} = \{(0, 0, 0), (12, 0, -8), (-8, 8, -8), (-8, -8, -8), (0, -4, -12), (0, 4, -12), (-4, 0, -12)\};$

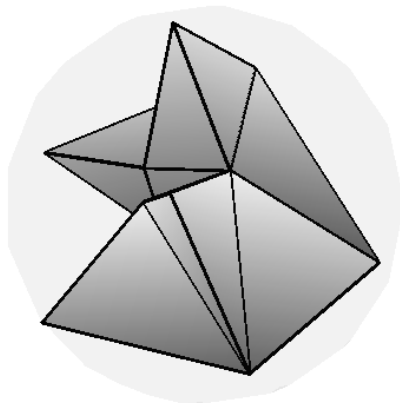
Кластерінг багатогранників A і B всередині мінімально охоплюючого шару S наведена на рис. 5.5 а). $F(u^*) = r^* = 12.2274$. Час розв'язання 2.351 с.

Приклад 5.5. Розглядається задача 2. Контейнер – шар S , багатогранники:

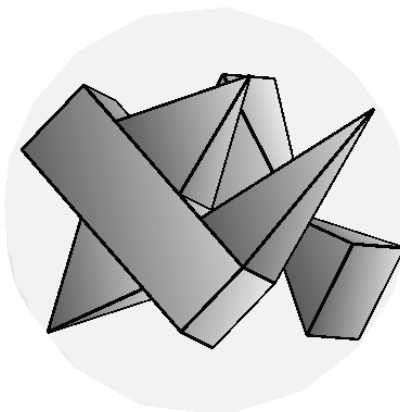
$A: \{ p_s(x_A, y_A, z_A), s=1, 2, \dots, 10 \} = \{(2, -3, 0), (-2, -3, 0), (-2, 3, 0), (2, 3, 0), (0, 0, 9), (0, 0, 4), (2, -5, 14), (2, 5, 14), (-2, 5, 14), (-2, -5, 14)\};$

$B: \{ p_s(x_B, y_B, z_B), s=1, 2, \dots, 24 \} = \{(0, 0, 0), (4, 0, 0), (4, 0, 16), (0, 0, 16), (0, 1, 0), (4, 1, 0), (4, 1, 16), (0, 1, 16), (0, 18, 16), (4, 18, 16), (4, 18, 0), (0, 18, 0), (0, 17, 16), (4, 17, 16), (4, 17, 0), (0, 17, 0), (4, 0, 2), (4, 18, 2), (0, 18, 2), (0, 0, 2), (4, 0, 14), (4, 18, 14), (0, 18, 14), (0, 0, 14)\};$

Кластерінг багатогранників A і B всередині мінімально охоплюючого шару S наведена на рис. 5.5 б). $F(u^*) = r^* = 11.2996$. Час розв'язання 2.352 с.



а)



б)

Рисунок 5.5 – Локально оптимальне розміщення багатогранників A і B :

а) для прикладу 5.4; б) для прикладу 5.5

Приклад 5.6. Розглядається задача 3. Контейнер – циліндр C , багатогранники:

$A: \{ p_s(x_A, y_A, z_A), s=1, 2, \dots, 21 \} = \{(0, 0, 0), (4, 0, 0), (4, 15, 0), (0, 15, 0), (4, 0, 5), (4, 15, 5), (0, 15, 5), (0, 0, 5), (3, 4, 0), (1, 4, 0), (1, 10, 0), (3, 10, 0), (2, 7, -6), (4, 4, 5), (0, 4, 5), (2, 2, 16), (4, 6, 5), (0, 6, 5), (0, 12, 5), (4, 12, 5), (2, 9, 12)\}$;

$B: \{ p_s(x_B, y_B, z_B), s=1, 2, \dots, 10 \} = \{(2, -3, 0), (-2, -3, 0), (-2, 3, 0), (2, 3, 0), (0, 0, 9), (0, 0, 4), (2, -5, 14), (2, 5, 14), (-2, 5, 14), (-2, -5, 14)\};$

Кластерінг багатогранників A і B всередині мінімально охоплюючого циліндра C наведена на рис. 5.6. $F(u^*) = \lambda^* = 0.682$, що відповідає $r = 20$, $h = 40$. Час розв'язання 2.65 с.

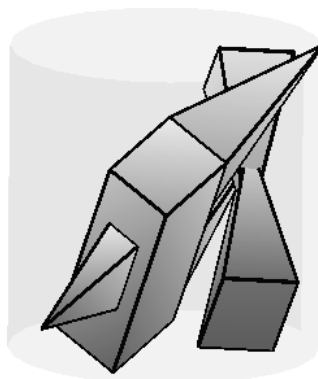


Рисунок 5.6 – Локально оптимальне розміщення багатогранників A і B для прикладу 5.6

5.3 Результати обчислювальних експериментів для ОРР

Приклад 5.7. Розв'язується задача мінімізації висоти кубоїда, завдяки щільному розміщенню довільних багатогранників. Розглядається набір з $n=30$ неопуклих багатогранників. На рис. 5.8 зображено розміщення набору неопуклих багатогранників: а) допустимий розв'язок. Контейнер має такі розміри і об'єм: $(l, w, h^*) = (30, 49, 35)$ і $F(u^*) = 35$ (рис. 5.7 а)). Використовувалось 100 стартових точок. Час обчислення 252 с.; б) локально оптимальний розв'язок $(l, w, h^*) = (40, 40, 31.564563)$ і $F(u^*) = 31.564563$ (рис. 5.7 б)). Використовувалась одна стартова точка. Час обчислення 3.5 г.

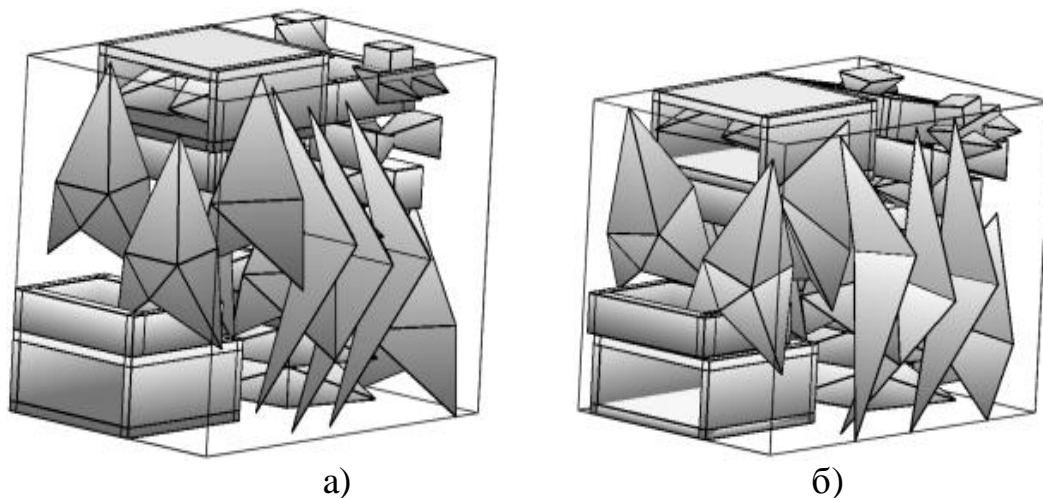
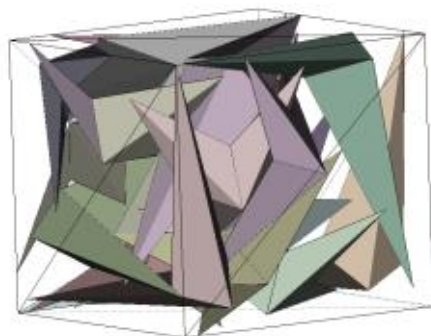


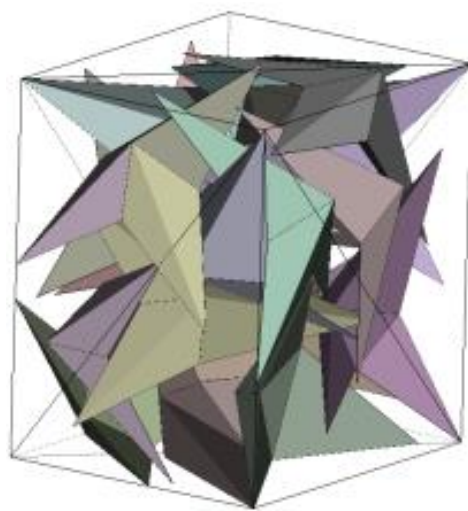
Рисунок 5.7. Розміщення опуклих багатогранників для прикладу 5.7: а) допустимий розв'язок; б) локально оптимальний розв'язок.

Далі наводяться приклади з використанням даних щодо опуклих багатогранників з [62] і неопуклих багатогранників з [71].

Приклад 5.8. Генерується набір з $n = 20$ неопуклих багатогранників. На рис. 5.9 наведено локально оптимальне розміщення багатогранників: а) неопуклі багатогранники, що складаються з 2 видів багатогранників, запропонованих в [71], по 10 багатогранників кожного виду. Контейнер має такі розміри і об'єм: $(l^*, w^*, h^*) = (26.352202, 23.751414, 24.405546)$ і $F(u^*) = 15275.481504955$ (рис. 5.8 а); Використовувалося 2 стартові точки. Час обчислення 8729.45 с. б) однакові неопуклі багатогранники, які запропоновані в [71]. Контейнер має такі розміри і об'єм: $(l^*, w^*, h^*) = (23.770676, 26.621255, 20.236356)$ і $F(u^*) = 12805.67185$ (рис. 5.8 б); Використовувалося 10 стартових точок. Час обчислення 59497.9 с.



а)

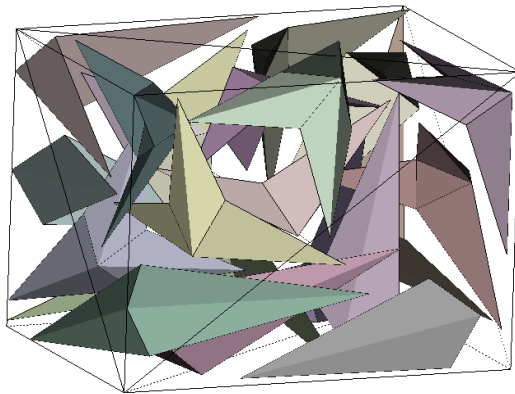


б)

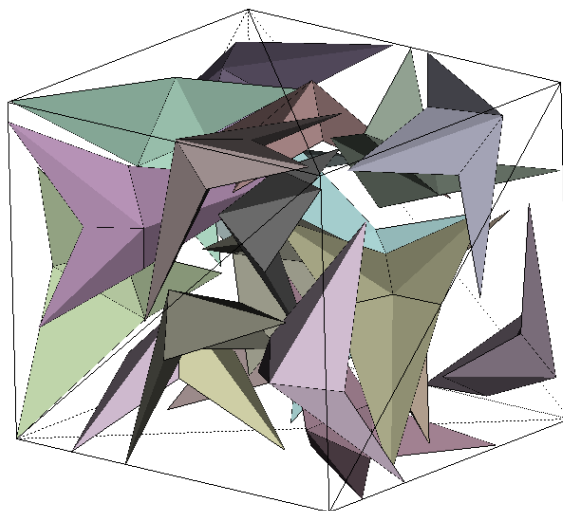
Рисунок 5.8 – Локально оптимальне розміщення неопуклих багатогранників для прикладу 5.8

Приклад 5.9. Розглядається набір з $n = 20$ неопуклих багатогранників. На рис. 5.9 наведено локально оптимальне розміщення багатогранників: а) неопуклі багатогранники, що складаються з 2 видів багатогранників, запропонованих в [71], по 10 багатогранників кожного виду. Контейнер має такі розміри і об'єм: $(l^*, w^*, h^*) = (26.589096, 26.523957, 36.170619)$ і $F(u^*) = 25509.257689$ з $\rho = 1.5$ (рис. 5.9 а). Використовувалося 10 стартових точок. Час обчислення 24696.46 с. б) однакові неопуклі багатогранники, які запропоновані в [71]. Контейнер має такі розміри і об'єм: $(l^*, w^*, h^*) = (27.979507, 26.540835, 30.672555)$ і

$F(u^*) = 22777.423352426$ з $\rho = 1.5$ (рис. 5.9 б). Використовувалося 10 стартових точок. Час обчислення 28700.16 с.



а)



б)

Рисунок 5.9 – Локально оптимальне розміщення неопуклих багатогранників для прикладу 5.9

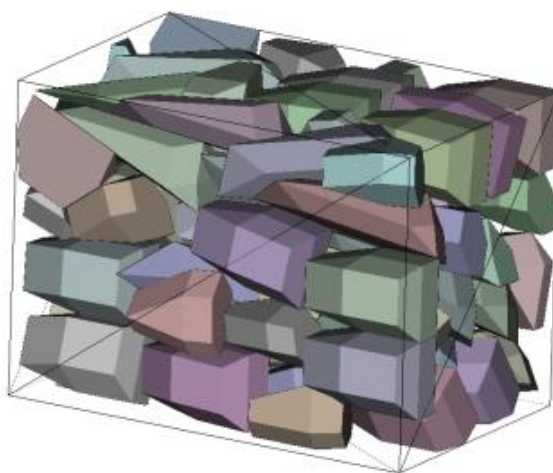
Далі порівнюються результати, отримані в цій роботі, з аналогами, запропонованими в [71]. Шукається локально оптимальний розв'язок, використовуючи метод POLYDEC: а) – старт з допустимої точки, згенерованої методом FSPA; б) – старт з допустимої точки, згенерованої алгоритмом, наведеним в [71].

Приклад 5.10. Розглядається набір з $n = 80$ опуклих багатогранників наведених в роботі [71]. На рис. 5.10 зображено локально оптимальне розміщення багатогранників. Контейнер має такі розміри і об'єм: а) $(l^*, w^*, h^*) = (43.433885, 41.843593, 45.005998)$ і $F(u^*) = 81795.216927$, стартуючи з допустимої точки, знайденої за допомогою FSPA; Час обчислення 46035.78 с.; б) $(l^*, w^*, h^*) = (36.356900, 40.876483, 56.255757)$ і $F(u^*) = 83604.054487$, стартуючи з допустимої точки, знайденої за допомогою алгоритму, запропонованого в [71]. Час обчислення 42950.4 с. Поліпшення значення функції цілі порівняно з результатом, наведеним в [71], становить: а) 27.88%; б) 26.29%.

Приклад 5.11. Розглядається набір з $n = 20$ неопуклих багатогранників наведених в роботі [71]. На рис. 5.11 наведено локально оптимальне розміщення багатогранників. Контейнер має такі розміри і об'єм: а) $(l^*, w^*, h^*) = (29.715973, 30.607091, 30.161686)$ і $F(u^*) = 27432.64$, стартуючи з допустимої точки, знайденої за допомогою FSPA; б) $(l^*, w^*, h^*) = (31.482020, 27.899495, 32.000000)$ і $F(u^*) = 28106.640706557$, стартуючи з допустимої точки, знайденої за допомогою алгоритму, запропонованого в [71]. Було використано 11 стартових точок і встановлено обмеження часу – 10 годин. Поліпшення значення функції цілі порівняно з результатом, наведеним в [71], становить: а) 18.36%; б) 16.35%.



а)

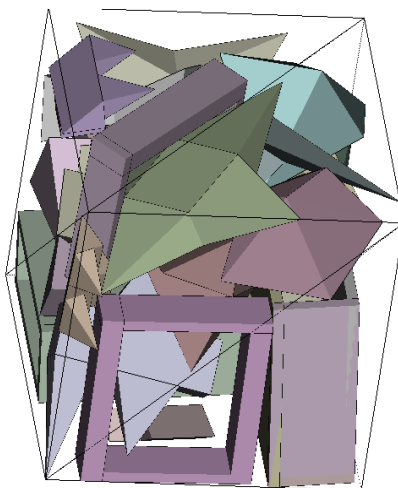


б)

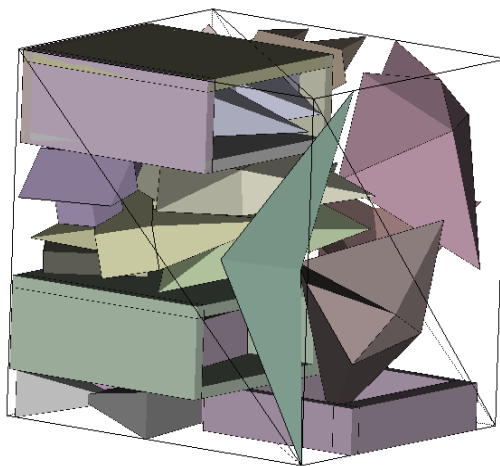
Рисунок 5.10 – Локально оптимальне розміщення опуклих багатогранників для прикладу 5.10, стартуючи з допустимої точки, знайденої за допомогою: а) – FSPA; б) – алгоритму в [71]

Приклад 5.12. Розглядається набір з $n = 30$ неопуклих багатогранників наведених в роботі [71]. На рис. 5.12 показано локально оптимальне розміщення багатогранників. Контейнер має такі розміри і об'єм: а) $(l^*, w^*, h^*) = (36.992957, 36.379659, 30.945493)$ і $F(u^*) = 41646.170954096$, стартуючи з допустимої точки, знайденої за допомогою FSPA; б) $(l^*, w^*, h^*) = (31.437607, 26.192043, 48.914802)$ і

$F(u^*) = 40277.1936470275$, стартуючи з допустимої точки, знайденої за допомогою алгоритму, запропонованого в [71]. Було використано 11 стартових точок і встановлено обмеження часу – 10 годин. Поліпшення значення функції цілі порівняно з результатом, наведеним в [71], становить: а) 19.06%; б) 21.72%

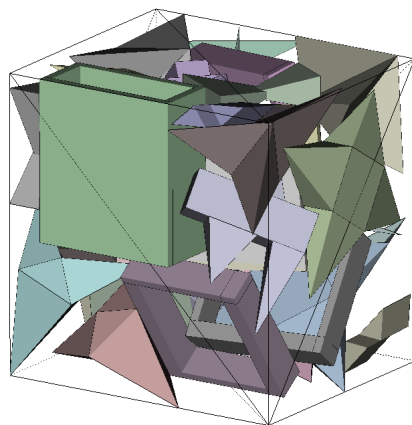


а)

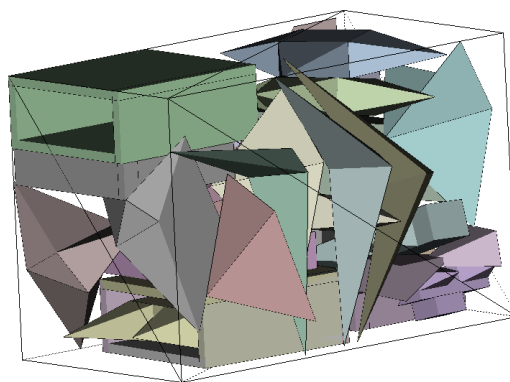


б)

Рисунок 5.11 – Локально оптимальне розміщення неопуклих багатогранників для прикладу 5.11, стартуючи з допустимої точки, знайденої за допомогою: а) – FSPA; б) – алгоритму в [71]



а)

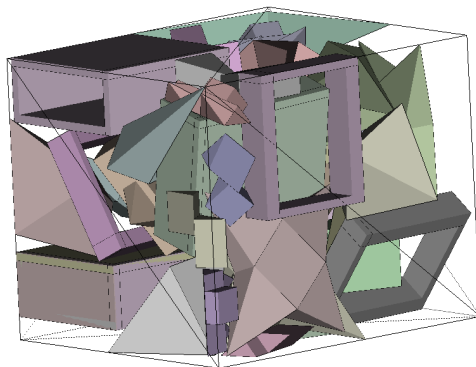


б)

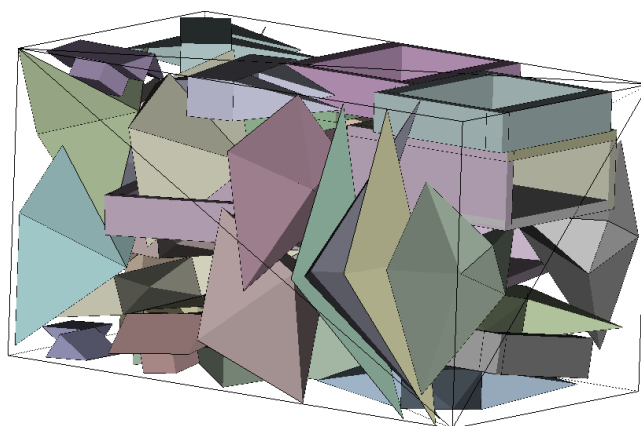
Рисунок 5.12 – Локально оптимальне розміщення неопуклих багатогранників для прикладу 5.12, стартуючи з допустимої точки, знайденої за допомогою: а) – FSPA; б) – алгоритму в [71]

Приклад 5.13. Розглядається набір з $n = 40$ неопуклих багатогранників наведених в роботі [71]. На рис. 5.13 зображено локально оптимальне розміщення для набору неопуклих багатогранників. Контейнер має такі розміри і об'єм: а) $(l^*, w^*, h^*) = (34.997472, 36.965583, 43.277713)$ і $F(u^*) = 55988.46483094$, стартуючи з допустимої точки, знайденої за допомогою FSPA; б) $(l^*, w^*, h^*) = (31.141934, 30.808630, 55.406139)$ і $F(u^*) = 53158.883489539$, стартуючи з допустимої точки, знайденої за допомогою алгоритму,

запропонованого в [71]. Було згенеровано 3 стартові точки і встановлено обмеження часу – 10 годин. Поліпшення значення функції цілі порівняно з результатом, наведеним в [71], становить: а) 15.64%; б) 19.91%



а)

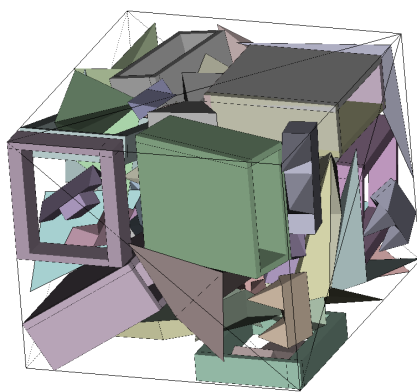


б)

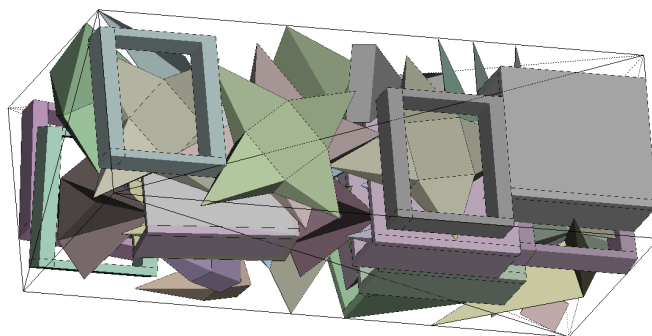
Рисунок 5.13 – Локально оптимальне розміщення неопуклих багатогранників для прикладу 5.13, стартуючи з допустимої точки, знайденої за допомогою: а) – FSPA; б) – алгоритму в [71]

Приклад 5.14. Розглядається набір з $n = 50$ неопуклих багатогранників наведених в роботі [71]. На рис. 5.14 наведено локально оптимальне розміщення для набору неопуклих багатогранників. Контейнер має такі розміри і об'єм: а) $(l^*, w^*, h^*) =$

(46.974275, 34.830562, 41.692364) і $F(u^*) = 68214.563945567$, стартуючи з допустимої точки, знайденої за допомогою FSPA; б) $(l^*, w^*, h^*) = (32.000000, 25.589466, 75.263728)$ і $F(u^*) = 61630.683947233$, стартуючи з допустимої точки, знайденої за допомогою алгоритму, запропонованого в [71]. Згенерована одна стартова точка і встановлено обмеження часу – 10 годин. Поліпшення значення функції цілі порівняно з результатом, наведеним в [71], становить: а) 17.45%; б) 25.42%.



а)



б)

Рисунок 5.14 – Локально оптимальне розміщення неопуклих багатогранників для прикладу 5.14, стартуючи з допустимої точки, знайденої за допомогою: а) – FSPA; б) – алгоритму в [71]

Приклад 5.15. Розглядається набір багатогранників, наведених в роботі [28]. На рис. 5.15 наведено локально оптимальне розміщення для

$n = 36$ неопуклих багатогранників. Контейнер має такі розміри і об'єм: $(l^*, w^*, h^*) = (21.585128, 19.868586, 24.393820)$ і $F(u^*) = 10461.67$. Було використано 5 стартових точок і встановлено обмеження часу – 10 годин. Поліпшення значення функції цілі порівняно з результатом, наведеним в [28], становить 16.18%.

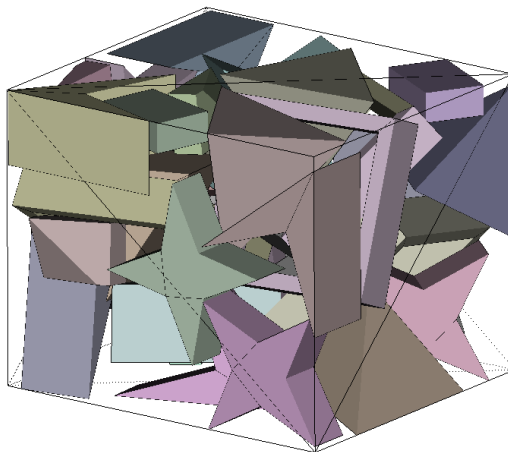


Рисунок 5.15 – Локально оптимальне розміщення неопуклих багатогранників для прикладу 5.15

Приклад 5.16. Генеруємо набір з $n = 98$ опуклих багатогранників, з [92]. На рис. 5.16 зображено локально оптимальне розміщення набору опуклих багатогранників з урахуванням обмежень на мінімально допустимі відстані. Контейнер має такі розміри і об'єм: $(l^*, w^*, h^*) = (41.3510, 33.0721, 31.7988)$ і $F(u^*) = 43487.0040$ з $\rho = 1.5$. Використовувалася одна стартова точка. Час обчислення 48152.79 с.

Приклад 5.17. Пакуємо 45 неопуклих багатогранників 10 типів, наведених в підрозділі 5.1, по 5 багатогранників кожного типу з верхнього списку багатогранників і по 4 – з нижнього (Рис. 5.1). На рис. 5.17 зображено локально оптимальне розміщення набору неопуклих багатогранників. Контейнер має такі розміри: $(l^*, w^*, h^*) = (39.7324,$

34.8629, 44.6587) і $F(u^*)=61860.807$. Використовували 3 стартових точки. Час обчислення - 159884.0 с.

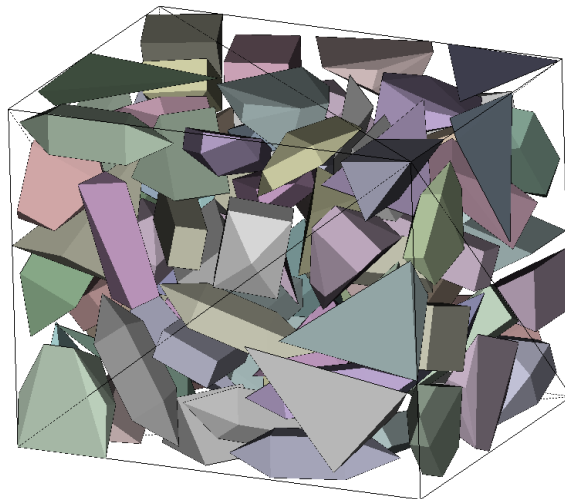


Рисунок 5.16 – Локально оптимальне розміщення опуклих багатогранників в прикладі 5.16

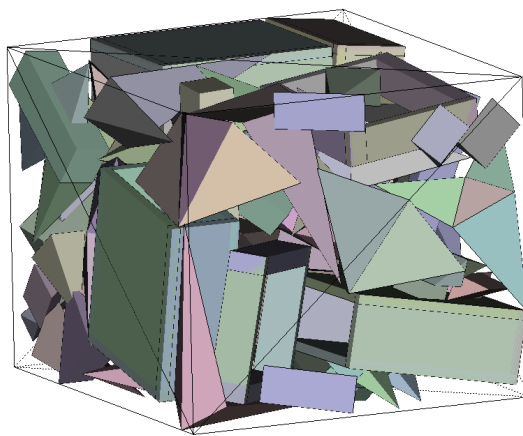


Рисунок 5.17 – Локально оптимальне розміщення неопуклих багатогранників в прикладі 5.17

У табл. 5.1 наведені результати порівняння для прикладів, поданих в [28]. Для кожного прикладу мінімальний об'єм контейнера, знайдений методом, що запропонований в цьому дослідженні, виявляється менше, ніж найкращий розв'язок, запропонований в [28].

Для кожного з прикладів розглядаються багатогранники, вихідна інформація щодо яких наведена в роботі [71]. Припускаємо, що $\varepsilon = 5$ для оптимізаційної процедури, що реалізує метод POLYDEC.

Таблиця 5.1 – Порівняння результатів для прикладів, наведених в [28]

Задача	Найкращий об'єм з [28]	Найкращий час з [28]	Об'єм, знайдений за допомогою FSPA * + POLYDEC	Час пошуку розв'язання за допомогою FSPA * + POLYDEC	Об'єм, знайдений за допомогою [71] ** + POLYDEC	Час пошуку розв'язання за допомогою [71] ** + POLYDEC
20 з [71]	32550	26202.1	27432.64	34313.34	28106.64	5360.67
30 з [71]	48300	53741.5	41646.17	35289.34	40277.19	33008.89
40 з [71]	61950	99952.0	53158.88	201501.5	55988.46	195051.51
50 з [71]	77280	125210.6	68214.56	215148.14	61630.68	270654.84
36 з [71]	12480	9637.5	10461.67	23023.12	-	-

* – допустима стартова точка, знайдена за допомогою FSPA;

** – допустима стартова точка, знайдена за допомогою алгоритму, запропонованого в [71]

Приклад 5.18. Розглядаються $n = 80$ опуклих багатогранників, вихідні дані щодо яких наведені в [71]. На рис. 5.18 а) наведено локально оптимальне розміщення заданого набору багатогранників в кулі радіуса $r^* = 27.327080$. Час розв'язання 42914.375 с.

Приклад 5.19. Розглядаються $n = 30$ неопуклих багатогранників, вихідні дані щодо яких наведені в [71]. На рис. 5.18 б) наведено локально оптимальне розміщення заданого набору багатогранників в кулі радіуса $r^* = 23.070962$. Час розв'язання 29491.35 с.

Приклад 5.20. Розглядаються $n = 25$ опуклих багатогранників,

вихідні дані щодо яких наведені в [71]. На рис. 5.19 наведено локально оптимальне розміщення заданого набору багатогранників в циліндрі за $r^* = h^* = 12.346384$. Час розв'язання 723.15 с.



а)



б)

Рисунок 5.18 – Локально оптимальне розміщення опуклих багатогранників: а) для прикладу 5.18; б) для прикладу 5.19



Рисунок 5.19 – Локально оптимальне розміщення опуклих багатогранників для прикладу 5.20

Приклад 5.21. Розглядаються $n = 40$ опуклих багатогранників. Контейнером є перетин куль з центрами в точках $(-10, 0, 0)$ і $(10, 0, 0)$ і з однаковим змінним радіусом r . На рис. 5.20 а) наведено локально оптимальне розміщення заданого набору багатогранників в довільній опуклій області (перетині куль). Значення функції цілі $r^* = 27.624$. Час розв'язання 3730.34 с.

Приклад 5.22. Розглядаються $n = 25$ неопуклих багатогранників. Контейнером є перетин куль з центрами в точках $(-10,0,0)$ і $(10,0,0)$ і з однаковим змінним радіусом r . На рис. 5.20 б) наведено локально оптимальне розміщення заданого набору багатогранників в довільній опуклій області (перетині куль). Значення функції цілі $r^* = 26.376$. Час розв'язання 8658.87 с.

Приклад 5.23. Розглядаються $n = 40$ неопуклих багатогранників з [71]. На рис. 5.21 зображено локально оптимальне розміщення набору неопуклих багатогранників в контейнері типу еліпсоїд, який має такі розміри: $a^* = 28.3608$, $b^* = 24.8157$, $c^* = 21.270$. Використовувалося 10 стартових точок. Час обчислення становив 29491.35 с.



а)



б)

Рисунок 5.20 – Локально оптимальне розміщення неопуклих багатогранників: а) для прикладу 5.21; б) для прикладу 5.22

Приклад 5.24. Розглядаються $n=98$ опуклих багатогранників з [71] із заданими вагами. На рис. 5.22 зображено локально оптимальне розміщення набору опуклих багатогранників в кулі радіусу $r^* = 18.0409$ з урахуванням обмежень балансу. Використовувалася 1 стартова точка. Час обчислення становив 26150.23 с.



Рисунок 5.21 – Локально оптимальне розміщення неопуклих багатогранників для прикладу 5.23



Рисунок 5.22 – Локально оптимальне розміщення опуклих багатогранників для прикладу 5.24

Приклад 5.25. Розглядаються $n = 25$ опуклих багатогранників, вихідні дані щодо яких наведені в [71]. На рис. 5.23 наведено локально оптимальне розміщення заданого набору багатогранників в опуклому багатограннику при $\lambda^* = 0.82309$. Час розв'язання 924.12 с.



Рисунок 5.23 – Локально оптимальне розміщення опуклих багатогранників для прикладу 5.25

Приклад 5.26. Розглядаються $n = 30$ неопуклих багатогранників, вихідні дані щодо яких наведені в [71]. На рис. 5.24 наведено локально оптимальне розміщення заданого набору багатогранників в опуклому багатограннику при $\lambda^* = 0.769032$ за умови що початковий радіус вписаної в багатогранник кулі $r^0 = 30$. Час розв'язання 27541.345 с.



Рисунок 5.24 – Локально оптимальне розміщення неопуклих багатогранників для прикладу 5.26

Приклад 5.27. Розглядається набір з $n=98$ опуклих багатогранників з [92] із заданими вагами. На рис. 5.25 зображено локально оптимальне розміщення набору опуклих багатогранників з урахуванням обмежень балансу. Контейнер має такі розміри і об'єм $(l^*, w^*, h^*) = (30.9324, 28.1897, 26.5064)$ і $F(u^*) = 23113.06$. Використовувалася одна стартова точка. Час обчислення 147967.3 с.

Приклад 5.28. Розглядаються $n=80$ опуклих багатогранників з [71] із заданими вагами. На рис. 5.26 наведено локально оптимальне розміщення заданого набору багатогранників в циліндрі при $r^* = 0.25$, $h^* = 17.874402$ з урахуванням обмежень балансу. Час розв'язання 43613.42 с.

Розглянемо обчислювальні експерименти, що підтверджують ефективність методу POLYDEC.

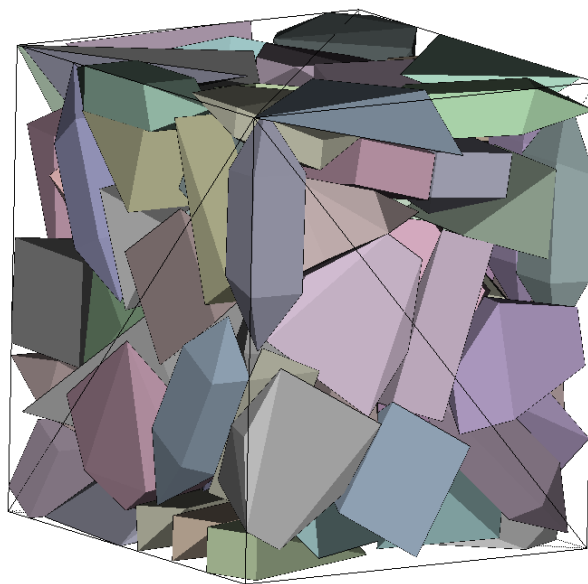


Рисунок 5.25 – Локально оптимальне розміщення опуклих багатогранників в прикладі 5.27



Рисунок 5.26 – Локально оптимальне розміщення опуклих багатогранників для прикладу 5.28

Приклад 5.29. Розглядаються $n = 47$ опуклих багатогранників з [71] із заданими вагами. На рис. 5.27 наведено локально оптимальне розміщення заданого набору багатогранників в циліндрі при $r^* = 12.280406$ $h^* = 24.560811$ з урахуванням обмежень балансу. Час розв'язання 1387.2 с.

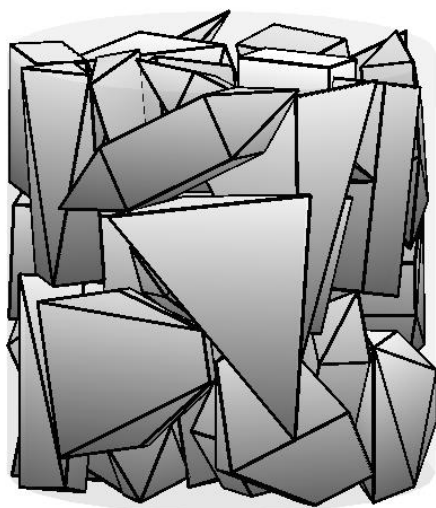


Рисунок 5.27 – Локально оптимальне розміщення опуклих багатогранників для прикладу 5.29

Приклад 5.30. Розглядаються $n = 47$ опуклих багатогранників з [71] із заданими вагами. На рис. 5.28 наведено локально оптимальне розміщення заданого набору багатогранників в циліндрі при $r^* = h^* = 15.09492$ з урахуванням обмежень балансу. Час розв'язання 2312.855 с.

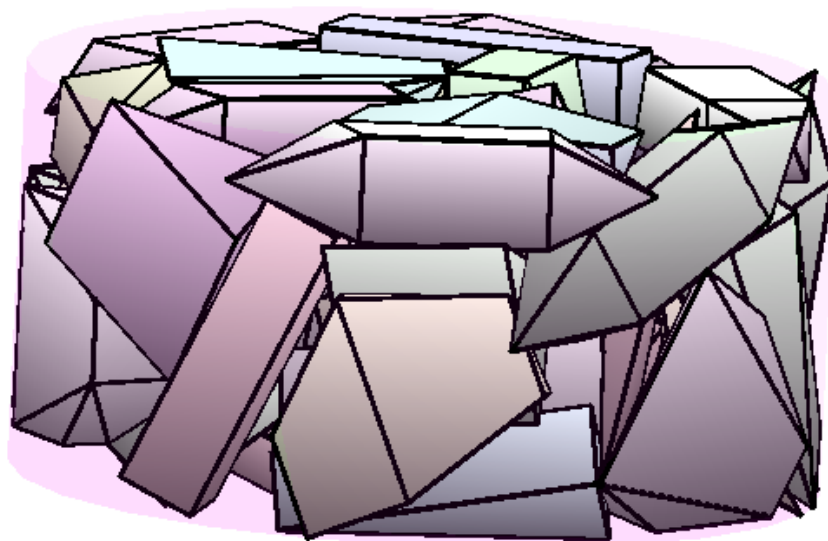


Рисунок 5.28 – Локально оптимальне розміщення опуклих багатогранників для прикладу 5.30

Наведемо порівняння результатів розв'язання різних задач упаковки довільних багатогранників з застосуванням та без застосування метода POLYDEC.

Результати цього порівняння наведено у табл. 5.2.

Далі наведено діаграми порівняння середнього часу обчислення, числа змінних та числа нерівностей для різних задач упаковки довільних багатогранників з застосуванням та без застосування метода POLYDEC.

Таблиця 5.2. Результати аналізу ефективності методу декомпозиції POLYDEC

Задача	Середній час обчислень (с) (рис. 5.29)		Число змінних (рис. 5.30)		Число нерівностей (рис. 5.31)	
	Із застосуванням POLYDEC	Без застосування POLYDEC	Із застосуванням POLYDEC	Без застосування POLYDEC	Із застосуванням POLYDEC	Без застосування POLYDEC
$N = 10$	283	1380	626	1791	3086	7934
$N = 20$	4980.74	75026.31	1334	7471	8028	30916
$N = 30$	35289.34	-	2817	17334	16047	69336

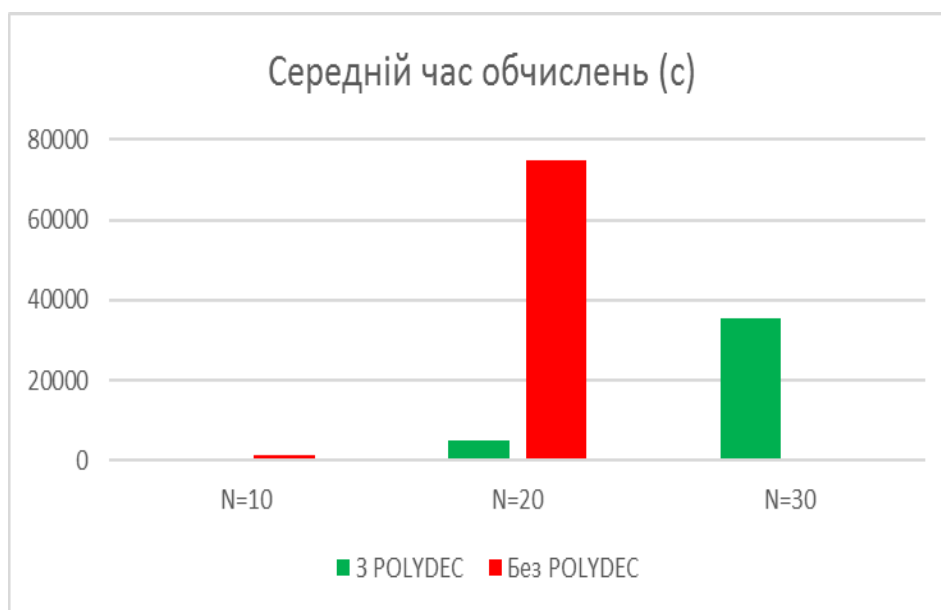


Рисунок 5.29 – Порівняння середнього часу обчислення для різних задач упаковки довільних багатогранників з застосуванням та без застосування методу POLYDEC

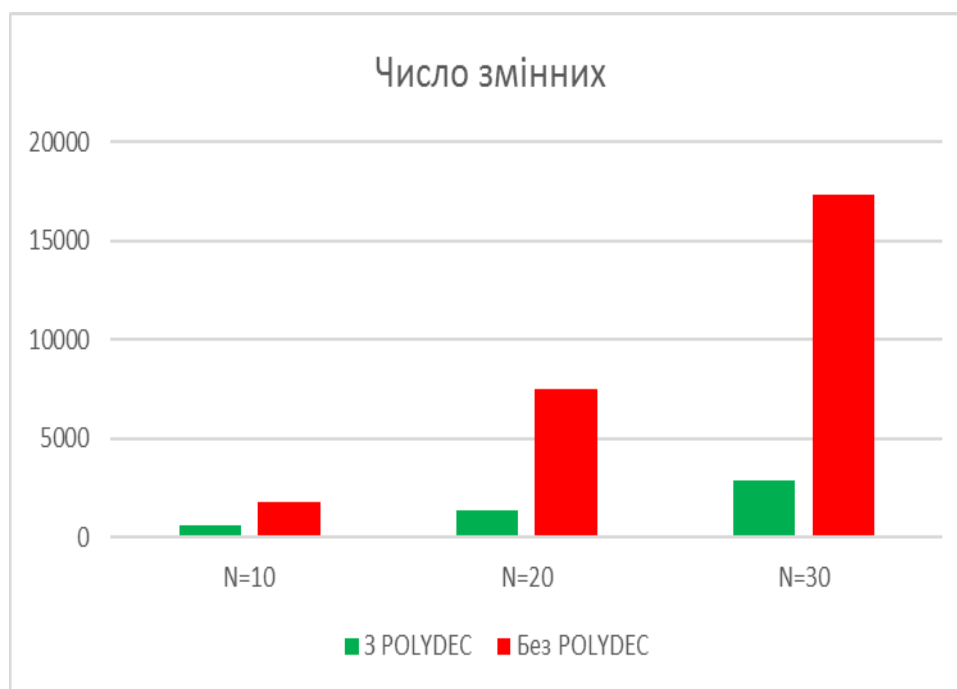


Рисунок 5.30 – Порівняння числа змінних для різних задач упаковки довільних багатогранників з застосуванням та без застосування метода POLYDEC

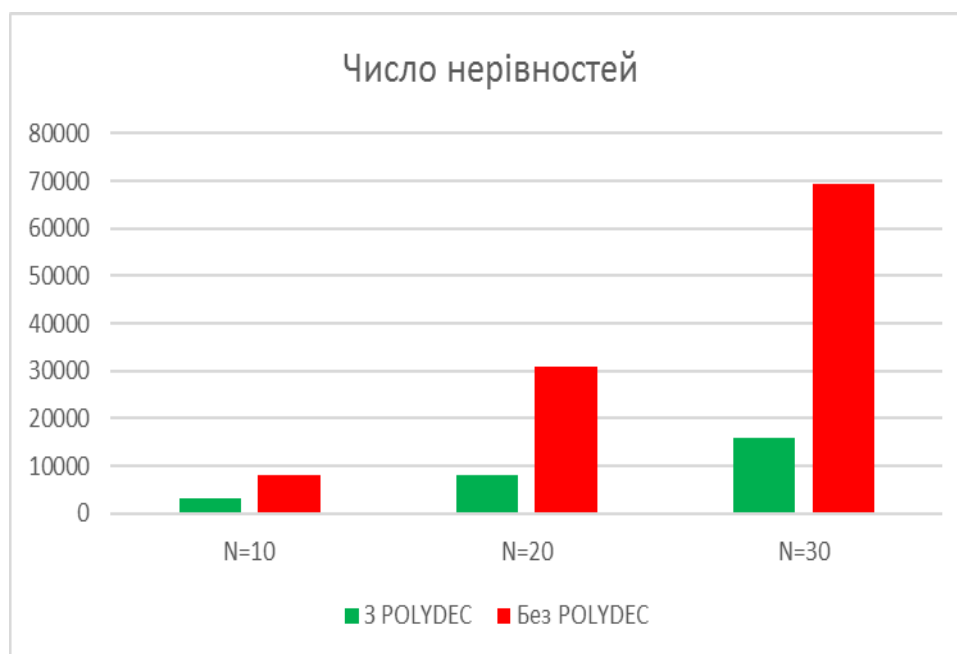


Рисунок 5.31 – Порівняння числа нерівностей для різних задач упаковки довільних багатогранників з застосуванням та без застосування метода POLYDEC

У прикладі для $N = 10$ неопуклих багатогранників (рис. 5.1):

Середній час пошуку одного локального екстремум становив:

- 1380 с без використання методу POLYDEC;
- 283 с з використанням методу POLYDEC.

Число нерівностей і змінних:

- 1791 і 7934 без використання методу POLYDEC;
- 626 і 3086 з використанням методу POLYDEC на останній

ітерації.

У прикладі для $N = 20$ неопуклих багатогранників:

Середній час пошуку одного локального екстремум становив:

- 75026.31 с без використання методу POLYDEC;
- 4980.74 с з використанням методу POLYDEC.

Число нерівностей і змінних:

- 7471 і 30916 без використання методу POLYDEC;
- 1334 і 8028 з використанням методу POLYDEC на останній

ітерації.

У прикладі для $N = 30$ неопуклих багатогранників:

– Локальний мінімум не був знайдений за обмежений час (72 години) без використання методу POLYDEC.

– Середній час пошуку одного локального екстремуму склав 35289.34 с з використанням методу POLYDEC.

Число нерівностей і змінних:

- 17334 і 69336 без використання методу POLYDEC;
- 2817 і 16047 з використанням методу POLYDEC на останній

ітерації.

5.4 Висновки по розділу

Проведені обчислювальні експерименти переконливо підтвердили конструктивність розроблених засобів математичного моделювання основних відношень довільних багатогранників, адекватність побудованої математичної моделі базової задачі, ефективність запропонованих стратегій, методів побудови стартових точок і пошуку локально оптимальних розв'язків задачі упаковки *OPP* для різних опуклих контейнів та задачі кластерінгу *OPC* для різних охоплюючих оболонок.

Проведено порівняння результатів задачі упаковки довільних багатогранників з найкращими світовими результатами. Для кожного прикладу мінімальний об'єм контейнера, знайдений методом, що пропонується в цьому дослідженні, виявляється менше, ніж найкращий розв'язок. Проведено порівняння результатів розв'язання задач упаковки довільних багатогранників з застосуванням та без застосування методу POLYDEC.

Результати обчислювальних експериментів наведені в таких роботах: [32, 35, 39, 46].

Список джерел, які використано у даному розділі наведено у повному списку використаних джерел [28, 71, 92, 211]

ВИСНОВКИ

У дисертації розв'язано важливу наукову задачу оптимальної упаковки неопуклих багатогранників на основі отриманих нових фундаментальних, теоретично обґрунтованих результатів, включаючи створення конструктивних засобів математичного моделювання, побудову нових математичних моделей, розробку ефективних методів розв'язання.

Основні наукові результати дисертації:

а) набув подальшого розвитку метод *phi*-функцій: вперше побудовані *phi*-функції, псевдонормалізовані *phi*-функції, квазі *phi*-функції та псевдонормалізовані квазі *phi*-функції як засоби математичного моделювання обмежень розміщення для задачі *OPP*, що дозволяє описати в аналітичному вигляді: неперетин довільних багатогранників; включення багатогранників в опуклий контейнер; мінімально допустимі відстані між довільними багатогранниками та між багатогранниками та границею контейнера. Формалізовано обмеження балансу у вигляді системи нелінійних нерівностей з гладкими функціями;

б) вперше побудована математична модель задачі *OPP* у вигляді задачі нелінійного програмування (що включає всі глобально оптимальні розв'язки) для неопуклих багатогранників в опуклому контейнері, границя якого формується за допомогою сферичних, циліндричних, еліптичних поверхонь та площини з урахуванням обмежень розміщення та обмежень балансу, що дозволяє використовувати сучасні *NLP*-солвери;

в) вперше побудована математична модель задачі кластерінгу неопуклих багатогранників (*OPC* – Optimal Polytopes Clustering) в сферичній, кубоїдній та циліндричній областях мінімального об'єму, що

дозволяє генерувати ефективні допустимі стартові точки для пошуку локальних екстремумів задачі *OPP*;

г) набули подальшого розвитку методи розв'язання задач геометричного проектування: запропонована стратегія розв'язання задачі *OPP* та розроблені ефективні методи для основних її реалізацій, які на відміну від існуючих підходів: враховують одночасно *неперервні трансляції* та *обертання* об'єктів, *мінімально допустимі відстані* і *обмеження балансу*; дозволяють отримувати локально оптимальні розв'язки для задач *OPP*, що є кращими за значенням цільової функції (порівняно з *benchmark instances* – відомими опублікованими результатами);

д) створено програмне забезпечення для розв'язання основних реалізацій задачі *OPP*;

е) створено програмний модуль для розв'язання задачі *OPC*.

є) виконано порівняння результатів обчислювальних експериментів та відомих світових аналогів, отримано найкращі значення функції цілі для усіх експериментів, що порівнювались, та покращено час розв'язання – для багатьох тестових прикладів. Виконано аналіз ефективності методу декомпозиції POLYDEC.

Отримані наукові результати являють собою подальший розвиток теорії математичного моделювання і обчислювальних методів в геометричному проектуванні.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Сергиенко И.В., Шило В.П. Задачи дискретной оптимизации: Проблемы, методы решения, исследования // НАН Украины; Институт кибернетики. К.: Наукова думка, 2003. 264с.
2. Сергієнко І.В. Інформатика та комп'ютерні технології. // Київ: Наук. думка, 2004. 432 с.
3. Сергієнко І.В. Методи оптимізації та системного аналізу для задач трансобчислювальної складності. // К.: Академперіодика, 2010. 296 с.
4. Klyushin D.A, Lyashko S.I., Nomirovsky D.A., Petunin Yu.,I., Semenov V.V. Generalized solutions of operator equations and extreme elements //Springer.- 2012. New York, Dordrecht, Heidelberg, London. 200 p.
5. Lyashko S.I. Generalized Optimal Control of Linear Systems with Distributed Parameters // Dordrecht, Boston, London.- Kluwer Academic Publishers.2002. 455 p.
6. Грицик В.В., Шевченко А.І., Кісельова О.М. Математичні методи оптимізації та інтелектуальні комп'ютерні технології моделювання складних процесів і систем з урахуванням просторових форм об'єктів // Донецьк: ІІІ "Наука і освіта", 2011. 480 с..
7. Шевченко А. И. Актуальные проблемы теории искусственного интеллекта // Шевченко АИ–Київ: ІІІІ «Наука і освіта. 2003.
8. Hulianytskyi L.F., Riasna I.I. Formalization and classification of combinatorial optimization problems. In: Optimization Methods and Applications (eds. Butenko S., Pardalos P.M., Shylo V.). // Cham: Springer International Publishing AG, 2017. P. 239-250.
9. Гуляницький Л.Ф. Диверсифікація пошуку в алгоритмах оптимізації мурашиними колоніями // Теорія оптимальних рішень. 2017. С. 47-57.
10. Гуляницький Л.Ф., Мулеса О.Ю. Прикладні методи комбінаторної оптимізації. // К.: Видавничо-поліграфічний центр «Київський університет»,

2016. 142 с.

11. Бідюк П.И., Касицкий А.В. Анализ сходимости метода оптимизации Розенброка // Системш технологи, № 3, 2011. С. 51-60.

12. Бідюк П.И., Демківський Є.О., Бідюк О.П. Аналіз даних з використанням байєсівських моделей // Наукові вісті НТУУ "КПІ", 2012, № 1, С. 40-54.

13. Крак Ю.В. Оптимізація обчислень при побудові рівнянь руху маніпуляційних роботів // Доповіді АН УРСР. Сер. А. 1988. № 9. С. 72 - 76.

14. Крак Ю.В. Оптимізаційні постановки та методи в задачах дослідження маніпуляційних систем // Вісник Київського університету. Серія фізико-математичні науки. 2000. №2. С. 56-62.

15. Стецюк П.И. Приближенный метод эллипсоидов // Кибернетика и системный анализ. 2003. №.3. С. 141-146.

16. Шор Н.З., Журбенко Н.Г., Лиховид А.П., Стецюк П.И. Развитие алгоритмов недифференцируемой оптимизации и их приложения // Кибернетика и системный анализ. 2003. №.4. С. 80 - 94.

17. Shor N.Z., Stetsyuk P.I. Lagrangian bounds in multiextremal polynomial and discrete optimization problems // Journal of Global Optimization. Vol. 23. 2002. P. 1-41.

18. Киселёва Е.М., Шор Н.З. Непрерывные задачи оптимального разбиения множеств: теория, алгоритмы, приложения: монография // К.: Наук. думка, 2005. 564 с.

19. Kiseleva E.M., Stepanchuk T. On the Efficiency of a Global Non-differentiable Optimization Algorithm Based on the Method of Optimal Set Partitioning // Journal of Global Optimization. 2003. V. 25 P. 209-235.

20. Шило В.П., Шило О.В. Решение задачи о максимальном разрезе графа методом глобального равновесного поиска // Кибернетика и системный анализ. 2010. № 5. С. 68-79.

21. Шило В.П., Рошин В.А., Градинар И.П. Приближённое решение задачи нахождения максимального ρ -плотного множества вершин графа // Компьютерная математика. Киев: Ин-т кибернетики им. В.М. Глушкова НАН Украины, 2011. С. 157–164.
22. Р. Базилевич, Р. Кутельмах. Декомпозиційні алгоритми для розв'язування задачі комівояжера // Вісн. Нац. ун-ту «Львів. політехніка». Комп'ют. науки та інформ. технології. 2007. N 598. С. 138—148.
23. Р. Базилевич, А. Ждан. Ієрархічна кластеризація складних схем // Вісн. Нац. ун-ту «Львів. політехніка». Комп'ют. науки та інформ. технології. 2008. N 616. С. 41-46.
24. Martínez A., Álvarez-Valdés R., Bennell J., Ruiz R. and Tamarit J. M. A Matheuristic for the Irregular Bin Packing with free rotation. // *European Journal of Operational Research*. DOI: 10.1016/j.ejor.2016.09.043.
25. Fasano G., Pinter J. Modeling and Optimization in Space Engineering. // Springer Opt. and Its Appl. 2013. №12. 404 p.
26. Fasano G., Pinter J. Optimized Packings and Their Applications. // Springer Opt. and its Appl. 2015. vol.105. 326 p.
27. Korte A.C. J., Brouwers H. J. H. Random packing of digitized particles. // *Powder Technology*. 2013. vol.233. P. 319-324.
28. Liu X., Liu J., Cao A., Yao Z. HAPE3D – a new constructive algorithm for the 3D irregular packing problem. // *Frontiers of Information Technology & Electronic Engineering*. 2015. 16(5). P. 380–390. DOI: 10.1631/FITEE.1400421.
29. Smeets B., Odenthal T., Vanmaercke S., Ramon H. Polygon – based contact description for modeling arbitrary polyhedra in the Discrete Element Method. // *Computer Methods in Applied Mechanics and Engineering*. 2015. 290. P. 277–289. DOI: 10.1016/j.cma.2015.03.004.
30. Верхотуров М. А., Верхотурова Г. Н., Данилов К. В., Ягудин Р. Р. Упаковка складних тривимірних об'єктів в прямокутний контейнер на базі

дискретно-логічного представлення інформації. // *Інформаційні технології високопродуктивних обчислень*. 2014. 16(4-2). С. 378–383.

31. Панкратов А.В., Романова Т.Е., Стоян Ю.Е. Псевдонормализованные квази ϕ -функции для многогранников. // *Радиоэлектроника и информатика*. 2015. № 3. С. 22-26.

32. Панкратов А.В., Романова Т.Е., Стоян Ю.Е., Чугай А.М. Задача оптимизации упаковки многогранников в сферическом и цилиндрическом контейнерах. // *Восточно-Европейский журнал передовых технологий*. 2016. № 1/4 (79). С. 39-47.

33. Chugay A., Stoian Ye. Cluster packing of concave non-oriented polyhedra in a cuboid. // *Advanced Information Systems*. 2018. № 2(1). P.16-21.

34. Стоян Ю.Е., Романова Т.Е., Панкратов А.В. Математическая модель задачи оптимальной компоновки многогранников в выпуклой многогранной области. // *Системи управління, навігації та зв'язку*. 2016. № 3 (39). С. 62-66.

35. Stoian Y., Pankratov A., Romanova T. Optimal clustering of polyhedra. // *Bionics Intelligence*. 2017. № 2 (89). P. 12-22.

36. Stoian Y. E., Chugay A. M., Pankratov A. V., Romanova T. E. Two Approaches to Modeling and Solving the Packing Problem for Convex Polytopes. // *Cybernetics and Systems Analysis*. 2018. № 4 (54). P. 585-593.

37. Чугай А.М., Панкратов О.В., Романова Т.Е., Стоян Ю.Е. Оптимізація процесу 3D-друку для SLS технології адитивного виробництва. // *Системи управління, навігації та зв'язку*. 2017. № 6 (46). С. 127-130

38. Стоян Ю. Е., Панкратов А.В., Романова Т.Е. Упаковка многогранников в выпуклый контейнер минимального объема. // *Системи управління, навігації та зв'язку*. 2018. № 2 (48). С. 48-54

39. Стоян Ю.Е. Псевдонормализованная квази ϕ -функция для невыпуклых многогранников. *Современные проблемы машиностроения: тез. докл. конф. молодых ученых и специалистов*, Харьков: Ин-т. проблем

машиностроения им. А. М. Подгорного НАН Украины. 2015. С. 18.

40. Стоян Ю.Е., Панкратов А.В., Романова Т.Е., Чугай А.М. О NLP-задаче оптимальной упаковки невыпуклых многогранников в сферическом и цилиндрическом контейнерах. Математическое моделирование, оптимизация и информационные технологии: тез. докл. 5-ой Межд. науч. конф., Кишинев, Молдавия. 2016. С. 345-348

41. Стоян Ю.Е., Романова Т.Е., Панкратов А.В. Упаковка невыпуклых многогранников в кубоиде минимального объема. Информационные системы и технологии: тез. докл. 5-ой Международная научно-техническая конф., Коблево. 2016. С. 129-130

42. Stoian Yu. Mathematical model and solution algorithm for packing problem of convex polyhedra. Information Technologies and Computer Modelling: тез. докл. международной научно-практической конф., Ивано-Франковск – Яремча. 2016. С. 187-188

43. Стоян Ю.Е., Панкратов А.В., Романова Т.Е. Об особенностях задачи оптимизации компоновки невыпуклых многогранников. Математичне та програмне забезпечення інтелектуальних систем: тезиси докладов XIII міжнародної науково-практичної конференції, Днепропетровск. 2016. С. 198-199

44. Stoian Yu. Packing polyhedrons using quasi-phi-functions. Современные проблемы машиностроения: тез. докл. конф. молодых ученых и специалистов, Харьков: Институт проблем машиностроения им. А.М. Подгорного НАН Украины. 2016. С. 33.

45. Chugay A., Pankratov A., Romanova T. Stoian Yu. Optimization of the process of 3D-Printing for SLS technologies of additive production. Application of Information and Communication Technology and Statistics in Economy and Education: тез. докл. 7-ой международной конференции. – София, Болгария. 2017. FA24.4p.

46. Стоян Ю.Е. Математическое моделирование оптимизации упаковки многогранников в выпуклой многогранной области. Радиоэлектроника и молодёжь в XXI веке: тез. докл. XXI Международного молодёжного форума, Харьков: ХНУРЭ. 2017. С. 104-105.

47. Stoian Y.E., Romanova T.E., Pankratov A.V. Packing of polyhedra within a convex container. Problem of Decision Making under Uncertainties: тез. докл. XXX международной конференции, Вильнюс, Литва. 2017. С. 116

48. Romanova T.E., Pankratov A.V., Stoian Y.E. Packing of Polytopes within Spherical and Cylindrical Containers. ESICUP meeting: тез. докл. 14-ой международной конференции, Льеж, Бельгия. 2017. С. 23.

49. Стоян Ю.Е. Упаковка многогранников в выпуклый контейнер минимального объема. Современные проблемы машиностроения: тез. докл. конф. молодых ученых и специалистов, Харьков: Ин-т. проблем машиностроения им. А. М. Подгорного НАН Украины. 2018. С. 18.

50. Romanova T., Bennell J.A., Stoyan Y., Pankratov A. Packing of arbitrary polyhedra with continuous rotations using nonlinear optimisation. ESICUP meeting: тез. докл. 15-ой международной конференции, Zoetermeer, Netherlands. 2018. P. 17-18

51. Araújo, L.J.P. Özcan, E., Atkin, J.A.D., Baumer, M. Analysis of irregular three-dimensional packing problems in additive manufacturing: a new taxonomy and dataset. // International Journal of Production Research. 2018. doi:10.1080/00207543.2018.1534016

52. Cephas M. The Impact and Application of 3D Printing Technology // International Journal of Science and Research (IJSR). 2014. Vol. 3(6). P. 2148-2152

53. Egeblad J., Nielse B. K., Odgaard A. Fast neighborhood search for two - and three - dimensional nesting problems. // European Journal of Operations Research. 2007. №183(3). P. 1249–1266.

54. Mona Saffarzadeh, Gregory James Gillispie, Philip Brown, Selective

laser sintering (sls) rapid prototyping technology: a review of medical applications. // Conference: Rocky Mountain Bio-engineering Symposium. 2016.

55. <https://www.nefab.com/en/insights/load-optimization/>

56. <https://www.thalesgroup.com/en>

57. Modeling and Optimization in Space Engineering Editors: Fasano, Giorgio, Pintér, János D. (Eds.). 2013

58. Space Engineering Modeling and Optimization with Case Studies Editors: Fasano, Giorgio, Pintér, János D. (Eds.). 2016

59. Optimized Packings with Applications Editors: Fasano, Giorgio, Pintér, János D. (Eds.). 2015

60. Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. АЛГОРИТМЫ: построение и анализ. // М.: Вильямс. 2005. 1296 с.

61. Pasha A. Geometric bin packing algorithm for arbitrary shapes. // Master thesis. University of Florida. 2003. 87 p.

62. Vivek A., Dobowsky S. Application of a Model-free Algorithm for the Packing of Irregular Shaped Objects in Semiconductor Manufacture. // IEEE International Conference on Robotics and Automation. 2000. vol. 2. P. 1545-1550

63. Egeblad J., Nielsen B.K., Brazil M. Translational packing of arbitrary polytopes. // Computational Geometry. 2009. vol. 42. P. 269–288

64. Dyckhoff H. A typology of cutting and packing problems. // European Journal of Operational Research. 1990. vol. 44. P. 145-159.

65. Wäscher G., Haußner H., Schumann H. An improved typology of cutting and packing problems. // European Journal of Operational Research. 2007. vol. 183. P. 1109-1130.

66. Cagan J., Shimada K., Yin S. A survey of computational approaches to three-dimensional layout problems. // Comp.-Aided Des. 2002. vol. 34. P. 597–611.

67. Sriramya P. A., Varthini P.B. State-of-the-Art Review of Bin Packing Techniques. // Eur. J. Scien. Res. 2012. Vol. 86, N 3. P. 360–364.

68. Jia X., Gan M., Williams R.A., Rhodes D. Validation of a digital packing algorithm in predicting powder packing densities. // Powder Tech. 2007. Vol. 174. P. 10–13.
69. Li S.X., Zhao J. Sphere assembly model and relaxation algorithm for packing of non-spherical particles. // Chin. J. Comp. Phys. 2009. Vol. 26, N 3. P. 167–173.
70. Li S.X., Zhao J., Lu P., Xie Y. Maximum packing densities of basic 3D objects. // Chin. Scien. Bull. 2010. Vol. 55, N 2. P. 114–119.
71. Stoyan Y.G., Gil N.I., Pankratov A., et al., Packing Non-Convex Polytopes into a Parallelepiped. // Technische Universitat Dresden. 2004. <http://www.math.tu-dresden.de/~scheith/ABSTRACTS/PREPRINTS/04-non-conv.pdf>
72. Chernov N., Stoyan Y., Romanova T. Mathematical model and efficient algorithms for object packing problem. // Comput. Geom.: Theory and Appl. 2010. Vol. 43, N 5. P. 535–553.
73. Stoyan Y., Chugay A. Mathematical modeling of the interaction of non-oriented convex polytopes. // Cyber. and Syst. Anal. 2012. Vol. 48, N 6. P. 837–845.
74. Stoyan Yu., Chugay A. Construction of radical free phi-functions for spheres and non-oriented polytopes. // Rep. of NAS of Ukraine. 2011. №12. P. 35-40.
75. Sutou A., Dai Y. Global optimization approach to unequal sphere packing problems in 3D. // Journal of optimization theory and applications. 2002. Vol. 114, № 3. P. 671-694.
76. Stoyan Y., Yaskov G., Scheithauer G. Packing of various radii solid spheres into a parallelepiped. // Central European Journal of Operations Research. 2003. № 11. P. 389-407.
77. Stoyan Y., Yaskov G. Packing congruent hyperspheres into a hypersphere. // Journal of Global Optimization. 2012. Vol. 52, № 4. P. 855-868.
78. Yaskov G., Stoyan Y., Chugay A. Packing identical spheres into a

cylindrical domain. // Proceedings of the Workshop on Cutting Stock Problems. Miercurea-Ciuc, Romania. 2006. P. 75-82.

79. Stoyan Y. G., Yaskov G. N. Packing identical spheres into a cylinder. // International Transactions in Operational Research. 2010. Vol. 17, N 1. P. 51-70.

80. Hifi M., M'Hallah R. A Literature Review on Circle and Sphere Packing Problems: Models and Methodologies. // Advances in Operations Research. 2009. doi:10.1155/2009/150624.

81. Сёмкин В. В. Упаковка выпуклых трёхмерных геометрических объектов с поворотами в связные области. // Дис. к-та техн. наук, 01.05.02. Харьков. 2014, 189 с.

82. Fasano G. A Global Optimization point of view for non-standard packing problems. // Journal of Global Optimization. 2013. Vol. 55, Issue 2. P. 279-299.

83. Petrov M. S., Gaidukov V. V., Kadushnikov R. M. Numerical method for modelling the microstructure of granular materials. // Powder Metallurgy and Metal Ceramics. 2004. Vol. 43, Issue 7-8. P. 330-335.

84. Egeblad J., Nielsen B. K., Brazil M. Translational packing of arbitrary polytopes. // Computational Geometry. 2009. Vol. 42, Issue 4. P. 269-288.

85. Мизгулин В. В., Кадушников Р. М., Алиевский Д. М., Алиевский В. М. Моделирование плотных материалов методом упаковки сферополиэдров. // Компьютерные исследования и моделирование. 2012. Т. 4, № 4. С. 757-766.

86. Lee Y., Yang C. -T., Chien C. -S. A 3D ellipsoid-based model for packing of granular particles. // International Journal of Computer Applications in Technology. 2003. Vol. 17, Issue 3. P. 148-155.

87. Gan M., Gopinathan N., Jia X., Williams R. A. Predicting packing characteristics of particles of arbitrary shapes. // KONA. 2004. № 22. P. 82-93.

88. Birgin E. G., Martinez J. M., Ronconi D. P. Optimizing the packing of cylinders into a rectangular container. // European Journal on Operations Research. 2005. Vol. 160, Issue 1. P. 19-33.

89. Egeblad J., Garavelli C., Lisi S., Pisinger D. Heuristics for Container Loading of Furniture. // *European Journal of Operational Research*. 2010. Vol. 200, Issue 3. P. 881-892.
90. Fasano G. MIP-based heuristic for non-standard 3D-packing problems. // *4OR: Quarterly Journal of the Belgian, French and Italian Operations Research Societies*. 2008. Vol. 6, Issue 3. P. 291-310.
91. Egeblad J., Nielsen B.K., Odgaard A. Fast neighborhood search for two- and three-dimensional nesting problems. // *Eur. J. Oper. Res.* 2007. Vol. 183, N 3. P. 1249–1266.
92. Stoyan Y., Gil N., Scheithauer G., Pankratov A., Magdalena I. Packing of convex polytopes into a parallelepiped. // *Optimization*. 2005. Vol. 54, N 2. P. 215 – 235.
93. Stoyan Yu., Chugay A. Packing cylinders and rectangular parallelepipeds with distances between them into a given region. // *European Journal of Operational Research*. 2009. Vol. 197, Issue 2. P. 446-455.
94. Torquato S., Jiao Y. Dense polyhedral packings: Platonic and Archimedean solids. // *Physics Review E*. 2009. Vol. 80, Article ID 041104. P. 041104-1–041104-21.
95. George J. A., Robinson D. F. A heuristic for packing boxes into a container. // *Computers and Operational Research*. 1980. № 7. P. 147-156.
96. Pisinger D. Heuristics for the container loading problem. // *European Journal of Operational Research*. 2002. Vol. 141, Issue 2. P. 382-392
97. Brunetta L., Grégoire P. A general purpose algorithm for three-dimensional packing. // *INFORMS Journal on Computing*. 2005. Vol. 17, Issue 3. P. 328-338.
98. Bortfeldt A. A., Gehring H., Mack D. parallel tabu search algorithm for solving the container loading problem. // *Parallel Computing*. 2003. Vol. 29, Issue 5. P. 641-662

99. Crainic T. G., Perboli G., Tadei R. A Two-Stage Tabu Search Heuristic for the Three-dimensional Bin Packing Problem. // *European Journal of Operational Research*. 2009. Vol. 195, Issue 3. P. 744-760.
100. Dai Z., Cha J. An octree method for interference detection in computer aided 3-D packing. // *Advances in Design Automation 1994: Proceedings of the 20th ASME Design Automation Conference*. 1994. № 1. P. 29-33.
101. Dai Z., Cha J. An octree based heuristic algorithm for 3-D packing. // *Advances in Design Automation: Proceedings of the 20th ASME Design Automation Conference*. 1994. № 2. P. 125-133.
102. Martello S., Pisinger D., Vigo D. The tree-dimensional bin packing problem. // *Operations Research*. 2000. Vol. 48, Issue 2. P. 256-267.
103. den Boef E., Korst J., Martello S., Pisinger D., Vigo D. Erratum to “The tree-dimensional bin packing problem: Robot-Packable and orthogonal variants of packing problems”. // *Operations Research*. 2005. Vol. 53, Issue 4. P. 735-736.
104. Landon M. D., Balling R. J. Optimal packaging of complex parametric solids according to mass property criteria. // *Journal of Mechanical Design*. 1994. Vol. 116, Issue 2. P. 375-381.
105. Dixon L.C.W., Szegő G. P. Towards global optimization. // Amsterdam: North-Holland. 1975,1978. Vols. 1-2
106. Glover F. Heuristics for integer programming using surrogate constraints. // *Decisions Science*. 1977. Vol. 8. P. 156-166.
107. Glover F. Tabu Search - Part I. // *ORSA Journal on Computing*. 1989. Vol. 1, N 3. P. 190-206.
108. Glover F. Tabu Search - Part II. // *ORSA Journal on Computing*. 1990. Vol. 2, N 1. P. 4-32.
109. Valenzuela C. L., Wang P. Y. VLSI placement and area optimization using a genetic algorithm to breed normalized postfix expressions. // *IEEE Transactions on Evolutionary Computation*. 2002. Vol. 6, Issue 4. P. 390-401.

110. Vinoth J. A., Batri K. Layout problem optimization in VLSI circuits using genetic algorithm. // International Journal of Engineering Sciences & Research Technology. 2014. Vol. 3, Issue 4. P. 711-716.

111. Muntean O. An evolutionary approach for the 3D packing problem. // Proceedings of the International Conference on Knowledge Engineering, Principles and Techniques, KEPT2007. Cluj-Napoca, Romania. June 6-8, 2007. P. 193-200.

112. Ikonen I., Biles W., Kumar A., Wissel J. C., Ragade R. K. A genetic algorithm for packing three-dimensional non-convex objects having cavities and holes. // Proceedings of 7th International Conference on Genetic Algorithms. 1997. P. 591-598.

113. Goldberg D.E. Genetic algorithms in search, optimization and machine learning. // Addison-Wesley. Reading, MA, USA. 1984. 412p..

114. McCulloch W.S., Pitts W. A logical calculus of the ideas immanent in nervous activity. // Bulletin of Mathematical Biophysics. 1943. Vol. 5. P. 115-137.

115. Dorigo M., Maniezzo V., Colorni A. Ant system: optimization by a colony of cooperating agents. // IEEE Transactions on Systems, Man and Cybernetics. 1996. Vol. 26, N 1. P. 29-41.

116. Kirkpatrick S., Gellat C.D., Vecchi M. P. Optimization by simulated annealing. // Science. 1983. Vol. 220, № 4598. P. 671-679.

117. Jang S.-H. A study on three dimensional layout design by the simulated annealing method. // Journal of Mechanical Science and Technology. 2008. Vol. 22, Issue 11. P. 2016-2023.

118. Jajodia S., Minis I., Harhalakis G., Proth J. M. CLASS: Computerized Layout Solutions Using Simulated Annealing. // International Journal of Production Research. 1992. Vol. 30, Issue 1. P. 95-108.

119. Szykman S., Cagan J. A simulated annealing approach to three-dimensional component packing. // Journal of Mechanical Design. 1995. Vol. 95, Issue 2A. P. 308-314.

120. Szykman S., Cagan J. Constrained three dimensional component layout using simulated annealing. // Journal of Mechanical Design. 1997. Vol. 119, Issue 1. P. 28-35.
121. Szykman S. Optimal product layout using simulated annealing. // PhD Thesis. Carnegie Mellon University. 1995. 212 p.
122. Huang M. D., Romeo F., Sangiovanni-Vincentelli A. An efficient general cooling schedule for simulated annealing. // IEEE International Conference on Computer Aided Design – Digest of Technical Papers. Santa Clara, CA. 11-13 November, 1986. P. 381-384.
123. Hustin M. D., Sangiovanni-Vincentelli A. TIM, a new standard cell placement program based on the simulated annealing algorithm. // IEEE Physical Design Workshop on placement and Floorplanning. 1987. 122 p.
124. Szykman S., Cagan J. Synthesis of optimal non-orthogonal routes. // Journal of Mechanical Design. 1996. Vol. 118, Issue 3. P. 419-424
125. Kolli A., Cagan J., Rutenbar R. A. Packing of generic, three dimensional components based on multi-resolution modeling. // Proceedings of the 22nd ASME Design Automation Conference (DAC-1479). Irvine, CA. 19-22 August, 1996.
126. Campbell M. I., Amon C. H., Cagan J. Optimal three-dimensional placement of heat generating electronic components. // Journal of Electronic Packaging. 1997. Vol. 119, Issue 2. P.106-113.
127. Hooke R., Jeeves T. A. Direct search solution of numerical and statistical problems. Journal of the Association for Computing Machinery. // Journal of the Association for Computing Machinery. 1980. Vol. 8, Issue 2. P. 212-229.
128. Torczon V., Trosset M. From evolutionary operation to parallel direct search: pattern search algorithms for numerical optimization. // Computing Science and Statistics. 1998. Vol. 29. P. 396-401.
129. Yin S., Cagan J. An extended pattern search algorithm for three-dimensional component layout. // ASME Journal of Mechanical Design. 2000. Vol.

122, Issue 1. P. 102-108.

130. Yin S., Cagan J., Hodges P., Li X. Layout of an automobile transmission using three-dimensional shapeable components. // *Journal of Mechanical Design*. 2004. Vol. 126, Issue 1. P. 188-191.

131. Yin S., Cagan J. Exploring the effectiveness of various patterns in an extended pattern search layout algorithm. // *Journal of Mechanical Design*. 2004. Vol. 126, Issue 1. P. 22-28.

132. Aladahalli C., Cagan J, Shimada K. Objective function effect based pattern search – theoretical framework inspired by 3D component layout. // *Journal of Mechanical Design*. 2007. Vol. 129, Issue 3. P. 243-254.

133. Aladahalli C., Cagan J, Shimada K. Objective function effect based pattern search – an implementation for 3D component layout. // *Journal of Mechanical Design*. 2007. Vol. 129, Issue 3. P. 255-265.

134. Allen S. D., Burke E. K., Kendall G. A hybrid placement strategy for the three-dimensional strip packing problem. // *European Journal of Operational Research*. 2011. Vol. 209, Issue 3. P. 219-227.

135. Peng Y., Zhang D., Francis Y., Chin L. A hybrid simulated annealing algorithm for container loading problem. // *Proceedings of the first ACM/SIGEVO Summit on Genetic and Evolutionary Computation*. Shanghai, China. 12-12 June, 2009. P. 919-928.

136. Eisenbrand F., Funke S., Karrenbauer A., Schömer E., Reichel J. Packing a trunk – now even with a twist. // *Proceedings of ACM Symposium on Solid and Physical Modeling (SPM)*. Cambridge. 2005. P. 197-206.

137. Панкратов А. В. Математические модели, методы и информационные технологии решения оптимизационных задач размещения геометрических объектов. // Дис. д-ра техн. наук, 01.05.02. Харьков. 2013. 334 с.

138. Романова Т.Е. Средства построения математических моделей оптимизационных задач размещения геометрических объектов и их

применение. Дис. // д-ра техн. наук, 01.05.02. Харьков. 2003. 324 с.

139. Рвачев В.Л. Об аналитическом описании некоторых геометрических объектов. // Доклады АН Украины. 1963. N 4. С. 765-767.

140. Рвачев В.Л., Ющенко Е.Л. О классе функций, удобных для аналитического описания некоторых геометрических образов. // Кибернетика и техника вычислений. К.: Наук. Думка. 1964. С. 18-31.

141. Рвачев В.Л., Стоян Ю.Г. К задаче распознавания непересечения фигур специального вида. // Кибернетика. 1965. N 6. С.85-94.

142. Рвачев В.Л., Стоян Ю.Г. Алгоритмы построения неравенств, которым удовлетворяют параметры размещения непересекающихся тел. // Кибернетика. 1966. N 7. С. 82-92.

143. Рвачев В.Л., Стоян Ю.Г., Глушко А.Г. К вопросу о поиске абсолютного экстремума в одной задаче оптимального раскроя. // Алгоритмические языки, оптимизация, программирование. 1966. Вып. 2. С. 70-81.

144. Стоян Ю.Г., Гиль Н.И. Свойства и способы реализации функции плотного размещения. // Препр. АН Украины. Ин-т кибернетики; 18. - К. 1972. 46 с.

145. Hopper E., Turton V.C.H. A review of the application of meta-heuristic algorithms to 2D strip packing problems. // Artificial Intelligence Review. 2001. Vol. 16. P. 257-300.

146. Стоян Ю.Г., Пономаренко Л.Д. Сумма Минковского и годограф вектор-функции плотного размещения. // Докл. АН УССР. Сер. А. 1977. № 10. С. 888-890.

147. Stoyan Yu.G., Loiko A.F. Automation of arrangement design of engineering systems of block design. // Mathematical model. Engineering simulation. 1998. N. 15. P. 815-825.

148. Гиль Н.И. Математическое моделирование нерегулярного

размещения плоских геометрических объектов в системах автоматизации проектирования (теоретические основы, методы, приложения). // Автореф. дис. д-ра техн. наук: 05.12.13. Ин-т техн. кибернетики АН БССР. Минск. 1990. 32 с.

149. Стоян Ю.Г., Гиль Н.И. Метод асимптотического перебора локальных экстремумов. // Препр. АН УССР. Ин-т пробл. Машиностроения. Харьков. 1974. 1-74. 48 с.

150. Стоян Ю.Г., Яковлев С.В. Математические модели и оптимизационные методы геометрического проектирования. // К.: Наук. Думка. 1986. 267 с.

151. Романова Т.Е. Система построения математической модели класса $E_k(R^2)$ -задач размещения. // Автореф. дис. канд. физ.-мат. наук: 05.13.16. Ин-т кибернетики им. В.М. Глушкова АН УССР. К. 1990. 16 с.

152. Стоян Ю.Г. Об одном обобщении функции плотного размещения. // Доклады АН УССР. 1980. N 8. С. 70-74.

153. Магас С.Л. Методы решения экстремальных задач размещения многоугольных геометрических объектов на полосе. // Автореф. дис. канд. физ.-мат. наук: 01.05.03. ВЦ АН СССР -М. 1984. 20с.

154. Stoyan, Y., Novozhilova, M., & Kartashov, A. Mathematical model and method of searching for a local extremum for the non-convex oriented polygons allocation problem. // European Journal of Operational Research. 1996. № 92. P. 193-210.

155. Пандорин А.К. Методы и алгоритмы построения раскладок в системах управления раскроем изотропных материалов. // Автореф. дис...канд. техн. наук: 01.05.03. Харьковский ин-т радиоэлектроники. Харьков. 1988. 17с.

156. Stoyan Y.G., Zlotnik M.V., Chugay A.M. Solving an optimization packing problem of circles and non-convex polygons with rotations into a multiply connected region. // Journal of the Operational Research Society. 2012. № 63(3). P. 379-391.

157. Стоян Ю.Г., Галата А.Я. О плотной упаковке параллелепипедов

произвольных размеров в параллелепипеде наименьшего объема. // Кибернетика. 1972. № 2. С. 81-86.

158. Стоян Ю.Г., Пономаренко Л.Д. Алгоритм приближенного решения задачи плотнейшей упаковки набора параллелепипедов в параллелепипеде с областями запрета. // Автоматика и вычислительная техника. 1975. №1. С. 46-54.

159. Стоян Ю.Г., Винарский В.Я. Методы оптимального двухрядного периодического размещения трехмерных многогранников. // Препр.АН Украины. Ин-т пробл. машиностр. Харьков. 1985. 211. 26 с..

160. Пономаренко Л.Д., Туранов И.Н. Один подход к построению плотных упаковок выпуклых многогранников в заданной области трехмерного пространства. // Препр.АН Украины. Ин-т пробл. машиностр. Харьков. 1979. 154. 13 с.

161. Панкратов А.В., Пономаренко Л.Д. Линейная невыпуклая модель и метод локальной оптимизации в задачах упаковки выпуклых многогранников. // Препр.АН Украины. Ин-т пробл. машиностр. Харьков. 1987. 262. 28 с.

162. Панкратов А.В., Пономаренко Л.Д. Оптимизационная задача упаковки многогранников (линейная модель и метод решения). // Автоматизированные системы управления и приборы автоматики. 1987. N 83. С.14-19.

163. Гиль Н.И., Черноморец А.А. Метод построения поверхности 0-уровня Ф-функции произвольных многогранников. // Препр.АН Украины. Ин-т пробл. машиностр. Харьков. 1992. 358. 29 с.

164. Гиль Н.И., Черноморец А.А. Особенности алгоритмической реализации построения годографа функции плотного размещения многогранных объектов. // Препр. АН УССР. Ин-т пробл. машиностр. Харьков. 1992. № 363. 33 с.

165. Scheithauer G., Stoyan Yu., Romanova T. Mathematical modeling of interactions of primary geometric 3D objects. // Cybernet. Systems Anal. 2005. Vol.

41. P. 332-342.

166. Stoyan Y., Terno J., Gil M., Romanova T., Scheithauer G. Construction of a Φ -function for two convex polytopes. // *Applicationes Mathematicae*. 2002. Vol. 2, № 29. P. 199 - 218.

167. Стоян Ю.Г., Придатко Д.И., Романова Т.Е., Уварова М.А. Φ -функции параллелепипедов и цилиндров. // Докл. НАН Украины. 2002. № 10. С. 68-72.

168. Stoyan Yu., Scheithauer G., Pridatko D. Φ -function for primary 3D objects. // *Prepr. Technische Univarsitat Dresden; MATH-NM-15-2002*. Dresden, 2002. 27 p.

169. Стоян Ю.Г., Придатко Д.И., Романова Т.Е. Φ -функции объектов с конической и сферической границами. // Докл. НАН Украины. 2004. № 3. С. 25-29.

170. Стоян Ю.Г., Придатко Д.И., Романова Т.Е., Шайтхауэр Г. Φ -функции объектов, имеющих пространственную форму границы конус, цилиндр или параллелепипед. // Докл. НАН Украины. 2004. № 5. С. 27-31.

171. Стоян Ю.Г., Евсеева Л.Г., Романова Т.Е. Φ -функции усеченных конусов. // Докл. НАН Украины. 2005. № 7. С. 30-35.

172. Стоян Ю.Г., Романова Т. Е., Евсеева Л. Г., Придатко Д. И. Математическое моделирование взаимодействий геометрических объектов, имеющих пространственную форму тора и шара. // Доп. НАН України. 2006. № 4. С. 22-26.

173. Stoyan, Yu.G., Yaskov G.N., Scheithauer G. Packing spheres of various radii into a parallelepiped. // *Prepr. Technische Univarsitat Dresden; MATH-NM-15-2001*. Dresden, 2001. 25 p.

174. Магдаліна І.В. Математична модель та методи наближеного розв'язання задачі упакування опуклих багатогранників. // Автореф. дисс. канд. техн. наук: 01.05.02. Ін-т проблем машинобудування ім. А.М. Підгорного НАН

України. Харків. 2001. 19 с.

175. Стоян Ю.Г., Придатко Д.И. Математическая модель и метод упаковки различных цилиндров в параллелепипеде. // Прикладная радиоэлектроника. Состояние и перспективы развития - 1-й Междунар. радиоэлектронный форум - МРФ-2002: Сб. науч. тр. Харьков: АН ПРЭ, ХНУРЭ. 2002. Ч. 2. С.227 - 230.

176. Yaskov G., Stoyan Yu., Chugay A. Packing a great number of identical spheres into a composed domain. // Proc. of the 2nd ESICUP Meeting.- University of Southampton (United Kingdom). 2005. P. 24.

177. Yaskov G.N., Stoyan Yu.G. Packing identical spheres into a cylinder. // Proc. of Workshop on Cutting Stock Problems, Sapientia University, Miercurea-Ciuc (Romania). 2005. P. 15.

178. Стоян Ю. Г., Гиль Н. И., Муравьева М. С. Ф-функция n-мерных параллелепипедов. // Доп. НАН України. 2005. № 3. С. 22-27.

179. Чугай А.М. Математична модель і метод розв'язання оптимізаційної задачі розміщення циліндрів і паралелепедів у призмі з урахуванням спеціальних обмежень. // Автореферат дисс. канд. техн. наук: 01.05.02. Ін-т проблем машиностроєння. Харків. 2006. 18 с.

180. Stoyan Y., Yaskov G. Packing identical spheres into a rectangular parallelepiped. Intelligent Decision Support. Current Challenges and Approaches. // Betriebswirtschaftlicher Verlag Dr. Th. Gabler, GWV Fachverlage GmbH. Wiesbaden, Germany. 2008. P. 47-67.

181. Stoyan Y., Chugay A. Packing cylinders and rectangular parallelepipeds with distances between them into a given region. // EJOR. 2009. Vol. 197, N 2. P. 446-455

182. Stoyan Y.G., Romanova T., Pankratov A., Chugay A. Optimized object packings using quasi-phi-functions. // Springer Optimization and Its Applications. 2015. Vol. 105. P. 265-293.

183. Стоян Ю.Г., Чугай А.М. Математическая модель и метод решения задачи размещения сфероцилиндров и цилиндров с учетом специальных ограничений. // Электронное моделирование. 2008. Т.30, №5. С. 3-21.

184. Stoyan Yu., Chugay A. Packing cylinders and rectangular parallelepipeds with distances between them into a given region. // European Journal of Operational Research. 2009. Vol. 197. P. 446-455.

185. Стоян Ю.Г., Чугай А.М. Математическое моделирование взаимодействия неориентированных выпуклых многогранников. // Кибернетика и системный анализ. 2012. Т. 48, № 6. С. 36-44.

186. Стоян Ю.Г., Чугай А.М. Построение свободной от радикалов Φ -функции для шара и неориентированного многогранника. // Доповіді НАН України. 2011. № 12. С. 35-40.

187. Сёмкин В.В., Чугай А.М. Нормализованная Φ -функция сферических сегментов. // Доповіді НАН України. 2012. № 12. С. 41-48.

188. Сёмкин В. В., Чугай А.М. Нормализованные Φ -функции сферического сегмента с параллелепипедом, цилиндром, шаром и сфероцилиндром. // Вісник Харківського національного університету ім. В.Н. Каразіна. Серія «Математичне моделювання. Інформаційні технології. Автоматизовані системи управління»: зб. наук. праць. Харків: ХНУ ім. В.Н. Каразіна. 2012. № 1037. С. 190-201.

189. Сёмкин В.В., Чугай А.М. Нормализованная Φ -функция параллелепипеда и сфероцилиндра. // Доповіді НАН України. 2013. № 2. С. 36-41.

190. Стоян Ю. Г., Сёмкин В. В., Чугай А. М. Построение начальных точек и поиск локальных экстремумов задачи компоновки 3D объектов в цилиндрической области. // Доповіді НАН України. 2013. № 12. С. 52-58.

191. Стоян Ю. Г., Сёмкин В. В., Чугай А. М. Оптимизация компоновки трёхмерных объектов в многосвязной области с учётом кратчайших расстояний.

// Кибернетика и системный анализ. 2014. № 3. С. 58-70.

192. Сёмкин В. В., Чугай А. М. Поиск локальных экстремумов в задаче плотной упаковки неориентированных сфероконусов. // Искусственный интеллект. 2014. № 1. С. 74-79.

193. Стоян Ю.Г., Чугай А.М. Математическая модель задачи упаковки параллелепипедов и сфер в параллелепипеде. // Матеріали Всеукраїнської науково-практичної конференції “Інформатика та системні науки – 2011”. Полтава: ПУСКУ. 2011. С. 297-300.

194. Чугай А.М., Яськов Г.Н. Оптимальная упаковка трехмерных геометрических объектов. // Математичне та програмне забезпечення інтелектуальних систем: тези доповідей ІХ Міжнар. наук.-практ. конф. Дніпропетровськ. 2011. С. 281-283.

195. Сёмкин В. В., Чугай А. М. Математическое моделирование отношений геометрических объектов, построенных с помощью шаров и цилиндров. // Съома міжнар. наук.-практ. конф. «Математичне та імітаційне моделювання систем. МОДС 2012». Тези доповідей. (Чернігів-Жукин, 25-28 червня 2012 р.) Чернігів-Жукин. 2012. С. – 238-240.

196. Сёмкин В.В., Панкратов А. В., Чугай А. М. Квази Ф-функция для сфероконусов с поворотами. // Радиоэлектроника и информатика. 2014. № 1. С. 30-33.

197. Stoyan Yu., Chugay A. Packing Cuboids and Spheres. // Proc. 10th ESICUP Meeting. – Lille (France). 2013. P. 21.

198. Сёмкин В. В., Чугай А. М. Моделирование взаимодействия неориентированных сфероконусов. // Материалы 4-й международной конференции «Математическое моделирование, оптимизация и информационные технологии». (Кишинёв, 25-28 марта 2014 г.) Кишинёв. 2014. Т. 2. С. 416-420.

199. Сёмкин В. В., Чугай А. М. Решение задачи плотной упаковки

неориентированных трёхмерных объектов. // Системный анализ и информационные технологии: материалы 16-й Международной научно-технической конференции SAIT 2014. (Київ, 26-30 мая 2014 г.) К.: УНК «ИПСА» НТУУ «КПИ». 2014. С. 262.

200. Chugay A., Pankratov A., Romanova T. Optimal packing of convex polytopes. // Тезисы докладов IV Международная научно-технической конференции «Информационные системы и технологии» (ИСТ-2015). Харьков: Харьковский национальный университет радиоэлектроники. 2015. С. 11-12

201. Chugay A., Pankratov A., Romanova T. Optimized object packings using quasi-phi-functions. // Тези доповідей XXI Всеукраїнська наукова конференція «Сучасні проблеми прикладної математики та інформатики». Львів: Львівський національний університет імені Івана Франка. 2015. С. 23-24.

202. Чугай А.М., Панкратов А.В., Романова Т.Е. Сравнение двух подходов к моделированию и решению задачи упаковки многогранников. // Тезисы докл. 6-й международной научно-технической конференции "Информационные системы и технологии" (ИСТ-2017). Харьков, Коблево. 2017. С. 141-142.

203. Куратовский К. Топология. // Москва, Мир. 1966. Т.1. 594 с.

204. Кроуэлл Р., Фокс Р. Введение в теорию узлов. // Москва: Мир. 1967. 348 с.

205. Chernov N., Stoyan Y., Romanova T., Pankratov A. Phi-functions for 2D objects formed by line segments and circular arcs. // Advances in Operations Research. 2012. doi:10.1155/2012/346358.

206. Chernov N., Stoyan Y., Romanova T. Mathematical model and efficient algorithms for object packing problem. // Computational Geometry: Theory and Applications. 2010. Vol. 43, N 5. P. 535-553.

207. Стоян Ю.Г., Панкратов А.В., Романова Т.Е., Чернов Н.И. Квази phi-функции для математического моделирования отношений геометрических

об'єктів. // Доповіді НАН України. – 2014. – Т.9. – С. 49-54.

208. Belov, Gleb, 2002. A Modified Algorithm for Convex Decomposition of 3D Polyhedra,” // Technical report MATH-NM-03-2002, Institut for Numerische Mathematik, Technische Universitet, Dresden, <http://www.math.tu-dresden.de/~belov/cd3/cd3.ps>.

209. Chernov N., Stoyan Y., Pankratov A., Romanova T. Quasi-phi-functions and optimal packing of ellipses. // Subm. to J. of Glob. Optim. 2014. Vol. 65. Issue 2. P 283-307.

210. Barber C. B., Dobkin D. P., Huhdanpaa H. T. The Quickhull algorithm for convex hull. // GCG53, The Geometry Center, Minneapolis. 1993. (<ftp.geom.umn.edu/pub/software/qhull.tar.Z>)

211. Wachter A., Biegler L.T. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. // Math. Program. 2006. № 106 (1). P. 25–57.

Додаток А

Список публікацій здобувача

1. Stoian Y. E., Chugay A. M., Pankratov A. V., Romanova T. E. Two Approaches to Modeling and Solving the Packing Problem for Convex Polytopes // Cybernetics and Systems Analysis. 2018. № 4 (54). P. 585-593.
2. Стоян Ю. Е., Панкратов А.В., Романова Т.Е. Упаковка многогранников в выпуклый контейнер минимального объема // Системи управління, навігації та зв'язку. 2018. № 2 (48). С. 48-54
3. Chugay A., Stoian Ye. Cluster packing of concave non-oriented polyhedra in a cuboid // Advanced Information Systems. 2018. № 2(1). P.16-21
4. Stoian Y., Pankratov A., Romanova T. Optimal clustering of polyhedra // Bionics Intelligence. 2017. № 2 (89). P. 12-22
5. Чугай А.М., Панкратов О.В., Романова Т.Є., Стоян Ю.Є. Оптимізація процесу 3D-друку для SLS технології адитивного виробництва // Системи управління, навігації та зв'язку. 2017. № 6 (46). С. 127-130
6. Стоян Ю.Е., Романова Т.Е., Панкратов А.В. Математическая модель задачи оптимальной компоновки многогранников в выпуклой многогранной области // Системи управління, навігації та зв'язку. 2016. № 3 (39). С. 62-66.
7. Панкратов А.В., Романова Т.Е., Стоян Ю.Е., Чугай А.М. Задача оптимизации упаковки многогранников в сферическом и цилиндрическом контейнерах // Восточно-Европейский журнал передовых технологий. 2016. № 1/4 (79). С. 39-47.
8. Панкратов А.В., Романова Т.Е., Стоян Ю.Е. Псевдонормализованные квази ϕ -функции для многогранников // Радиоэлектроника и информатика. 2015. № 3. С. 22-26.
9. Стоян Ю.Е. Упаковка многогранников в выпуклый контейнер

минимального объема. Современные проблемы машиностроения: тез. докл. конф. молодых ученых и специалистов, Харьков: Ин-т. проблем машиностроения им. А. М. Подгорного НАН Украины. 2018. С. 18.

10. Romanova T., Bennell J.A., Stoyan Y., Pankratov A. Packing of arbitrary polyhedra with continuous rotations using nonlinear optimisation. ESICUP meeting: тез. докл. 15-ой международной конференции, Zoetermeer, Netherlands. 2018. P. 17-18

11. Stoian Y.E., Romanova T.E., Pankratov A.V. Packing of polyhedra within a convex container. Problem of Decision Making under Uncertainties: тез. докл. XXX международной конференции, Вильнюс, Литва. 2017. С. 116

12. Стоян Ю.Е. Математическое моделирование оптимизации упаковки многогранников в выпуклой многогранной области. Радиоэлектроника и молодёжь в XXI веке: тез. докл. XXI Международного молодёжного форума, Харьков: ХНУРЭ. 2017. С. 104-105.

13. Romanova T.E., Pankratov A.V., Stoian Y.E. Packing of Polytopes within Spherical and Cylindrical Containers. ESICUP meeting: тез. докл. 14-ой международной конференции, Льеж, Бельгия. 2017. С. 23.

14. Chugay A., Pankratov A., Romanova T. Stoian Yu. Optimization of the process of 3D-Printing for SLS technologies of additive production. Application of Information and Communication Technology and Statistics in Economy and Education: тез. докл. 7-ой международной конференции. – София, Болгария. 2017. FA24.4p.

15. Stoian Yu. Packing polyhedrons using quasi-phi-functions. Современные проблемы машиностроения: тез. докл. конф. молодых ученых и специалистов, Харьков: Институт проблем машиностроения им. А.М. Подгорного НАН Украины. 2016. С. 33.

16. Stoian Yu. Mathematical model and solution algorithm for packing problem of convex polyhedra. Information Technologies and Computer Modelling: тез. докл. международной научно-практической конф., Ивано-Франковск – Яремча. 2016. С. 187-188

17. Стоян Ю.Е., Панкратов А.В., Романова Т.Е., Чугай А.М. О NLP-задаче оптимальной упаковки невыпуклых многогранников в сферическом и цилиндрическом контейнерах. Математическое моделирование, оптимизация и информационные технологии: тез. докл. 5-ой Межд. науч. конф., Кишинев, Молдавия. 2016. С. 345-348

18. Стоян Ю.Е., Романова Т.Е., Панкратов А.В. Упаковка невыпуклых многогранников в кубоиде минимального объема. Информационные системы и технологии: тез. докл. 5-ой Международная научно-техническая конф., Коблево. 2016. С. 129-130

19. Стоян Ю.Е., Панкратов А.В., Романова Т.Е. Об особенностях задачи оптимизации компоновки невыпуклых многогранников. Математичне та програмне забезпечення інтелектуальних систем: тезиси докладов XIII міжнародної науково-практичної конференції, Днепропетровск. 2016. С. 198-199

20. Стоян Ю.Е. Псевдонормализованная квази ρ -функция для невыпуклых многогранников. Современные проблемы машиностроения: тез. докл. конф. молодых ученых и специалистов, Харьков: Ин-т. проблем машиностроения им. А. М. Подгорного НАН Украины. 2015. С. 18.

Додаток Б

Акт про впровадження засобів математичного моделювання задач ОРР в навчальному процесі. Акт про використання результатів дисертаційної роботи в ТОВ «Клауд Воркс». Лист підтримки від Дж. Фазано. Довідка про використання програмного модуля на кафедрі «Прикладне матеріалознавство та обробка матеріалів» Національного університету «Львівська Політехніка»

ЗАТВЕРДЖУЮ

Ректор
Харківського національного
університету радіоелектроніки
В.В. Семенець
«___» _____ 2018 р.



АКТ

про використання результатів дисертаційної роботи
Стояна Ю. Є. «Математичне моделювання та методи розв'язання
оптимізаційних задач упаковки довільних багатогранників»

Ми, що нижче підписалися, начальник Навчального відділу ХНУРЕ Міхнова А.В., завідувач кафедри прикладної математики професор Тевяшев А.Д., професор кафедри прикладної математики Гусарова І.Г. склали цей акт про те, що результати кандидатської дисертаційної роботи Стояна Ю. Є., щодо математичного і комп'ютерного моделювання та методів розв'язання задач оптимальної упаковки еліпсоїдів, впроваджені у навчальний процес на кафедрі прикладної математики в дисциплінах «Математичне та комп'ютерне моделювання в системах підтримки прийняття рішень» (теми 4-5 робочої програми д.т.н, проф. Т.Є. Романової, О.М. Хлуд, Ю. Є. Стояна), «Моделювання геометричних об'єктів» (теми 2, 5, 7 робочої програми к.т.н., доц. Г.М. Яськова, Т.Є. Романової, Ю. Є. Стояна. Комісія відзначає, що використання нових результатів в області математичного моделювання забезпечує поглиблене вивчення перспективних напрямків прикладної математики, а також знайомство студентів старших курсів з практичними аспектами застосування фундаментальних результатів при розв'язанні прикладних та наукових задач.

Начальник Навчального відділу

А.В. Міхнова

Завідувач кафедри
прикладної математики

А.Д. Тевяшев

Професор кафедри
прикладної математики

І.Г. Гусарова

Згідно з оригіналом.

Вчений секретар спецради Д 64.052.02

Підпис
Печатка

Л.В. Колесник



АКТ
про використання результатів дисертаційної роботи
на здобуття наукового ступеня кандидата технічних наук
Стояна Юрія Євгеновича

Комісія у складі голови комісії директора Свербілова Д.С., членів комісії програміста Горносталь О.А. та програміста Луцка В.А. склала даний акт про використання результатів дисертаційної роботи, виконаної Стояном Юрієм Євгеновичем.

Запропоновані Стояном Ю.Є. засоби, моделі, методи та програмні модулі використовуються в ТОВ «Клауд Воркс» для розв'язання: задач оптимізації процесу 3D-друку в адитивному виробництві, що використовує SLS технологію, за рахунок одночасного друку відразу декількох деталей із забезпеченням щільного заповнення усього об'єму робочої камери 3D принтера; задач оптимального пакування вантажів у довільних контейнерах у сфері логістики.

Програмні модулі, розроблені Стояном Ю.Є., можуть бути застосовані при розробці сучасних інтелектуальних систем, які дозволяють за вихідними даними одержати найкращий варіант розміщення довільних 3D об'єктів відповідно до заданого критерію якості.



Голова комісії: Свербілов Д.С.

Члени комісії: Горносталь О.А.

Луцик В.А.

Згідно з оригіналом.

Вчений секретар спецради Д 64.052.02

Підпис
Печатка

Л.В. Колесник



Thales Alenia Space
Strada Antica di Collegno, 253
10146 Torino
Italia

Kharkiv National University of Radio Electronics
61166, Nauky Avenue 14, Kharkiv, Ukraine
Specialized Academic Council D 64.052.02

November 12, 2018

Ladies and Gentlemen,

I am writing with reference to the proposed PhD Thesis "Mathematical modeling and solution methods for optimal polyhedra packing problems" by Stoian Yu.E. (scientific adviser is Prof. Dr. Romanova T.E.) This work is devoted to optimized packings of arbitrary polyhedra in containers of different types.

The development of innovative computer technologies focused on solving problems of 3D layout optimisation is a promising area of space engineering. These problems are being studied in our company "Thales Alenia Space", which is specialised in the development, construction, testing and management of innovative space systems. One major problem arising in spacecraft design entails the optimal placement of equipment of given shapes and sizes, taking into account geometrical constraints (e.g. allowable distance between objects, feasible positioning and orientation), as well as mechanical requirements (concerning the overall centre of mass and the axial/centrifugal moments of inertia).

As a co-editor, with Prof. J.D. Pinter (Leigh University, PA, USA) of 'Modeling and Optimization in Space Engineering 2019' (third volume included in the Springer Optimization Series, dedicated to modelling and optimization in space engineering) I would like to express my appreciation for the significant contribution offered by the doctoral candidate, in particular as a co-author of the chapters 'Optimized Packings in Space Applications Part I-II'.

The results conveyed in these works are expected to provide remarkable enhancements to the already existing topical technology, not only from the modelling and algorithmic perspective, but in particular in terms of practical added value in the specific field. I strongly endorse Stoian's research work.

Yours faithfully,
Giorgio Fasano, CMath CSci FIMA

A handwritten signature in black ink, appearing to read 'Giorgio Fasano', written over a horizontal line.

Senior researcher and optimization specialist at Thales Alenia Space
Chartered Mathematician (IMA, UK), Chartered Scientist (Science Council, UK),
Fellow of The Institute of Mathematics and its Applications (UK)

Згідно з оригіналом.
Вчений секретар спецради Д 64.052.02

Підпис
Печатка

Л.В. Колесник

ДОВІДКА

про використання програмного продукту, розробленого **Стояном Юрієм Євгеновичем** за темою дисертаційної роботи «МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ ТА МЕТОДИ РОЗВ'ЯЗАННЯ ОПТИМІЗАЦІЙНИХ ЗАДАЧ УПАКОВКИ ДОВІЛЬНИХ БАГАТОГРАННИКІВ»

Підтверджую, що розроблений Стояном Ю.Є. програмний модуль оптимізації упаковки довільних неорієнтованих багатогранників застосовується на кафедрі «Прикладне матеріалознавство та обробка матеріалів» Національного університету «Львівська політехніка» для розв'язання задачі оптимального заповнення заданого об'єму частинками порошку металевих сплавів несферичної форми. Ці результати важливі при проведенні досліджень, щодо можливості застосування вітчизняних порошків титанових сплавів несферичної форми, які пройшли процес гідрування і дегідрування, є на порядок дешевшими та можуть з успіхом використовувати в адитивних технологіях.

Професор, доктор технічних наук,
завідувач кафедри прикладного
матеріалознавства та обробки матеріалів



З.А.Дурягіна

Згідно з оригіналом.

Вчений секретар спецради Д 64.052.02

Підпис
Печатка

Л.В. Колесник

Додаток В

Точки локальних екстремумів для прикладів наведених в розділі 5.

Приклад 5.8 а) $U^* = (0.146702, 14.145826, 12.962412, 2.374913, 1.694902, 1.4268349.965800, 15.319625, 17.288539, 1.017404, 2.711385, -2.94064113.212947, 2.534816, 17.606976, 0.744542, -1.242426, -0.3805068.745566, 23.218093, 20.454604, 9.953979, -2.355555, 9.1660568.206502, 10.052497, 22.114048, -0.313907, -2.500803, 1.04466421.292557, 1.703646, 21.988579, -0.461570, -2.352012, -1.90768914.748182, 11.610017, 1.437365, 3.188657, -0.912789, -7.85398226.352202, 22.183139, 5.734517, -0.819743, 1.937358, -3.1415937.964806, 8.000000, 9.8.25435, 0.000000, 1.570796, -2.21191421.940909, 4.800922, 2.747301, -1.589630, 0.938539, -1.23890813.512377, -0.000000, 17.670896, 2.673757, 1.998148, -0.4710610.215681, 6.789123, 2.025954, -1.274823, -0.723070, 1.7673693.076395, 21.383110, -0.000000, 0.890543, 0.743160, -0.8117674.564984, 18.691769, 2.828685, 2.102027, 0.674136, -2.0011414.628181, 16.397652, 2.088671, 0.465791, 1.192226, -0.28481424.450054, -0.000000, 3.917589, -2.230292, 1.269676, -0.58006815.523911, 12.657982, 24.405546, 1.163953, -2.412970, -3.2708125.739121, 1.884029, 23.467463, 0.494274, -2.284237, 1.90984917.646370, 15.714004, 10.522558, 14.699745, 3.388658, 18.03239922.781987, 23.751414, 18.686472, 2.389314, -1.207973, -3.138890)$

Приклад 5.8 б) $U^* = (36.885338, 35.503460, 25.793767, 2.352422, -1.484154, 0.086969, 17.759101, 34.915428, 30.057749, 0.174306, 2.423390, -4.312353, 34.593055, 25.020752, 20.974990, 5.235413, 0.913794, -1.875005, 34.419338, 19.477799, 34.162771, 1.458252, -2.486805, -1.614763, 20.490120, 31.154521, 14.941944, -2.199053, -0.767431, -3.092972, 24.066427, 22.121937, 21.854494, -0.852794, 0.738633, 0.702417, 19.063026, 21.313143, 22.881822, 3.801741, -3.141593, 1.999161, 16.690816, 15.045102, 35.118178, -0.169211, -2.454051, 1.820032, 19.510741, 23.076703, 32.661046, 2.403476, 1.843122, 0.410177, 26.843120, 34.039331, 25.688975, 0.042371, 0.280032, 0.552572, 27.169256, 16.432759, 23.620684, 8.236123, 3.238827, 10.995574, 28.659460, 11.689372, 34.789489, 3.101671, 2.341055, -0.027806, 34.680623, 26.205145, 22.713123, -2.194970, 2.007079, -2.719988, 18.680270, 38.310628, 18.310600, 0.498419, 1.107853, -2.785763, 22.541060, 28.071549, 35.118178, 1.729903, 2.450304, -1.655780, 15.541818, 31.622953, 26.772319, -1.747177, -2.562292, -1.780642, 13.209524, 15.321636, 27.087553, 2.300598, 1.631955, 3.210999, 29.804461, 11.689372, 27.446104, 0.168263, -2.351495, 0.949931, 13.230676, 35.964309, 31.128280, 0.846016, 2.153830, -3.578464, 28.474021, 31.735408, 14.971348, 0.442033, 0.559795, 0.915184)$

Приклад 5.9 а) $U^* = (26.395426, 15.489902, 12.868681, 1.308769, -1.573759, -2.96620822.851181, 23.239835, 34.978180, 2.323712, -2.381771, -0.1006370.310377, 11.736847, 6.116693, -0.875572, -0.900468, -1.4844274.844118, 2.230925, 22.351128, -0.687721, -1.067179, -0.2807207.640375, 11.171033, 19.027615, -0.744174, -1.512555, -2.52478726.507508, 4.469599, 31.110945, 0.808027, -1.249047, -3.14159114.149772, 25.015349, 15.167317, 2.704961, -1.311110, 1.0185662.726343, 3.155701, 5.077110, -1.366478, -0.770656, 1.0741639.040130, 15.772993, 35.455540, -0.870483, 2.450319, 1.11075726.588874, 13.960332, 0.000126, 1.716891, -0.802696, -1.9601341.844389, 26.515139, 20.154412, 0.824127, 1.846142, 0.70985623.820193, 8.870133, 20.910827, 2.332518, -2.248892, 2.09007013.634684, 8.084435, 2.915751, 1.715901, 0.945981, -2.2538870.000035, 4.994693, 33.170111, 1.171817, 2.288598, -1.82085426.514303, 23.839943, 22.078979, -1.278199, 1.014581, 1.82361422.184139, 0.005682, 29.241222, -2.889707, 2.424283, -1.8405911.710353, 24.304255, 36.170138, -2.483893, -2.363986, -0.74394126.589031, 22.755349, 0.000014, -1.537214, 0.799850, 1.8100949.091753, 21.578496, 9.213167, -57.734150, 3.307780, -58.119464-0.000000, 7.521890, 0.000892, 1.231812, 0.785438, -1.570638)$

Приклад 5.9 б) $U^* = (9.902225, 9.393371, 7.996663, 3.296967, -1.686705, 2.50701317.311068, 15.699306, 12.343732, -3.804433, -1.137316, 2.3729229.172085, 14.570876, 29.643429, 4.006582, -2.251636, -0.4606010.013914, 9.117350, 25.079979, -1.428944, -1.461457, 3.4940742.058090, 25.875300, 14.747434, -2.687593, -0.881755, 0.83418816.799660, 9.743490, 19.375550, -2.222014, -0.824407, -0.7275550.908771, 18.692953, 21.315018, 1.422734, 2.415880, 1.35308711.548776, 0.005831, 19.169499, -3.166295, 0.787118, -1.53837220.041142, 25.439347, 12.946920, 2.893913, -1.549175, -2.58299121.903550, 18.016177, 0.000000, -0.326295, 0.786898, 3.36310218.687164, 17.170841, 20.386026, -0.197124, -1.133313, 3.2260049.718482, 10.519814, 1.698415, 1.553300, -1.148614, 2.66964821.189591, 25.695342, 30.672555, 2.648730, -2.500726, 2.0322943.298494, 20.153591, 17.678304, -1.196050, 0.627422, 7.4644323.059678, 13.348452, 12.800703, 3.363189, -0.862823, -1.33572126.205645, 24.473157, 0.289255, 2.565316, -0.810968, -1.57079626.848265, 3.522731, 30.372514, -2.346990, 2.393382, -1.42775826.456757, 0.755760, 5.142705, -2.403902, 1.121389, -0.37648827.203174, 7.192161, 11.408748, -2.043499, 1.068471, -0.09550714.343266, 23.626411, 2.059538, 1.823905, -0.824635, 1.016044)$

Приклад 5.11 а) $U^* = (33.635599, 12.197806, 12.419157, -14.137167, 0.109655, 14.136998, 19.213486, 38.786560, 32.630569, -1.567333, -2.355824, -3.103097, 32.801248, 40.702230, 19.695719, -2.570687, -2.127422, 3.174818, 14.642014, 24.187588, 23.377829, -3.141593, 1.055030, -0.000000, 19.293036, 17.706039, 12.815501, -1.455724, -1.570018, 4.026495, 11.063847, 18.715221, 36.227580, -6.271296, 3.306984, -6.688813, 31.623107, 12.196455, 30.698477, -1.570796, 1.588430, -3.141593, 37.767882, 17.244959, 29.936817, -2.209273, 0.561385, -2.580647, 19.811783, 25.573824, 30.663328, 1.027631, -1.706161, -0.427401, 31.874179, 34.890155, 38.141828, -0.024955, 1.973881, -2.326127, 33.312219, 23.710505, 14.321323, 1.570649, -1.666049, -1.572347, 19.831661, 22.058182, 33.429051, -0.729622, -1.248272, -1.936990, 14.011216, 21.304589, 25.804491, -0.310614, 2.395478, -1.805456, 13.733464, 39.862027, 33.623917, 1.055896, 1.297176, -0.445549, 15.585902, 33.008563, 34.535772, -0.165757, 2.340882, 6.649465, 12.151540, 39.160489, 41.508576, 0.005053, 1.566059, -1.363021, 27.406056, 35.911785, 42.580843, 0.765368, 3.180029, 1.570796, 15.727584, 41.663492, 23.582192, 3.892012, -1.273873, 2.778021, 32.857865, 31.701386, 30.614872, 2.395845, -2.265664, 1.449464, 42.357986, 39.247226, 12.419157, 1.709608, -0.785630, 1.601260)$

Приклад 5.11 б) $U^* = (0.000000, 0.247732, 0.346498, 3.141591, -0.000539, -3.1415918.933159, 7.147082, 10.449994, 3.141590, 0.126245, -3.02467914.756857, 0.000000, 8.887587, -2.821785, 0.636120, 3.06881423.751922, 5.689519, 3.229294, 0.155492, -0.108187, -0.1397480.489362, 23.907382, 2.638823, 1.569056, 0.655362, -1.2388671.195528, 19.898379, 7.644867, 0.168782, 0.123714, 0.08037427.482020, 9.899495, -0.000000, 0.042740, -0.000000, -0.04274010.874776, 19.171518, 20.651053, -2.275599, 0.666042, 2.39654421.009296, 17.070043, 27.762589, 0.394511, 0.780684, -0.45863622.813717, 22.839851, 6.214195, 4.015686, 0.515308, -4.08310519.482020, 0.243084, -0.000000, 10.978990, -0.000000, -10.97899015.741742, 10.471603, 9.405139, -1.141746, 0.179666, 1.4605261.275677, 1.817703, 12.809674, 1.362920, 0.028713, -1.35928029.482020, 2.969848, 0.504091, -0.000000, -0.141897, 0.00000017.082020, 4.737046, 0.000000, -3.141593, 0.611452, 2.4980929.789175, -0.523474, 0.228723, 1.609838, -0.060110, -1.39019127.482020, 7.076081, 16.000000, -1.560872, 0.000000, 1.56087217.539324, 14.930905, 19.781149, -1.699237, -0.518021, 2.00931615.187821, 22.539831, 11.732208, 8.110095, 0.217269, -7.84699216.934753, 9.420935, 17.452733, -0.180171, 0.429428, -0.000000)$

Приклад 5.12 а) $U^* = (19.852580, 24.833315, 32.166202, 0.094288, -1.791160, -1.570796, 45.947283, 15.753494, 27.263080, 1.942535, -0.519779, -0.330706, 18.837646, 26.088630, 30.364960, -0.706214, 0.670405, -1.061161, 26.898815, 26.916115, 26.096039, -1.300665, -1.815344, 0.221072, 34.003521, 28.254670, 41.233245, -12.503079, 1.235236, 12.590390, 23.369450, 16.261245, 14.627254, -0.655696, -1.570796, -2.214297, 31.443299, 25.647428, 17.027254, -1.387796, 1.383482, 1.229173, 47.533960, 31.517233, 30.269742, -9.230870, -3.872723, -10.476921, 50.985078, 29.100305, 36.608427, 1.783387, 2.502149, 0.761030, 32.231415, 30.860645, 30.377584, -2.349334, 2.391940, -0.087290, 33.076431, 17.064420, 36.689520, 15.802251, -3.325628, 14.137167, 25.606499, 50.689829, 23.427690, -1.570796, 0.553938, 0.000000, 29.675564, 48.162174, 34.886605, -$

18.611215, 2.584737, -1.411642, -3.457078, -0.698197, 1.877628, 13.013315, 3.430249, -1.059805, 0.286976, 6.741888, -12.497368, 35.902762, -0.825652, 4.220511, -6.072105, -9.442387, -18.224740, 3.327959, 0.998857, -1.183780, -0.541782, 3.640439, 13.256167, 44.116350, -4.980404, 4.092208, -4.660112, 12.126208, 0.363863, 24.038960, 2.596117, -1.587286, 0.064206, -13.038672, -0.823511, 33.501782, 3.587907, -2.113603, -0.055918, -2.655539, 8.886616, 65.231285, 4.615605, -1.136222, 4.529230, 10.629075, -9.413928, 13.775132, -0.679790, -1.427126, 0.064103, 12.045860, -5.612703, 56.534382, 2.436961, 0.586881, -2.840143, -1.026743, 12.797447, 57.311241, 4.325417, 4.512673, 5.934530, -2.277752, 17.728676, 19.585085, -1.443018, 2.126125, 0.029649, 3.753127, -7.590258, 62.975251, -1.439995, 1.565883, -3.452643, -6.269700, -9.921543, 47.206052, 4.182542, 0.194096, 3.229591, -13.396572, -3.437769, 56.832257, 0.588942, 1.518240, 1.473310, 14.299652, 0.887968, 13.932027, -2.299760, 0.720091, -0.123962, -5.037827, 5.711933, 6.543939, -2.477302, 1.695151, 1.570796, -3.054571, -13.245166, 54.236365, -1.243516, -0.424368, -0.671724, 15.222739, 3.259466, 12.728776, -0.143338, -1.866823, 2.027643, -2.429031, -6.842636, 63.710905, 1.126128, 2.801732, 5.509938, 6.084515, 10.945059, 65.752429, 1.505862, -2.323874, -0.094078, 0.996091, 12.737957, 12.963198, -2.252659, -0.744077, 3.364922, -0.971186, 7.221345, 38.381308, -3.129114, 1.485505, 5.429102, -11.499708, -1.319377, 24.238989, -3.734575, 2.139874, 0.725786)

*

Приклад 5.23. $U = (-8.626455, 3.117095, -14.959977, 2.635264, -0.758525, -2.447398, 10.933822, 2.742218, 4.593312, 3.079391, 2.337685, 2.832284, 11.615462, 16.992569, -10.421272, 3.045367, -2.100178, -1.5763509, 19.7416, -0.725253, 1.660408, 2.768632, -2.066525, -1.069052, 13.978626, 3.942951, -8.837414, 0.613726, -4.416006, 1.117047, 7.32885, -4.195522, 11.539977, -3.264062, -2.294416, -5.342250, -1.796660, 5.869778, -15.066148, -0.290678, 0.957854, -0.518002, -6.69992, -16.178441, 2.588902, 2.630330, 2.085248, 3.128650, -4.796162, 9.292848, 20.564750, 0.529452, -1.039773, -0.306202, 3.829064, 4.973213, 9.799168, -1.414063, 2.240356, -2.446420, -4.240684, 22.610360, -1.748563, -3.616727, -0.483951, 1.869323, 19.710275, 5.932145, 4.043079, 0.906613, -0.695686, 2.299674, 10.991643, -19.850784, 3.862729, -1.312192, 1.352741, 3.028763, 16.585333, -6.260951, 6.103724, 2.795246, 2.541866, -0.346676, -11.641610, -5.946586, -12.408724, -1.014726, -5.636394, -0.485817, -14.125599, 15.194499, 14.016279, -0.224361, -2.539198, 0.170194, 8.818103, -1.257540, -21.282126, 0.407921, 0.979542, 1.500507, -4.000254, -0.406884, -9.694826, -1.593809, 2.442962, -5.076036, -1.467898, 3.439968, -11.402465, -2.540898, -1.357600, 0.415384, -16.991187, -13.925090, -1.051949, -0.743188, -0.679285, -0.187701, 11.121081, 2.664710, 12.869154, -3.013449, -2.360612, -1.819888, -18.339688, -1.635490, -9.539497, -4.070155, -0.299101, -2.242920, -5.477278, -4.223541, -12.175858, -2.097937, -1.055698, 1.140172, -1.746995, -19.711796, -7.022709, -1.805461, 0.463236, 3.113108, 2.771053, -2.830916, -8.257927, -3.105955, 0.634272, -3.181983, -7.327297, -12.600474, -10.206854, 1.360706, -0.535146, 2.095318, -18.178672, -13.346580, 4.865585, 1.413495, 0.800698, -1.165544, 16.747698, 7.112911, -11.105521, -0.779508, 1.431308, 2.244559, 21.454502, -8.049873, -2.464294, -4.621294, 1.328907, -1.294773, 14.878263, 15.732723, -3.851476, 1.867298, -2.086134, 2.438738)$

*

Приклад 5.24. $U = (2.627369, -0.407258, 1.431135, 0.128311, 2.628057, -2.506838, -5.913039, -2.151135, 6.904554, 1.493500, 1.504165, -1.127777, -1.500647, 6.271016, -0.352254, -1.942843, -2.656087, -0.693331, 8.046423, -4.712764, -3.116070, 0.930068, -4.532812, -3.243858, -3.754828, -9.108862, -4.343222, -3.370261, -1.626720, 1.454279, 0.537453, -9.875977, -11.896052, 0.640330, 0.964973, -5.073745, 11.265079, -0.957937, -3.017934, 2.385181, -2.831860, -1.543244, 3.021031, 5.511428, 8.420399, 3.457499, 2.110076, 4.535928, -8.013260, 7.466015, -3.010317, 2.321131, -0.568407, -3.615754, -4.981760, 3.737168, -11.062755, 2.297205, 2.110914, 2.682721, -0.606742, -6.476916, 9.558578, 2.404992, -1.344283, 0.186697, 13.249452, 6.370308, -4.817782, -2.795813, -1.873125, -0.332292, 9.585480, 3.035900, 3.665898, -2.335569, 1.317761, -0.942315, -10.258006, -2.020118, -7.637750, -0.840761, 1.673238, 1.364359, 6.595790, 2.768805, -6.722539, 1.711927, -2.260853, 4.270720, -12.174651, 3.879878, -3.244814, 4.303014, -1.292177, -4.070575, 7.217658, -6.088471, -12.255029, 2.595715, 1.357044, 2.843004, 5.609069, -7.154960, 7.244436, 1.658982, -2.033275, 1.602339, -4.590924, -1.066208, -14.735550, -1.318164, 4.665977, -5.899537, -7.182120, -2.907066, 10.815054, -0.427393, 2.262213, 0.327466, -7.770059, 1.220130, -3.788540, -1.397906, -2.067199, 2.636292, -3.762717, -0.503823, 0.719347, -1.758543, 1.533865, -1.178283, 2.220340, -0.757252, 9.073980, 0.729335, -3.707287, -2.641046, 3.858898, 5.065047, -7.773654, -1.954833, -1.167228, 3.602336, 0.042198, 3.634358, 12.402102, -1.494284, 2.016975, 0.423266, 1.318964, 8.973754, -10.828402, 0.875467, 1.275112, 0.952290, 5.898124, 10.834623, -3.472457, -1.979536, 2.508594, -0.019336, -4.610393, 2.427856, 14.069541, -0.398858, -1.460060, 0.388833, 14.920475, 0.116789, -4.023204, 1.323514, 0.231717, 1.856746, -7.087567, -5.444798, 6.669422, -1.103751, 1.156772, 2.678381, 6.233168, -11.448280, 2.015023, 1.735973, 1.145479, 1.924635, 8.639058, -4.258511, 5.949276, -1.490048, 3.790647, 2.990398, 10.573456, 2.627321, 7.512519, -2.696770, -1.547398, 3.907046, -14.001078, 1.698501, -6.059705, -0.929141, 0.746470, -2.908065, -13.741784, -1.909681, 0.073084, -0.667219, 0.164285, -0.061876, 11.293531, -5.824999, 8.978022, 1.087399, 2.565061, -1.589549, -6.891766, 0.685119, 2.899851, -1.305798, -1.512843, 0.455588, -8.295543, -7.747875, -10.142018, 2.189898, -0.918270, 1.032099, 11.066168, 9.892174, 1.497547, -2.589255, 2.821957, -0.157267, 3.998360, -9.353364, -5.704460, 1.769113, -0.682374, 2.794155, 4.402236, -6.942385, -4.204027, 1.691831, 2.297434, 0.291398, 5.401729, -8.943970, 4.062306, -1.381309, 1.934456, 2.797102, -2.236165, -14.368289, 5.068650, -1.733985, -1.522128, -1.688419, 15.286446, -0.419494, 4.595120, -3.076926, -1.893973, -5.977589, -5.169231, 13.669361, -5.697913, -0.699013, -0.659955, 2.387105, -2.693966, -2.767087, -1.738517, -2.264941, -0.830219, 2.713885, 6.069307, 0.508439, 14.760476, 0.219297, 1.635069, -1.958361, 11.785209, -9.677679, -3.156765, 2.064599, 0.519096, 3.278108, 8.005006, 7.258253, -11.158084, -2.146824, -1.360906, 3.792912, -1.351975, 14.062743, -7.281495, -3.854941, -2.489763, -2.310898, 2.354224, 13.005918, 7.449977, 2.482883, 2.538286, 1.160769, -11.658549, -9.631846, 0.044650, -2.627413, 0.642170, 4.934838, -4.335797, -5.547031, 1.985313, -2.263321, -0.873663, 0.151998, 14.015350, -5.480660, -4.435758, 2.311985, 0.573328, -2.786268, -13.372527, 7.210586, -3.912616, -1.133592, 0.394864, 3.993587, -8.880466, -13.212655, 1.236815, 4.053868, -2.357422, 3.039627, -5.427411, 4.869398, 6.802358, 2.205225, 1.075426, -2.665444, -12.410441, 2.248886, 9.910688, 2.701663, -5.225996, -5.606892, -14.748166, -6.141194, -1.662176, 0.957839, -0.394212, 2.588817, 6.693447, 9.196350, 11.340653, -2.688595, 1.871426, 2.260707, 1.989444, 3.380736, -2.122716, -0.969326, -2.485057, -2.929013, -3.019297, 7.154915, -7.979967, 0.926168, -2.070549, 2.267894, 7.535006, -0.967001, -9.163623, 0.947755, 1.529458, -1.198398, 9.321602, -10.455145, -0.435688, -2.995087, 0.903881, 3.844783, -0.389831, 2.558906, 15.834780, 2.466346, -1.336166, -1.466236, 1.105792, 5.383013, -15.069799, -0.601137, -1.375126, 1.683238, -7.895195, 6.277248, -12.499903, -1.065443, 1.404094, -2.184013, -6.650386, -11.339679, 9.228732, -1.404589, 2.171521, 0.896978, -6.621984, -4.545920, 13.884609, -1.123358, -0.923706, 4.767986, -9.866699, 6.330471, 4.947943, 3.653640, 0.627295, -1.418376, 7.754135, 12.791749, 4.063228, 0.038454, 0.314657, -1.634821, 6.389931, -1.488340, 3.999534, 0.215401, -0.181865, 2.196856, 4.717573, 6.869417, 2.055507, 1.418586, -0.420217, -0.110077, 14.367413, 5.553504, 0.487188, 0.378941, 1.879768, -0.202016, -10.270409, -1.961450, -11.321340, 0.441587, -1.343712, -4.230900, -11.443472, 9.435158, 4.504445, 2.374846, -1.819612, 3.333546, -1.960048, 3.945388, 7.101923, 3.236235, 2.681556, -1.916944, -1.309612, 0.481884, -6.575025, -2.207341, 2.351471, -4.432195, 2.758456, 13.669015, -0.077529, -1.750433, 0.982173, -1.102739, 1.765100, 10.197113, -4.855771, -0.660945, 2.271418, -3.565191, -14.984224, 1.439270, 3.301828, 2.108843, 0.265774, 0.718740, 0.782613, -11.013798, 9.651599, 2.876146, -2.513091, 4.115267, -2.787382, 10.835589, 0.751847, 4.723278, 0.899466, -4.321622, -12.043962, -5.724381, 7.725010, -2.925992, -5.084525, 0.332969, 1.627347, -5.202522, 14.258527, 1.462720, -1.412799, 1.460808, 7.823715, 1.634301, -13.005331, -0.980312, -1.059088, -1.545018, 1.020770, -4.252513, -8.229203, -3.390918, -1.519700, -0.946901, -4.533985, -12.035786, -8.222100, -4.992132, 1.707012, -5.485870, 6.969133, -7.731116, 1.348475, 0.726024, 0.600682, -0.035905, 6.850668, -11.871725, 7.048319, 2.871839, 0.624802, 2.038654, 4.774650, -13.444428, -5.577247, 2.015341, 1.844574, 3.290760, 6.889421, 11.469383, -7.349097, -1.339809, 2.482231, 1.115200, -5.807941, 8.738195, 11.087464, -1.776333, 5.312061, -4.992681, -9.364072, 11.897147, -1.947318, 1.959287, 2.723696, 3.059522, 10.753381, 0.834805, 10.698159, 0.180364, 4.824081, 5.234599, 0.957064, -1.955359, -13.039012, 2.320126, -1.670215, 4.514218, -3.285675, 13.441216, 5.371054, 1.895006, -1.340129, -0.712374, -1.254688, -15.359800, 0.225010, 1.504008, 0.490780, -3.379911)$

Додаток Г Опис програми "Consave Demo"

Програма "Consave Demo" реалізує кластерінг неопуклих багатогранників в кулі, циліндрі і кубоїді мінімального обсягу. Допускаються неперервні обертання і трансляції неопуклих багатогранників. Стратегія розв'язання задачі заснована на алгоритмі пошуку допустимих стартових точок і методі декомпозиції. Для пошуку локальних екстремумів задачі нелінійної оптимізації використовується IPOPT.

Програма розроблена в середовищі Visual C 6.0.

Г 1. Опис інтерфейсу програми

Після запуску діалогове вікно дослідницької програми "Consave Demo" має вигляд (Рис. Г.1):

Спершу потрібно вибрати контейнер (шар, циліндр, кубоїд), який буде використовуватись для розв'язання задачі. На рисунку Г.1 представлені 10 неопуклих багатогранників, які можуть бути обрані для розв'язання задачі оптимального кластерінгу, описаної у Розділі 3. Вибрати можна тільки по одному багатограннику кожного виду, а відповідно не більше 10 багатогранників можуть бути обрані для розв'язання однієї задачі. Перемикач «Decomposition» відповідає за використання (невикористання) алгоритму декомпозиції POLYDEC (описаної у Розділі 4) при розв'язанні задачі. Поле «Minimal allowable distance» предназначено для введення мінімально допустимої відстані між багаторанниками. Кнопка "Run" служить для запуску процесу пошуку розв'язання задачі, а кнопка "Close" – для завершення роботи програми.

Відразу після старту за замовчуванням обрані 3 неопуклих багатогранника, поле «Minimal allowable distance» має значення 0 і перемикач «Decomposition» має положення – «використовувати».

Після завершення процесу розв'язання задачі в діалоговому вікні (Рис Г.2) виводяться значення функції мети ("Obj F ..."), змінні метричні характеристики контейнера ((l, w, h) , r , (r, h)), Час пошуку локально оптимального розв'язання задачі ("Time is t")

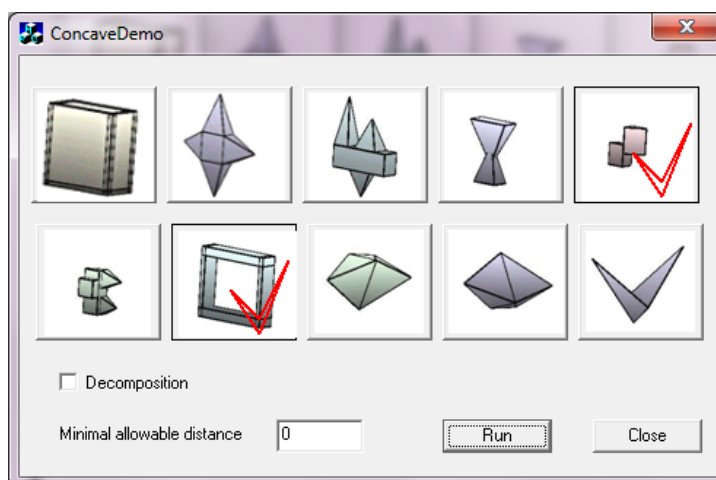


Рис. Г.1. Інтерфейс дослідницької програми Concave Demo

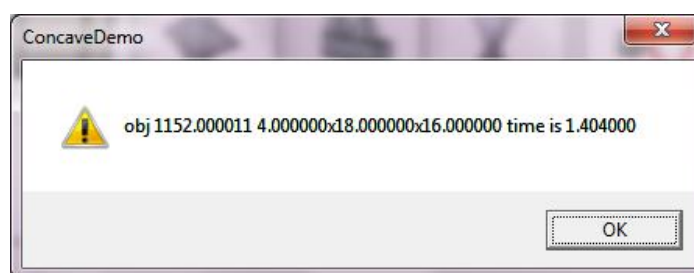


Рис. Г.2. Вікно з результатами розв'язання задачі

Після натискання кнопки «ОК», в новому діалоговому вікні виводиться малюнок, обраних неопуклих багатогранників з розрахованими параметрами розміщення, укладених в обраний контейнер (Рис. Г.3.).

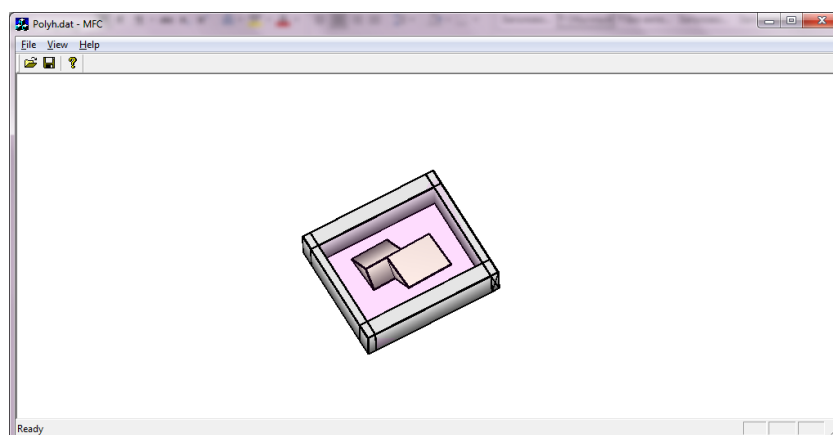


Рис. Г.3. Вікно з візуалізацією розв'язання задачі

Для збереження вихідних даних, необхідно натиснути на меню "File" і вибрати опцію "Save as ...". Після цього з'явиться діалогове вікно, в якому потрібно вибрати директорію, куди потрібно зберегти файл з вихідними даними з розширенням .dat і вибрати ім'я файлу.

На рис. Г.4. наведено розв'язання задачі для тієї ж пари багатогранників при заданих мінімально допустимих відстанях.

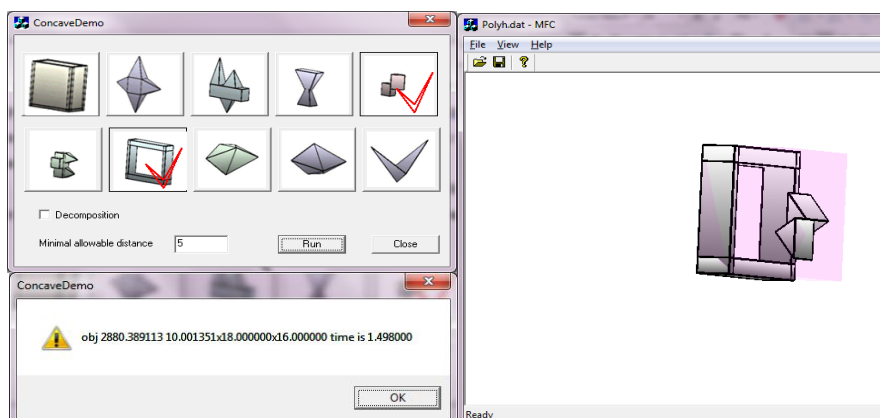


Рис. Г.4. Розв'язання задачі при заданих мінімально допустимих відстанях

Г 2. Вимоги до програми

Програма працює під управлінням операційної системи Microsoft Windows 98 / ME / 2000 / XP і вище. Для роботи програми не потрібно русифікована або налаштована на роботу з мовними установками, відповідними російській мові, операційна система.

Для роботи програми необхідна наявність бібліотеки Ipropt38.dll в тій же директорії, що і виконуваний файл програми.

Для роботи програми рекомендується наступна мінімальна конфігурація комп'ютера: IBM PC / AT- сумісний комп'ютер з процесором Pentium-166 (або вище), оперативною пам'яттю 256Мб, монітором, що підтримує колірну палітру High Color (16 біт), з роздільною здатністю 800x600 пікселів, маніпулятор "миша".

Г 3. Система нерівностей для задачі кластерингу ОРС

Як приклад розглянемо систему, що описує область допустимих розв'язків (3.20) задачі кластерингу (3.19)-(3.20) двох неопуклих багатогранників $Q_1(u_1)$ і $Q_2(u_2)$ (рис. В.5.) в мінімально охоплюючий кубоїд $\Omega = \{(x, y, z) \in R^3 : -l \leq x \leq l, -w \leq y \leq w, -h \leq z \leq h\}$. Кожний багатогранник має вигляд $Q = K_1 \cup K_2 \cup K_3 \cup K_4$, де K_1, K_2, K_3, K_4 – тетраедри з вершинами:

$$K_1: \{ p_s(x_A, y_A, z_A), s = 1, 2, 3, 4 \} = \{(0, 0, 0), (0, -4, -12), (0, 4, -12), (-4, 0, -12)\};$$

$K_2: \{ p_s(x_A, y_A, z_A), s=1, 2, 3, 4 \} = \{(0, 0, 0), (12, 0, -8), (0, -4, -12), (0, 4, -12)\};$

$K_3: \{ p_s(x_A, y_A, z_A), s=1, 2, 3, 4 \} = \{(0, 0, 0), (-8, 8, -8), (0, 4, -12), (-4, 0, -12)\};$

$K_4: \{ p_s(x_A, y_A, z_A), s=1, 2, 3, 4 \} = \{(0, 0, 0), (-8, -8, -8), (0, -4, -12), (-4, 0, -12)\};$

Число змінних задачі дорівнює $\sigma = 3 + 6 \cdot 2 + 3 \cdot 4 = 27$.

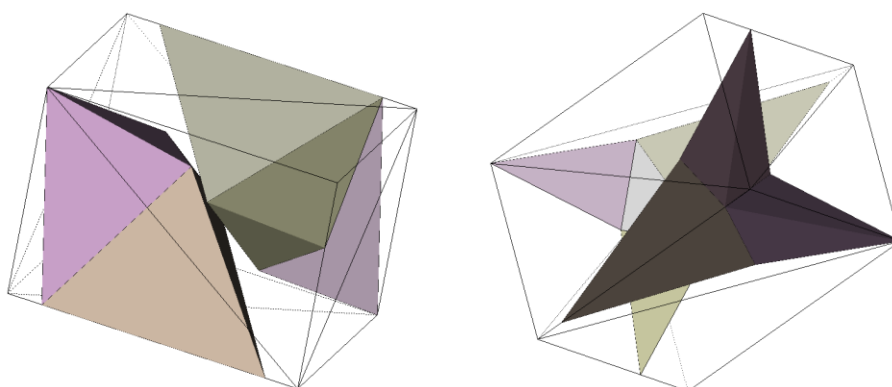


Рисунок Г.5. Неопуклі багатогранники $Q_1(u_1)$ і $Q_2(u_2)$ в кубоді

Система нерівностей, що описує область допустимих розв'язків (3.24) має наступний вигляд:

$$\begin{aligned}
 &x[3]-x[0] \geq 0 \\
 &x[4]-x[1] \geq 0 \\
 &x[5]-x[2] \geq 0 \\
 &-x[0]-x[3]+37.000000 \geq 0 \\
 &-x[1]-x[4]+37.000000 \geq 0 \\
 &-x[2]-x[5]+37.000000 \geq 0 \\
 &(4.000000 \cdot (\cos(x[6]) \cdot \sin(x[8]) + \sin(x[6]) \cdot \cos(x[8]) \cdot \cos(x[7])) - 12.000000 \cdot \sin(x[6]) \cdot \sin(x[7])) + x[3]-x[0] \geq 0 \\
 &(4.000000 \cdot (\sin(x[6]) \cdot \sin(x[8]) - \cos(x[6]) \cdot \cos(x[8]) \cdot \cos(x[7])) + 12.000000 \cdot \cos(x[6]) \cdot \sin(x[7])) + x[4]-x[1] \geq 0 \\
 &(-4.000000 \cdot \sin(x[7]) \cdot \cos(x[8]) - 12.000000 \cdot \cos(x[7])) + x[5]-x[2] \geq 0 \\
 &-x[0] - (4.000000 \cdot (\cos(x[6]) \cdot \sin(x[8]) + \sin(x[6]) \cdot \cos(x[8]) \cdot \cos(x[7])) - 12.000000 \cdot \sin(x[6]) \cdot \sin(x[7])) - x[3] + 37.000000 \geq 0 \\
 &-x[1] - (4.000000 \cdot (\sin(x[6]) \cdot \sin(x[8]) - \cos(x[6]) \cdot \cos(x[8]) \cdot \cos(x[7])) + 12.000000 \cdot \cos(x[6]) \cdot \sin(x[7])) - x[4] + 37.000000 \geq 0 \\
 &-x[2] - (-4.000000 \cdot \sin(x[7]) \cdot \cos(x[8]) - 12.000000 \cdot \cos(x[7])) - x[5] + 37.000000 \geq 0 \\
 &(-4.000000 \cdot (\cos(x[6]) \cdot \sin(x[8]) + \sin(x[6]) \cdot \cos(x[8]) \cdot \cos(x[7])) - 12.000000 \cdot \sin(x[6]) \cdot \sin(x[7])) + x[3]-x[0] \geq 0 \\
 &(12.000000 \cdot \cos(x[6]) \cdot \sin(x[7]) - 4.000000 \cdot (\sin(x[6]) \cdot \sin(x[8]) - \cos(x[6]) \cdot \cos(x[8]) \cdot \cos(x[7]))) + x[4]-x[1] \geq 0 \\
 &(4.000000 \cdot \sin(x[7]) \cdot \cos(x[8]) - 12.000000 \cdot \cos(x[7])) + x[5]-x[2] \geq 0 \\
 &-x[0] - (-4.000000 \cdot (\cos(x[6]) \cdot \sin(x[8]) + \sin(x[6]) \cdot \cos(x[8]) \cdot \cos(x[7])) - 12.000000 \cdot \sin(x[6]) \cdot \sin(x[7])) - x[3] + 37.000000 \geq 0 \\
 &-x[1] - (12.000000 \cdot \cos(x[6]) \cdot \sin(x[7]) - 4.000000 \cdot (\sin(x[6]) \cdot \sin(x[8]) - \cos(x[6]) \cdot \cos(x[8]) \cdot \cos(x[7]))) - x[4] + 37.000000 \geq 0 \\
 &-x[2] - (4.000000 \cdot \sin(x[7]) \cdot \cos(x[8]) - 12.000000 \cdot \cos(x[7])) - x[5] + 37.000000 \geq 0 \\
 &(-4.000000 \cdot (\cos(x[6]) \cdot \cos(x[8]) - \sin(x[6]) \cdot \sin(x[8]) \cdot \cos(x[7])) - 12.000000 \cdot \sin(x[6]) \cdot \sin(x[7])) + x[3]-x[0] \geq 0 \\
 &(12.000000 \cdot \cos(x[6]) \cdot \sin(x[7]) - 4.000000 \cdot (\sin(x[6]) \cdot \cos(x[8]) + \cos(x[6]) \cdot \sin(x[8]) \cdot \cos(x[7]))) + x[4]-x[1] \geq 0 \\
 &(-4.000000 \cdot \sin(x[7]) \cdot \sin(x[8]) - 12.000000 \cdot \cos(x[7])) + x[5]-x[2] \geq 0
 \end{aligned}$$

Число нерівностей в системі дорівнює 320.

Г 4. Код програми

```

#include "stdafx.h"
#include "math.h"
#include "POLIP.h"
#include "POLIPDlg.h"
#include "afxtempl.h"
#include "IpStdCInterface.h"
#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif
#define NEW_BOX_MODE
double IniDeltaX,IniDeltaY,IniDeltaZ;
#ifdef NEW_BOX_MODE
double DeltaX,DeltaY,DeltaZ;
#endif
double FixedX,FixedY,FixedZ,MinDist;
int tofix,Domain;
bool IsConvex,HaveFasets;
extern "C" __declspec(dllexport) GetInitialValues(double *x,int n);
extern "C" int __declspec(dllexport) SolveEx(double &obj,double *x,int in,int out,int m_WriteLog,int m_OptimisationWay,int m_StartingCorrection,int m_Adaptive,int m_UseHessianApproximation,int m_DerivationTest);
extern "C" int __declspec(dllexport) Solve(double &obj,double *x,int in,int out,int m_WriteLog);
FILE *Log;
double Koef;
extern "C" int __declspec(dllexport) LoadSystem(const char *FileName);
char buf[1024];
extern "C" __declspec(dllexport) void *CreateEncloseSphere();
extern "C" __declspec(dllexport) void AddPointsToEncloseSphere(void *ptr,int kol,double *points);
extern "C" __declspec(dllexport) bool GetEncloseSphere(void *ptr,double &x,double &y,double &z,double &squared_radius);
extern "C" __declspec(dllexport) void DeleteEncloseSphere(void *ptr);
struct Pair
{Pair(){type=true;}
int n1,n2,m1,m2,type;
double xyz[3];};
CArray<Pair,Pair&> Pairs;
struct Faset{CUIIntArray v;};
CString FileName="NonConvex50.con";
int KolPol;
double SizeX,SizeY,SizeZ;
struct Point
{double xyz[3];
void Shift(double x,double y ,double z);
void Shift(double *shift);
void Shift(Point &shift);
Point(){Init(0.,0.,0.);}
void Init(double a,double b,double c){xyz[0]=a;xyz[1]=a;xyz[2]=a;}
void Translate(Point *origin,double x,double y ,double z);
void Translate(Point *origin,double *translate);
void Translate(Point *origin,Point &translate);
void Copy (Point *origin);
void Move(Point *origin,double x,double y ,double z,double fi,double q,double t);
void Move(Point *origin,double *param);
void Move(Point *origin,Point &translate,double fi,double q,double t);
void Move(Point *origin,double x,double y ,double z,double sin1,double cos1,double sin2,double cos2,double sin3,double cos3);
void Move(Point *origin,Point &translate,double sin1,double cos1,double sin2,double cos2,double sin3,double cos3);
void Rotate(Point *origin,double x,double y ,double z,double fi,double q,double t);
void Rotate(Point *origin,double *param);
void Rotate(Point *origin,Point &translate,double fi,double q,double t);
void Rotate(Point *origin,double fi,double q,double t);
void Restore(Point *origin,double *param);
void Restore(Point *origin,double *translate,double sin1,double cos1,double sin2,double cos2,double sin3,double cos3);
void Restore(Point *origin,double x,double y ,double z,double sin1,double cos1,double sin2,double cos2,double sin3,double cos3);
void Load(FILE *f);};
struct cPolyhedra
{int kol,kolf;
Faset *fassets;
Point *Ini,*Curr,IniCenter,Center,Min,Max;
double Rad,DistToCenter2;
cPolyhedra(){Ini=Curr=NULL;kol=0;fassets=NULL;}
~cPolyhedra(){Finit();}
void Refresh();
void Move(double *param);
void Translate(double *param);
void Rotate(double *param);
void Rotate(double sin1,double cos1,double sin2,double cos2,double sin3,double cos3);
void Shift(double *param);
void GetMinMax();
void Load(FILE *f,bool HaveFasets=NULL);
void Finit();
double GetDistToPlane(double *plane);
double GetDistToInvetPlane(double *plane);};
struct Polyhedra
{int kol;
double Rad,Params[6],BestParams[6];
Point IniCenter,Center,Min,Max;
cPolyhedra *Pol;
Polyhedra(){Pol=NULL;kol=0;BestParams[0]=BestParams[1]=BestParams[2]=BestParams[3]=BestParams[4]=BestParams[5]=BestParams[0]=Params[1]=Params[2]=Params[3]=Params[4]=Params[5]=0.;}
~Polyhedra(){Finit();}
void Refresh();
void Move(double *param);
void Translate(double *param);
void Rotate(double *param);
void Rotate(double sin1,double cos1,double sin2,double cos2,double sin3,double cos3);
void Shift(double *param);
void GetMinMax();
void Load(FILE *f,bool HaveFasets=NULL,bool IsSingle=false);
void Finit();
void Move();
void Rotate();
void Translate();
} *Polyhedrons;
void Point::Shift(double x,double y ,double z){xyz[0]+=x;xyz[1]+=y ;xyz[2]+=z;}
void Point::Shift(double *shift){xyz[0]+=shift[0];xyz[1]+=shift[1];xyz[2]+=shift[2];}
void Point::Shift(Point &shift){xyz[0]+=shift.xyz[0];xyz[1]+=shift.xyz[1];xyz[2]+=shift.xyz[2];}
void Point::Translate(Point *origin,double x,double y ,double z){xyz[0]=origin->xyz[0]+x;xyz[1]=origin->xyz[1]+y ;xyz[2]=origin->xyz[2]+z;}
void Point::Translate(Point *origin,double *translate){xyz[0]=origin->xyz[0]+translate[0];xyz[1]=origin->xyz[1]+translate[1];xyz[2]=origin->xyz[2]+translate[2];}
void Point::Translate(Point *origin,Point &translate){xyz[0]=origin->xyz[0]+translate.xyz[0];xyz[1]=origin->xyz[1]+translate.xyz[1];xyz[2]=origin->xyz[2]+translate.xyz[2];}
void Point::Move(Point *origin,double *param){Move(origin,param[0],param[1],param[2],param[3],param[4],param[5]);}
void Point::Move(Point *origin,double x,double y ,double z,double fi,double q,double t){Move(origin,x,y ,z,sin(fi),cos(fi),sin(q),cos(q),sin(t),cos(t));}

```

```

void Point::Move(Point *origin, Point &translate, double fi, double
q, double
t){Move(origin, translate, sin(fi), cos(fi), sin(q), cos(q), sin(t), cos(t));}
void Point::Rotate(Point *origin, double fi, double q, double
t){Rotate(origin, sin(fi), cos(fi), sin(q), cos(q), sin(t), cos(t));}
void Point::Copy(Point
*origin){xy z[0]=origin->xy z[0];xy z[1]=origin->xy z[1];xy z[2]=origin->xy z[2];}
void Point::Move(Point *origin, double x, double y, double z, double
cos1, double cos2, double sin2, double cos3, double sin3, double cos3)
{if (sin1==0. && sin2==0)
    Translate(origin, x, y, z);
else
    { Rotate(origin, sin1, cos1, sin2, cos2, sin3, cos3);
      Shift(x, y, z); }}
void Point::Move(Point *origin, Point &translate, double sin1, double
cos1, double sin2, double cos2, double sin3, double cos3)
{if (sin1==0. && sin2==0)
    Translate(origin, translate);
else
    { Rotate(origin, sin1, cos1, sin2, cos2, sin3, cos3);
      Shift(translate); }}
void Point::Restore(Point *origin, double
*param){Restore(origin, param, sin(param[3]), cos(param[3]), sin(param[4]), cos(param[
4]), sin(param[5]), cos(param[5]));}
void Point::Restore(Point *origin, double *translate, double sin1, double
cos1, double sin2, double cos2, double sin3, double cos3)
{Translate(origin, -translate[0], -translate[1], -translate[2]);}
void Point::Rotate(Point *origin, double sin1, double cos1, double
sin2, double cos2, double sin3, double cos3)
{xy z[0]=origin->xy z[0]*cos1*cos3-origin->xy z[0]*sin1*cos2*sin3-origin-
>xy z[1]*cos1*sin3-origin->xy z[1]*sin1*cos2*cos3+origin->xy z[2]*sin1*sin2;
xy z[1]=origin->xy z[0]*sin1*cos3+origin->xy z[0]*cos1*cos2*sin3-origin-
>xy z[1]*sin1*sin3+origin->xy z[1]*cos1*cos2*cos3-origin->xy z[2]*cos1*sin2;
xy z[2]=origin->xy z[0]*sin2*sin3+origin->xy z[1]*sin2*cos3+origin->xy z[2]*cos2;}
void cPolyhedra::Finit()
{kol=kolf=0;
delete []f assets;
delete []Curr;
delete []Ini;
Curr=Ini=NULL;
f assets=NULL;}
void cPolyhedra::Translate(double *param)
{for(int i=0; i<kol; i++)
    Curr[i].Translate(Ini+i, param);
Center.Translate(&IniCenter, param);}
void cPolyhedra::Rotate(double sin1, double cos1, double sin2, double
cos2, double sin3, double cos3)
{for(int i=0; i<kol; i++)
    Curr[i].Rotate(Ini+i, sin1, cos1, sin2, cos2, sin3, cos3);
Center.Rotate(&IniCenter, sin1, cos1, sin2, cos2, sin3, cos3);}
void cPolyhedra::Rotate(double
*param){Rotate(sin(param[3]), cos(param[3]), sin(param[4]), cos(param[4]), sin(param[
5]), cos(param[5]));}
void cPolyhedra::Shift(double *param)
{for(int i=0; i<kol; i++)
    Curr[i].Shift(param);
Center.Shift(param);}
void cPolyhedra::Move(double *param)
{if (param[3]==0. && param[4]==0.)
    Translate(param);
else
    { Rotate(param);
      Shift(param); }}
void cPolyhedra::Refresh()
{for(int i=0; i<kol; i++)
    Curr[i].Copy(Ini+i);
Center.Copy(&IniCenter);}
void cPolyhedra::GetMinMax()
{Min.Init(999999999., 999999999., 999999999.);
Max.Init(-999999999., -999999999., -999999999.);
for(int i=0; i<kol; i++)
    { if (Min.xy z[0]>Curr[i].xyz[0])
      Min.xy z[0]=Curr[i].xyz[0];
      if (Min.xy z[1]>Curr[i].xyz[1])
        Min.xy z[1]=Curr[i].xyz[1];
      if (Min.xy z[2]>Curr[i].xyz[2])
        Min.xy z[2]=Curr[i].xyz[2];
      if (Max.xy z[0]<Curr[i].xyz[0])
        Max.xy z[0]=Curr[i].xyz[0];
      if (Max.xy z[1]<Curr[i].xyz[1])
        Max.xy z[1]=Curr[i].xyz[1];
      if (Max.xy z[2]<Curr[i].xyz[2])
        Max.xy z[2]=Curr[i].xyz[2];
      } }
double cPolyhedra::GetDistToPlane(double *plane)
{double
sin1=sin(plane[0]), sin2=sin(plane[1]), cos1=cos(plane[0]), cos2=cos(plane[1]), d, dist=9
99999999999999.;
for(int i=0; i<kol; i++)
    {d=Curr[i].xy z[0]*cos1+Curr[i].xyz[1]*sin1*cos2+Curr[i].xyz[2]*sin1*sin2;
      if (dist>d)
        dist=d; }
return dist+plane[2];}
double cPolyhedra::GetDistToInvertPlane(double *plane)
{double
sin1=sin(plane[0]), sin2=sin(plane[1]), cos1=cos(plane[0]), cos2=cos(plane[1]), d, dist=9
99999999999999.;
for(int i=0; i<kol; i++)
    {d=-Curr[i].xy z[0]*cos1-Curr[i].xyz[1]*sin1*cos2-Curr[i].xyz[2]*sin1*sin2;
      if (dist>d)
        dist=d; }
return dist-plane[2];}
void cPolyhedra::Load(FILE *f, bool HaveFassets)
{int i, j, k, n;
Finit();
fscanf(f, "%d", &kol);
Ini=new Point[kol];
Curr=new Point[kol];
void *sph=CreateEncloseSphere();
for(i=0; i<kol; i++)
    { Ini[i].Load(f);
      AddPointsToEncloseSphere(sph, 1, Ini[i].xy z); }
GetEncloseSphere(sph, IniCenter.xy z[0], IniCenter.xy z[1], IniCenter.xy z[2], Rad);
DistToCenter2=IniCenter.xy z[0]*IniCenter.xy z[0]+IniCenter.xy z[1]*IniCenter.xy z[1]+
IniCenter.xy z[2]*IniCenter.xy z[2];
DeleteEncloseSphere(sph);
Rad=sqrt(Rad);
Refresh();
if (HaveFassets==false)
    return;
fscanf(f, "%d", &kolf);
//TRACE("%d %d\n", kol, kolf);
f assets=new Faset[kolf];
for(i=0; i<kolf; i++)
    { fscanf(f, "%d", &n);
      for(j=0; j<n; j++)
        { fscanf(f, "%d", &k);
          f assets[i].v.Add(k); } } }
void Polyhedra::Finit()
{kol=0;
delete []Pol;
Pol=NULL;
Params[0]=Params[1]=Params[2]=Params[3]=Params[4]=Params[5]=0.;}
void Polyhedra::Refresh()
{for(int i=0; i<kol; i++)
    Pol[i].Refresh();
Center.Copy(&IniCenter);
Params[0]=Params[1]=Params[2]=Params[3]=Params[4]=Params[5]=0.;}
void Polyhedra::Move(double *param)
{for(int i=0; i<kol; i++)
    Pol[i].Move(param);
Center.Move(&IniCenter, param);
Params[0]=param[0];
Params[1]=param[1];
Params[2]=param[2];
Params[3]=param[3];
Params[4]=param[4];
Params[5]=param[5];}
void Polyhedra::Move()
{for(int i=0; i<kol; i++)
    Pol[i].Move(Params);
Center.Move(&IniCenter, Params);}
void Polyhedra::Translate()
{for(int i=0; i<kol; i++)
    Pol[i].Translate(Params);
Center.Translate(&IniCenter, Params);}
void Polyhedra::Translate(double *param)
{for(int i=0; i<kol; i++)
    Pol[i].Translate(param);
Center.Translate(&IniCenter, param);
Params[0]=param[0];
Params[1]=param[1];
Params[2]=param[2];}

```

```

void Polyhedra::Rotate(double *param)
{Rotate(sin(param[3]),cos(param[3]),sin(param[4]),cos(param[4]),sin(param[5]),cos(p
aram[5]));
Params[3]=param[3];
Params[4]=param[4];
Params[5]=param[5];}
void
Polyhedra::Rotate(){Rotate(sin(Params[3]),cos(Params[3]),sin(Params[4]),cos(Pa
ms[4]),sin(Params[5]),cos(Params[5]));}
void Polyhedra::Rotate(double sin1,double cos1,double sin2,double
cos2,double sin3,double cos3)
{for(int i=0;i<kol;i++)
    Pol[i].Rotate(sin1,cos1,sin2,cos2,sin3,cos3);
Center.Rotate(&IniCenter,sin1,cos1,sin2,cos2,sin3,cos3);}
void Polyhedra::Shift(double *param)
{for(int i=0;i<kol;i++)
    Pol[i].Shift(param);
Center.Shift(param);
Params[0]+=param[0];
Params[1]+=param[1];
Params[2]+=param[2];}
void Polyhedra::GetMinMax()
{Min.Init(999999999.,999999999.,999999999.);
Max.Init(-999999999.,-999999999.,-999999999.);
double *min,*max;
for(int i=0;i<kol;i++)
    { Pol[i].GetMinMax();
    min=Pol[i].Min.xyz;
    max=Pol[i].Max.xyz;
    if (Min.xyz[0]>min[0])
        Min.xyz[0]=min[0];
    if (Min.xyz[1]>min[1])
        Min.xyz[1]=min[1];
    if (Min.xyz[2]>min[2])
        Min.xyz[2]=min[2];
    if (Max.xyz[0]<max[0])
        Max.xyz[0]=max[0];
    if (Max.xyz[1]<max[1])
        Max.xyz[1]=max[1];
    if (Max.xyz[2]<max[2])
        Max.xyz[2]=max[2]; } }
void Polyhedra::Load(FILE *f,bool Hav eFasets,bool IsSingle)
{int i,j;
Finit();
if (IsSingle)
    kol=1;
else
    fscanf(f,"%d",&kol);
Pol=new cPolyhedra[kol];
void *sph=CreateEncloseSphere();
for(i=0;i<kol;i++)
    { Pol[i].Load(f,Hav eFasets);
    for(j=0;j<Pol[i].kol;j++)
        AddPointsToEncloseSphere(sph,1,Pol[i].Ini[j].xyz); }
GetEncloseSphere(sph,IniCenter.xyz[0],IniCenter.xyz[1],IniCenter.xyz[2],Rad);
DeleteEncloseSphere(sph);
Rad=sqrt(Rad);}
bool LoadObjects(const char *FileName)
{FILE *f=fopen(FileName,"r");
if (f==NULL)
    return false;
delete []Polyhedrons;
fscanf(f,"%d",&KolPol);
Polyhedrons=new Polyhedra[KolPol];
for(int i=0;i<KolPol;i++)
    Polyhedrons[i].Load(f,Hav eFasets,IsConvex);
fclose(f);
return true;}
bool LoadProblem(const char *f name,double &Size_X,double &Size_Y,double
&Size_Z,bool IsConvex=true)
{int i;
FILE *f=fopen(f name,"r");
if (f==NULL)
    return false;
delete []Polyhedrons;
fscanf(f,"%d",&KolPol);
Polyhedrons=new Polyhedra[KolPol];
for(i=0;i<KolPol;i++)
    Polyhedrons[i].Load(f,Hav eFasets,IsConvex);
fclose(f);
return true;}

bool LoadProblem2(const char *f name,double &Size_X,double &Size_Y,double
&Size_Z,bool IsConvex=true,bool Hav eFasets=false)
{int i;
FILE *f=fopen(f name,"r");
if (f==NULL)
    return false;
delete []Polyhedrons;
fscanf(f,"%d",&KolPol);
Polyhedrons=new Polyhedra[KolPol];
for(i=0;i<KolPol;i++)
    Polyhedrons[i].Load(f,Hav eFasets,IsConvex);
fclose(f);
return true;}
class CAboutDlg : public CDialog
{public:
    CAboutDlg();
protected:
    virtual void DoDataExchange(CDataExchange* pDX); // DDX/DDV
support
    DECLARE_MESSAGE_MAP();
CAboutDlg::CAboutDlg() : CDialog(CAboutDlg::IDD)
void CAboutDlg::DoDataExchange(CDataExchange* pDX)
{ CDialog::DoDataExchange(pDX);}
BEGIN_MESSAGE_MAP(CAboutDlg, CDialog)
END_MESSAGE_MAP()
CPOLIPDLg::CPOLIPDLg(CWnd* pParent )
: CDialog(CPOLIPDLg::IDD, pParent)
{ m_SizeX = 37.0;
m_SizeY = 37.0;
m_SizeZ = 37.0;
m_IsRandom = TRUE;
m_IniRandom = 0;
m_Delta = 3.0;
m_KolSteps = 1;
m_LoadMode = 0;
m_FixedX = 0.0;
m_FixedY = 0.0;
m_FixedZ = 0.0;
m_MinDist = 0.0;
m_IsConvex = FALSE;
m_Hav eFasets = FALSE;
m_Domain = 0;
m_Restart = FALSE;
m_hIcon = AfxGetApp()->LoadIcon(IDR_MAINFRAME);}
void CPOLIPDLg::DoDataExchange(CDataExchange* pDX)
{ CDialog::DoDataExchange(pDX);
DDX_Text(pDX, IDC_SIZEX, m_SizeX);
DDX_Text(pDX, IDC_SIZEY, m_SizeY);
DDX_Text(pDX, IDC_SIZEZ, m_SizeZ);
DDX_Check(pDX, IDC_IS_RANDOM, m_IsRandom);
DDX_Text(pDX, IDC_INIRANDOM, m_IniRandom);
DDX_Text(pDX, IDC_DELTA, m_Delta);
DDX_Text(pDX, IDC_KOSTEPS, m_KolSteps);
DDX_Radio(pDX, IDC_LOADMODE, m_LoadMode);
DDX_Text(pDX, IDC_SIZEX2, m_FixedX);
DDX_Text(pDX, IDC_SIZEY2, m_FixedY);
DDX_Text(pDX, IDC_SIZEZ2, m_FixedZ);
DDX_Text(pDX, IDC_MINDIST, m_MinDist);
DDX_Check(pDX, IDC_CONVEX, m_IsConvex);
DDX_Check(pDX, IDC_FASSETS, m_Hav eFasets);
DDX_Radio(pDX, IDC_DOMAIN1, m_Domain);
DDX_Check(pDX, IDC_RESTART, m_Restart);}
BEGIN_MESSAGE_MAP(CPOLIPDLg, CDialog)
ON_WM_SYSCOMMAND()
ON_WM_PAINT()
ON_WM_QUERYDRAGICON()
ON_BN_CLICKED(IDC_LOAD, OnLoad)
END_MESSAGE_MAP()
BOOL CPOLIPDLg::OnInitDialog()
{ CDialog::OnInitDialog();
ASSERT((IDM_ABOUTBOX & 0xFFF0) == IDM_ABOUTBOX);
ASSERT(IDM_ABOUTBOX < 0xF000);
CMenu* pSysMenu = GetSystemMenu(FALSE);
if (pSysMenu != NULL)
    { CString strAboutMenu;
    strAboutMenu.LoadString(IDS_ABOUTBOX);
    if (!strAboutMenu.IsEmpty())
        { pSysMenu->AppendMenu(MF_SEPARATOR);
        pSysMenu->AppendMenu(MF_STRING, IDM_ABOUTBOX, strAboutMenu); } }
SetIcon(m_hIcon, TRUE); // Set big icon
SetIcon(m_hIcon, FALSE); // Set small icon
}

```



```

return TRUE; // return TRUE unless you set the focus to a control)
void CPOLIPDlg::OnSysCommand(UINT nID, LPARAM lParam)
{
    if ((nID & 0xFFFF) == IDM_ABOUTBOX)
    {
        CAboutDlg dlgAbout;
        dlgAbout.DoModal();
    }
    else
    {
        CDialog::OnSysCommand(nID, lParam);
    }
}
void CPOLIPDlg::OnPaint()
{
    if (!IsIconic())
    {
        CPaintDC dc(this); // device context for painting
        SendMessage(WM_ICONERASEBKGD, (WPARAM) dc.GetSafeHdc(), 0);
        int cxIcon = GetSystemMetrics(SM_CXICON);
        int cyIcon = GetSystemMetrics(SM_CYICON);
        CRect rect;
        GetClientRect(&rect);
        int x = (rect.Width() - cxIcon + 1) / 2;
        int y = (rect.Height() - cyIcon + 1) / 2;
        dc.DrawIcon(x, y, m_hIcon);
    }
    else
    {
        CDialog::OnPaint();
    }
}
HCURSOR CPOLIPDlg::OnQueryDragIcon()
{
    return (HCURSOR) m_hIcon;
}
bool ReadPoint()
{
    return true;
}
double GenerateRandomNumber(double a, double b)
{
    double d = (b - a) / 32768.000030519;
    if (d <= 0.)
        return 0.;
    return a + d * (rand() + rand() / 32767. + rand() / 1073676289.);
}
bool GeneratePoint()
{
    int i, j, k, i1, k1;
    double dx, dy, dz, r;
    for (i = 0; i < KolPol; i++)
    {
        for (j = 0; j < 10000; j++)
        {
            Polyhedrons[i].Params[0] = Polyhedrons[i].Params[1] = Polyhedrons[i].Params[2] = 0.;
            Polyhedrons[i].Params[3] = GenerateRandomNumber(-3.1415926535898, 3.1415926535898);
            Polyhedrons[i].Params[4] = GenerateRandomNumber(-3.1415926535898, 3.1415926535898);
            Polyhedrons[i].Params[5] = GenerateRandomNumber(-3.1415926535898, 3.1415926535898);
            Polyhedrons[i].Move();
            Polyhedrons[i].GetMinMax();
            switch (Domain)
            {
                case 0:
                    dx = Polyhedrons[i].Max.xyz[0] - Polyhedrons[i].Min.xyz[0];
                    dy = Polyhedrons[i].Max.xyz[1] - Polyhedrons[i].Min.xyz[1];
                    dz = Polyhedrons[i].Max.xyz[2] - Polyhedrons[i].Min.xyz[2];
                    if (SizeX <= dx || SizeY <= dy || SizeZ <= dz)
                        continue;
                    Polyhedrons[i].Params[0] = GenerateRandomNumber(0., SizeX - dx) - Polyhedrons[i].Min.xyz[0];
                    Polyhedrons[i].Params[1] = GenerateRandomNumber(0., SizeY - dy) - Polyhedrons[i].Min.xyz[1];
                    Polyhedrons[i].Params[2] = GenerateRandomNumber(0., SizeZ - dz) - Polyhedrons[i].Min.xyz[2];
                    break;
                case 1:
                    r = GenerateRandomNumber(0., SizeX - Polyhedrons[i].Rad);
                    dx = GenerateRandomNumber(-3.1415926535898, 3.1415926535898);
                    dy = GenerateRandomNumber(-3.1415926535898, 3.1415926535898);
                    Polyhedrons[i].Params[0] = r * cos(dx) - Polyhedrons[i].Center.xyz[0];
                    Polyhedrons[i].Params[1] = r * sin(dx) * cos(dy) - Polyhedrons[i].Center.xyz[1];
                    Polyhedrons[i].Params[2] = r * sin(dx) * sin(dy) - Polyhedrons[i].Center.xyz[2];
                    break;
                case 2:
                    r = GenerateRandomNumber(0., SizeX - Polyhedrons[i].Rad);
                    dx = GenerateRandomNumber(-3.1415926535898, 3.1415926535898);
                    Polyhedrons[i].Params[0] = r * cos(dx) - Polyhedrons[i].Center.xyz[0];
                    Polyhedrons[i].Params[1] = r * sin(dx) - Polyhedrons[i].Center.xyz[1];
                    Polyhedrons[i].Params[2] = GenerateRandomNumber(-0.5 * SizeZ + Polyhedrons[i].Rad, 0.5 * SizeZ - Polyhedrons[i].Rad) - Polyhedrons[i].Center.xyz[2];
                    break;
            }
            Polyhedrons[i].Move();
            Polyhedrons[i].GetMinMax();
            for (k = 0; k < i; k++)
            {
                for (i1 = 0; i1 < Polyhedrons[i].kol; i1++)
                {
                    for (k1 = 0; k1 < Polyhedrons[k].kol; k1++)
                    {
                        dx = Polyhedrons[i].Pol[i1].Center.xyz[0] - Polyhedrons[k].Pol[k1].Center.xyz[0];
                        dy = Polyhedrons[i].Pol[i1].Center.xyz[1] - Polyhedrons[k].Pol[k1].Center.xyz[1];
                        dz = Polyhedrons[i].Pol[i1].Center.xyz[2] - Polyhedrons[k].Pol[k1].Center.xyz[2];
                        r = Polyhedrons[i].Pol[i1].Rad + Polyhedrons[k].Pol[k1].Rad + MinDist;
                        if (dx * dx + dy * dy + dz * dz < r * r)
                            break;
                    }
                    if (k1 < Polyhedrons[k].kol)
                        break;
                }
                if (i1 < Polyhedrons[i].kol)
                    break;
            }
            if (k == i)
                break;
            if (j == 10000)
                return false;
        }
    }
    // return ret <= -1. ? true : false;
    return true;
}
bool FindPlane(cPolyhedra *p1, cPolyhedra *p2, double *v ar)
{
    int j;
    Point *p;
    double
        *c1 = p1->Center.xyz, *c2 = p2->Center.xyz, dt[3] = {c1[0] - c2[0], c1[1] - c2[1], c1[2] - c2[2]}, d = sqrt(dt[0] * dt[0] + dt[1] * dt[1] + dt[2] * dt[2]), a1, a2, ret, d1, sin1;
    dt[0] /= d;
    dt[1] /= d;
    dt[2] /= d;
    if (dt[0] > 1.)
        dt[0] = 1.;
    else if (dt[0] < -1.)
        dt[0] = -1.;
    a1 = acos(dt[0]);
    sin1 = sin(a1);
    if (a1)
    {
        a2 = dt[2] / sin1;
        if (a2 > 1.)
            a2 = 1.;
        else if (a2 < -1.)
            a2 = -1.;
        a2 = asin(a2);
    }
    else
        a2 = 0.;
    d = dt[1] * sin1 * cos(a2);
    d1 = dt[2] * sin1 * sin(a2);
    if (d1 < 0. && d1 < 0.) a1 = -a1;
    else if (d1 < 0.) a2 = -a2;
    else if (d < 0.) {a1 = -a1; a2 = -a2;}
    FILE *f = fopen("PlaneSystem.dat", "w");
    fprintf(f, "$P_p3:={c[0]+i}s[0]c[1]+j}s[0]s[1]n");
    fprintf(f, "$P_p3:={c[0][3]+i}s[0]c[1][3]+j}s[0]s[1][3]n");
    fprintf(f, "#D:=$P_p(%1.12lf, %1.12lf, %1.12lf)+[2]n", c1[0], c1[1], c1[2]);
    for (j = 0, p = p1 ->Curr; j < p1 ->kol; j++)
        fprintf(f, "#D+$P_p3(%1.12lf, %1.12lf, %1.12lf)n", p[j].xyz[0] - c1[0], p[j].xyz[1] - c1[1], p[j].xyz[2] - c1[2]);
    fprintf(f, "#D:=$P_p(%1.12lf, %1.12lf, %1.12lf)-[2]n", -c2[0], -c2[1], -c2[2]);
    for (j = 0, p = p2 ->Curr; j < p2 ->kol; j++)
        fprintf(f, "#D+$P_p3(%1.12lf, %1.12lf, %1.12lf)n", c2[0] - p[j].xyz[0], c2[1] - p[j].xyz[1], c2[2] - p[j].xyz[2]);
    if (a2)
        fprintf(f, "@-3]:[3]:=0:[0]:=%1.12lf:[1]:=%1.12lf:[2]:=%1.12lf n", a1, a2, -0.5 * (dt[0] * (c1[0] + c2[0]) + dt[1] * (c1[1] + c2[1]) + dt[2] * (c1[2] + c2[2])));
    else
        fprintf(f, "@-3]:[3]:=0:[0]:=%1.12lf:[1]:=%1.12lf:[2]:=%1.12lf n", a1, 0.001, -0.5 * (dt[0] * (c1[0] + c2[0]) + dt[1] * (c1[1] + c2[1]) + dt[2] * (c1[2] + c2[2])));
    fflush(f);
    fclose(f);
    if (j = LoadSystem("PlaneSystem.dat") == false)
    {
        AfMessageBox("Wrong plane file");
        return false;
    }
    SolveEx(ret, v ar, 0, 3, 0, 0, 0, 0);
    fflush(Log);
    return true;
}
bool GenerateSSystem(double delta, int tof ix)
{
    int i, j, k1, k2, l, i1, j1, m = 6 * KolPol + 3, l, mm = 0;
    double
        v ar[3], dx, dy, dz, r, r1, d, delta3 = delta * 1.3660254037844, delta2 = delta * 1.4142135623731, Koeff2 = Koeff * Koeff;
    Point *pt;
    FILE *f = fopen("PolySystem.dat", "w");
    fprintf(f, "$cc=c[0]c[0+2]n");
    fprintf(f, "$cs=c[0]s[0+2]n");
    fprintf(f, "$sc=s[0]c[0+2]n");
    fprintf(f, "$ss=s[0]s[0+2]n");
    fprintf(f, "$k11:=#cc-c[0+1]#ssn");
    fprintf(f, "$k12:=#cs+c[0+1]#scn");
}

```

```

f printf (f, "$k13:=s{0}s{[0+1]}\n");
f printf (f, "$k21:=#sc+c{[0+1]}#cs\n");
f printf (f, "$k22:=#ss-c{[0+1]}#cc\n");
f printf (f, "$k23:=c{0}s{[0+1]}\n");
f printf (f, "$k31:=s{[0+1]}s{[0+2]}\n");
f printf (f, "$k32:=s{[0+1]}c{[0+2]}\n");
f printf (f, "$k33:=c{[0+1]}\n");
f printf (f, "$SetMatrix:=#cc:=#cc(0)|#cs:=#cs(0)|#sc:=#sc(0)|#ss:=#ss(0)|#k11:=
$K11(0)|#k12:=#k12(0)|#k13:=#k13(0)|#k21:=#k21(0)|#k22:=#k22(0)|#k23:=
k23(0)|#k31:=#k31(0)|#k32:=#k32(0)|#k33:=#k33(0)\n");
f printf (f, "$RotX:={#k11-}{#k12-}{#k13-}\n");
f printf (f, "$RotY:={#k21-}{#k22-}{#k23-}\n");
f printf (f, "$RotZ:={#k31-}{#k32-}{#k33-}\n");
f printf (f, "$RotPoint:={#x(0)_1}_{#y(0)_1}_{#z(0)_1}=#RotX({3},{4},{5})|#{y(0)_1}_{#z(0)_1}=#RotY({3},{4},{5})|
#{z(0)_1}_{#z(0)_1}=#RotZ({3},{4},{5})\n");
f printf (f, "$RotCenter:={#x(0)_1}_{#y(0)_1}_{#z(0)_1}=#RotX({2},{3},{4})+{5}|#{y(0)_1}_{#z(0)_1}=#RotY({2},{3},{4})
+{5}|#{z(0)_1}_{#z(0)_1}=#RotZ({2},{3},{4})+{5}\n");
for(i=0;i<KolPol;i++)
{
i1=i*6+3;
f printf (f, "%d:=%1.12lf;%d:=%1.12lf;%d:=%1.12lf\n",i1+3,Polyhedrons[i].Params[3],i
1+4,Polyhedrons[i].Params[4],i1+5,Polyhedrons[i].Params[5]);
}
switch(Domain)
{
case 1:
f printf (f, "$Delta2:=#Delta2:=[0]^2-[0]^2-[[0+1]^2-[[0+2]^2\n");
2[[3+2]]#z(0)_1}_{#z(0)_1}=#Delta2(2,2)\n");
f printf (f, "$SinS:=[0]^2-#x(0)_1}_{#y(0)_1}_{#z(0)_1}=#SinS(2,2)\n");
break;
case 2:
f printf (f, "$Delta2:=#Delta2:=[0]^2-[0]^2-[[0+1]^2\n",Koeff 2);
f printf (f, "$SinC:=[0]^2-#x(0)_1}_{#y(0)_1}_{#z(0)_1}=#
SinC(2,2)\n",Koeff 2,2.*Koeff);
}
switch(Domain)
{
case 0:
#if def NEW_BOX_MODE
if (IniDeltaX==0&&(tof ix&1))
f printf (f, "[0]=0\n");
if (IniDeltaY==0&&(tof ix&2))
f printf (f, "[1]=%lf\n",IniDeltaY);
if (IniDeltaZ==0&&(tof ix&4))
f printf (f, "[2]=%lf\n",IniDeltaZ);
if (delta>0)
{
if ((tof ix&1)==0&&FixedX==0.)
f printf (f, "[0]<%lf\n",DeltaX+0.5*delta);
if ((tof ix&2)==0&&FixedY==0.)
f printf (f, "[1]<%lf\n",DeltaY+0.5*delta);
if ((tof ix&4)==0&&FixedZ==0.)
f printf (f, "[2]<%lf\n",DeltaZ+0.5*delta);
}
f printf (f, "#SizeX:=%1.12lf-2[0];#SizeY:=%1.12lf-2[1];#SizeZ:=%1.12lf-
2[2]\n",SizeX+2.*IniDeltaX,SizeY+2.*IniDeltaY,SizeZ+2.*IniDeltaZ);
f printf (f, "[0]:=%lf;[1]:=%lf;[2]:=%lf;@#SizeX#SizeY#SizeZ\n",DeltaX,DeltaY,DeltaZ);
#else
if (tof ix&1)
f printf (f, "[0]=%lf\n",SizeX-IniDeltaX);
if (tof ix&2)
f printf (f, "[1]=%lf\n",SizeY-IniDeltaY);
if (tof ix&4)
f printf (f, "[2]=%lf\n",SizeZ-IniDeltaZ);
if (delta>0.)
f printf (f, "[0]>%lf;[1]>%lf;[2]>%lf\n",SizeX-0.5*delta,SizeY-0.5*delta,SizeZ-0.5*delta);
f printf (f, "[0]:=%lf;[1]:=%lf;[2]:=%lf;@[0][1][2]\n",SizeX,SizeY,SizeZ);
#endif
break;
case 1:
if (delta)
f printf (f, "[0]>%lf\n",SizeX-0.5*delta);
f printf (f, "[0][0]:=%lf;[1]:=1;[2]:=1;[1]=1;[2]=1;@[0]\n",SizeX);
break;
case 2:
if (delta)
f printf (f, "[0]>%lf\n",0.5*SizeZ-0.5*delta);
f printf (f, "[0]:=%lf;[1]:=%lf;[2]:=%lf;@[0]\n",0.5*SizeZ,0.5*Koeff*SizeZ,
0.5*Koeff*SizeZ,0.5*Koeff*SizeZ);
break;
}
for(i=0;i<KolPol;i++)
{
i1=i*6+3;
f printf (f, "$SetMatrix(%d)\n",i1+3);
f printf (f, "%d:=%1.12lf;%d:=%1.12lf;%d:=%1.12lf;%d:=%1.12lf;%d:=%1.12lf\n",i1,
Polyhedrons[i].Params[0],i1+1,Polyhedrons[i].Params[1],i1+2,Polyh
edrons[i].Params[2],i1+3,Polyhedrons[i].Params[3],i1+4,Polyhedrons[i].Params[4],i1
+5,Polyhedrons[i].Params[5]);
switch(Domain)
{
case 1: f printf (f, "$Delta2(%d)\n",i1); break;
case 2: f printf (f, "$Delta2(%d)\n",i1); break;
}
for(j=0;j<Polyhedrons[i].kol;j++)
{f printf (f, "$RotCenter(%d,%d,%d,%d,%d)\n",i,j,Polyhedrons[i].Pol[j].IniCenter.xy z
0],Polyhedrons[i].Pol[j].IniCenter.xyz z1],Polyhedrons[i].Pol[j].IniCenter.xyz z2],i1);
if (delta>0)
{
pt=&Polyhedrons[i].Pol[j].Center;
f printf (f, "#x(%d)_%d<%1.12lf;#y(%d)_%d<%1.12lf\n",i,j,pt->xyz[0]+delta*0.5,i,j,pt-
>xyz[0]-delta*0.5);
f printf (f, "#y(%d)_%d<%1.12lf;#z(%d)_%d<%1.12lf\n",i,j,pt->xyz[1]+delta*0.5,i,j,pt-
>xyz[1]-delta*0.5);
f printf (f, "#z(%d)_%d<%1.12lf;#z(%d)_%d<%1.12lf\n",i,j,pt->xyz[2]+delta*0.5,i,j,pt-
>xyz[2]-delta*0.5);
switch(Domain)
{
case 1:
d=pt->xyz[0]*pt->xyz[0]+pt->xyz[1]*pt->xyz[1]+pt->xyz[2]*pt->xyz[2];
r=SizeX*Polyhedrons[i].Pol[j].Rad-delta3;
if (r<0.0||d>r)
f printf (f, "$SinS(%d,%d,%1.12lf)\n",i,j,Polyhedrons[i].Pol[j].Rad);
break;
case 2:
if (pt->xyz[2]+delta+Polyhedrons[i].Pol[j].Rad>0.5*SizeZ)
f printf (f, "[0]-#z(%d)_%d-%lf\n",i,j,Polyhedrons[i].Pol[j].Rad);
if (pt->xyz[2]-delta-Polyhedrons[i].Pol[j].Rad<-0.5*SizeZ)
f printf (f, "#z(%d)_%d+0-%lf\n",i,j,Polyhedrons[i].Pol[j].Rad);
d=pt->xyz[0]*pt->xyz[0]+pt->xyz[1]*pt->xyz[1];
r=0.5*Koeff*SizeZ-Polyhedrons[i].Pol[j].Rad-delta2;
if (r<0.0||d>r)
f printf (f, "$SinC(%d,%d,%1.12lf)\n",i,j,SizeX*Polyhedrons[i].Pol[j].Rad);
f printf (f, "$SinC(%d,%d,%1.12lf)\n",i,j,Polyhedrons[i].Pol[j].Rad);
break;
case 0:
#if def NEW_BOX_MODE
r=Polyhedrons[i].Pol[j].Rad+delta;
if (pt->xyz[0]<r+DeltaX+IniDeltaX)
f printf (f, "#x(%d)_%d-[0]-%lf\n",i,j,Polyhedrons[i].Pol[j].Rad);
if (pt->xyz[1]<r+DeltaY+IniDeltaY)
f printf (f, "#y(%d)_%d-[1]-%lf\n",i,j,Polyhedrons[i].Pol[j].Rad);
if (pt->xyz[2]<r+DeltaZ+IniDeltaZ, Polyhedrons[i].Pol[j].Rad)
f printf (f, "#z(%d)_%d-[2]-%lf\n",i,j,Polyhedrons[i].Pol[j].Rad);
if (SizeX-pt->xyz[0]<r+DeltaX+2*IniDeltaX)
f printf (f, "%1.12lf-#x(%d)_%d-[0]-%lf\n",SizeX+2.*IniDeltaX,i,j,Polyhedrons[i].Pol[j].Rad);
if (SizeY-pt->xyz[1]<r+DeltaY+2*IniDeltaX)
f printf (f, "%1.12lf-#y(%d)_%d-[1]-%lf\n",SizeY+2.*IniDeltaY,i,j,Polyhedrons[i].Pol[j].Rad);
if (SizeZ-pt->xyz[2]<r+DeltaZ+2*IniDeltaY)
f printf (f, "%1.12lf-#z(%d)_%d-[2]-%lf\n",SizeZ+2.*IniDeltaZ,i,j,Polyhedrons[i].Pol[j].Rad);
}else
r=Polyhedrons[i].Pol[j].Rad;
if (pt->xyz[0]<delta*0.5+r+IniDeltaX)
f printf (f, "#x(%d)_%d+[%d]\n",i,j,i1);
if (pt->xyz[1]<delta*0.5+r+IniDeltaY)
f printf (f, "#y(%d)_%d+[%d]\n",i,j,i1+1);
if (pt->xyz[2]<delta*0.5+r+IniDeltaZ)
f printf (f, "#z(%d)_%d+[%d]\n",i,j,i1+2);
if (SizeX-pt->xyz[0]<delta+r)
f printf (f, "[0]-#x(%d)_%d-[%d]\n",i,j,i1);
if (SizeY-pt->xyz[1]<delta+r)
f printf (f, "[1]-#y(%d)_%d-[%d]\n",i,j,i1+1);
if (SizeZ-pt->xyz[2]<delta+r)
f printf (f, "[2]-#z(%d)_%d-[%d]\n",i,j,i1+2);
#endif
break;
}
}
else
switch(Domain)
{
case 0:
#if def NEW_BOX_MODE
f printf (f, "for(%d);>,0,1)
#x(%d)_%d_>>>[%d]-[0];#y(%d)_%d_>>>[%d]-[1];#z(%d)_%d_>>>[%d]-[2];%1.12lf-[0]-
#x(%d)_%d_>>>[%d];%1.12lf-[1]-#y(%d)_%d_>>>[%d];%1.12lf-[2]-#z(%d)_%d_>>>
[%d]\n",Polyhedrons[i].Pol[j].kol,i,j,i1,i1+1,i1+1+2,SizeX,i,j,i1,SizeY,i,j,i1+1,SizeZ,i,j,
i1+2);
#else
for(l=0,pt=&Polyhedrons[i].Pol[j].Ini; l<Polyhedrons[i].Pol[j].kol;l++)
{f printf (f, "#x(%d)_%d_%d+[%d];#y(%d)_%d_%d+[%d];#z(%d)_%d_%d+[%d]\n",i,j,l,i1,i,j,l,
i1+1,i,j,l,i1+2);
f printf (f, "[0]-#x(%d)_%d_%d-[%d];[1]-#y(%d)_%d_%d-[%d];[2]-#z(%d)_%d_%d-
[%d]\n",i,j,l,i1,i,j,l,i1+1,i,j,l,i1+2);
}
#endif
}
}
}

```



```

        fprintf(f, "#D:=$Pm(%d,%d)-%f\n",j1,m,MinDist*0.5);
        fprintf(f, "for(%d;>,0,1)
#D+$P_m(%d,%d,>, %d)\n", Poly hedrons[j].Pol[k2].kol,j,k2,m);
        m+=3;
    }
    else
    {
        mm-=3;
fprintf(f, "[%d]:=%1.12f;[%d]:=%1.12f;[%d]:=%1.12f;\n",mm,var[0],mm+1,var[1],mm
+2,var[2]);
        fprintf(f, "#D:=$Pp(%d,%d)-%f\n",i1,mm,MinDist*0.5);
        fprintf(f, "for(%d;>,0,1)
#D+$P_p(%d,%d,>, %d)\n", Poly hedrons[i].Pol[k1].kol,i,k1,mm);
//for(l=0,pt=Poly hedrons[i].Pol[k1].Ini;<Poly hedrons[i].Pol[k1].kol;l++)
        fprintf(f, "#D:=$Pm(%d,%d)-%f\n",j1,mm,MinDist*0.5);
        fprintf(f, "for(%d;>,0,1)
#D+$P_m(%d,%d,>, %d)\n", Poly hedrons[j].Pol[k2].kol,j,k2,mm); }
if (Domain==0)
{#if def NEW_BOX_MODE
if (tof ix&1)
    fprintf(f, "[0]=%f\n",IniDeltaX);
if (tof ix&2)
    fprintf(f, "[1]=%f\n",IniDeltaY);
if (tof ix&4)
    fprintf(f, "[2]=%f\n",IniDeltaZ);
if (FixedX)
    if (delta>0)
        {
            d=0.5*(SizeX-FixedX)-DeltaX;
            if (d>0.5*delta)
                fprintf(f, "[0]=%f\n",DeltaX+0.5*delta);
            else
                fprintf(f, "[0]=%f\n",DeltaX+d);
        }
    else
        fprintf(f, "[0]=%f\n",0.5*(SizeX-FixedX));
if (FixedY)
    if (delta>0)
        {
            d=0.5*(SizeY-FixedY)-DeltaY;
            if (d>0.5*delta)
                fprintf(f, "[1]=%f\n",DeltaY+0.5*delta);
            else
                fprintf(f, "[1]=%f\n",DeltaY+d);
        }
    else
        fprintf(f, "[1]=%f\n",0.5*(SizeY-FixedY));
if (FixedZ)
    if (delta>0)
        {
            d=0.5*(SizeZ-FixedZ)-DeltaZ;
            if (d>0.5*delta)
                fprintf(f, "[2]=%f\n",DeltaZ+0.5*delta);
            else
                fprintf(f, "[2]=%f\n",DeltaZ+d);
        }
    else
        fprintf(f, "[2]=%f\n",0.5*(SizeZ-FixedZ));
#endif}
fflush(f);
fclose(f);
return true;}
const double Pi2=6.2831853071796;
void SavePicture(const char *name)
{int i,j,m;
FILE *f=fopen(name,"w");
for(m=i=0;i<KolPol;i++)
    m+=Poly hedrons[i].kol;
fprintf(f, "%f %f %f\n%d",SizeX,SizeY,SizeZ,m);
for(i=0;i<KolPol;i++)
    for(m=0;m<Poly hedrons[i].kol;m++)
        {
            fprintf(f, "%f\n", Poly hedrons[i].Pol[m].kol);
            if (Domain)
                for(j=0;j<Poly hedrons[i].Pol[m].kol;j++)
                    fprintf(f, "%f %f %f
", Poly hedrons[i].Pol[m].Curr[j].xyz[0], Polyhedrons[i].Pol[m].Curr[j].xyz[1], Polyhedrons
[i].Pol[m].Curr[j].xyz[2]);
            else
                for(j=0;j<Poly hedrons[i].Pol[m].kol;j++)
                    #if def NEW_BOX_MODE
                    fprintf(f, "%f %f %f
", Poly hedrons[i].Pol[m].Curr[j].xyz[0]-DeltaX, Poly hedrons[i].Pol[m].Curr[j].xyz[1]-
DeltaY, Poly hedrons[i].Pol[m].Curr[j].xyz[2]-DeltaZ);
                    #else

```