Квадратичная оптимизация на булевых полиэдрально-сферических конфигурациях

СОДЕРЖАНИЕ

[ВСТУП 7](#_Toc532608481)

[1 КОРОТКІ ТЕОРЕТИЧНІ ВІДОМОСТІ ПО ТЕОРІЇ ДИПЛОМНОГО ПРОЕКТУ 9](#_Toc532608482)

[1.1 Поняття множини, мультимножини і конфігурації. Кінцеві множини і багатогранники. 9](#_Toc532608483)

[1.2 Поняття точкової конфігурації і багатогранника на такій конфігурації 15](#_Toc532608484)

[1.3 Поліедрально-поверхневі представлення 19](#_Toc532608485)

[1.4 Базові С-множини 23](#_Toc532608486)

[Усі класи, що розглядаються, є поліедрально-сферичні з наступними параметрами описаної сфери: і центром у точці . 24](#_Toc532608487)

[Висновки до розділу 1 24](#_Toc532608488)

[2 МАТЕМАТИЧНА МОДЕЛЬ ЗАДАЧІ 26](#_Toc532608489)

[2.1 Опис вхідних даних до дипломного проекту 26](#_Toc532608490)

[2.2 Побудова математичної моделі. 30](#_Toc532608491)

[2.3 Область застосування 34](#_Toc532608492)

[Висновки до розділу 2 37](#_Toc532608493)

[3 АНАЛІЗ РОЗВИВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕНЯ ЩО ВИКОРИСТОВУЄТЬСЯ ДЛЯ РІШЕННЯ ПОСТАВЛЕНОЇ ЗАДАЧІ 38](#_Toc532608494)

[3.1 Оптимізаційні можливості різноманітних математичних пакетів і вбудованих функцій у мови програмування для рішення задач квадратичної оптимізації. 38](#_Toc532608495)

[3.2 Короткий опис алгоритму пошуку рішення в задачі квадратичної оптимізації на булевих поліедрально-сферичних конфігураціях. 43](#_Toc532608496)

[Висновки до розділу 3 44](#_Toc532608497)

[4 ОПИСАНИЕ ПРОГРАММНОГО ПРОДУКТА 45](#_Toc532608498)

[4.1 Вектор необхідних та достатніх на вхід оптимізаційному пакету початкових даних для коректної роботи оптимізаційного алгоритму. 45](#_Toc532608499)

[4.2 Граф варіантів використання. Побудування діаграми класів. 46](#_Toc532608500)

[4.3 Вибір мови програмування та середовища розробки. 56](#_Toc532608501)

[4.4 Підключення бібліотеки IPOPT 58](#_Toc532608502)

[4.5 Короткий посібник користувачеві по використанню програмного продукту. 61](#_Toc532608503)

[4.6 Результати проведеного експерименту 70](#_Toc532608504)

[Висновки до розділу 4 73](#_Toc532608505)

[ВИСНОВКИ 86](#_Toc532608513)

[СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ 89](#_Toc532608514)

# ВСТУП

Математичне моделювання безлічі технічних систем в сучасному світі пов'язано з оптимізаційними процесами складних геометричних об'єктів. Обробка та усілякі перетворення інформації про складні об’єкти представляють собою широкий кластер задач і є базовою складовою передовою науково-технічного прогресу.

В даній магістерській роботі проведено огляд теоретичних даних та достовірних джерел інформації, щодо теми дипломного проекту, побудовано і розроблено власну математичну модель задачі, досліджено показники продуктивності математичних оптимізаційних пакетів для класу задач, розглянутих у межах цієї роботи, проведено математично-дослідницький експеримент з метою пошуку залежностей часу пошуку розв’зку від розмірності певної задачі, від кількості обмежень системи та від інших параметрів, котрі описано у пояснювальній записці.

Розглянутий в дипломному проекті клас задач охоплює приклади задач нелінійної, а саме комбінаторної, оптимізації.

В рамках дипломного проекту було розроблено власну математичну модель, котра стала прототипом програмного продукту, котрий, також, було розроблено для демонстрації отриманого рішення за допомогою сучасних інформаційних засобів.

Створений власний програмний комплекс здатен проводити пошук локально-оптимального розв’язку**,** починаючи оптимізаційний спуск до певної екстремалі з початкової точки. У класичному ж випадку, коли цільова функція не має обмежень, ми б мали змогу щоразу, в залежності від початкової точки, спускатися до певного екстремуму.

Складність пошуку локального рішення задачі квадратичної оптимізації на поліедрально-сферичних конфігураціях, полягає в знаходженні таких е-конфігурацій, які б задовольняли поставленим вимогам та заданій системі обмежень за якомога менший проміжок часу. Поліедрально-сферичні обмеження представляють собою обмеження системи, для виконання котрих необхідно, щоб виконалася умова потрапляння точки на перетин одиничного гіперкуба та гіперсфери радіуса одиниця.

Якщо коротко охарактеризувати область застосування цього класу задач, то такі задачі знаходять застосування в теорії графів, в NP-складних задачах, котрі зводяться до задачі теорії графів, телекомунікаційні технології, розміщення складових приміщень, оборонна галузь, соціальні та молекулярні мережі. Багато прикладів застосування це: VLSI-дизайн, обробка зображень, комп’ютерний зір, наукові розрахунки, паралельне програмування, прокладання маршрутів пересування транспортних засобів чи фізичних осіб та задачі балансування. Своє застосування такі задачі знайшли ще й в інформаційній кібербезпеці світових відомств з питань безпеки людини.

Актуальність цієї роботи полягає в розробці власної робочої математичної моделі, в розробці власного алгоритму пошуку рішення поставленої задачі, розглянуто та запропоновано метод пошуку локального рішення. Важливим пунктом цього диплому магістра – це отримання і описання отриманих власноруч результатів числових експериментів в процесі пошуку відповідного екстремуму.

# 1 КОРОТКІ ТЕОРЕТИЧНІ ВІДОМОСТІ ПО ТЕОРІЇ ДИПЛОМНОГО ПРОЕКТУ

## 1.1 Поняття множини, мультимножини і конфігурації. Кінцеві множини і багатогранники.

Поняття множини є первісним і визначення не має. Під множиною мається на увазі сукупність різноманітних елементів, об’єднаних загальною ознакою і тому розглядаються як одне ціле.

Для визначення множин використовуються, як правило, великі букви латинського алфавіту, а для їх елементів – відповідні маленькі букви. Наприклад, якщо сукупність елементів множини Х характеризується ознакою , то використовується запис , а сама ознака називається характеристичною властивістю множини X [1]. В залежності від кількості елементів множини бувають кінцевими, рахункові та іншими. Для кінцевих множин використовуються запис:

|  |  |
| --- | --- |
|  | (1.1) |

а для злічених:

|  |  |
| --- | --- |
| X = {x1, ..., xη, ...}. |  |

Для завдання кількості елементів кінцевої множини Х будемо використовувати означення |X|, тобто в поданні (1.1) .

Під мультимножиною (a multiset) будемо мати на увазі сукупність елементів

|  |  |
| --- | --- |
|  | (1.2) |

не обов’язково різних, об’єднаних загальною ознакою.

Різноманітні елементи мультимножини G утворюють множину, що називається його основою (the underlying set), що будемо позначати[1]:

|  |  |
| --- | --- |
|  | (1.3) |

Для завдання мультимножини необхідно вказувати також кратності (the multiplicities) його елементів. Нехай i – кратність елемента основи

Вектор кратності

|  |  |
| --- | --- |
|  | (1.4) |

що називається первісною специфікацією (the primary specification) мультимножини

З іншої сторони, мультимножину G представимо у вигляді:

|  |  |
| --- | --- |
|  | (1.5) |

Таким чином, для представлення мультимножини мається декілька способів. Далі будемо використовувати означення:

а) для невпорядкованих послідовностей елементів природи;

б) , - для упорядоченных последовательностей элементов;

г) - для векторів, тобто упорядкованих послідовностей дійсних чисел [2].

Зауважимо, що мультимножина G, що співпадає зі своєю основою S(G), є множиною, причому

|  |  |
| --- | --- |
|  | (1.6) |

При цьому що підкреслити, те що серед елементів G нас цікавлять тільки різні, будемо здійснювати перехід до S(G).

Відповідно з приведеними вище позначеннями якщо G – мультимножина виду (1.5), то упорядкована послідовність його елементів виду , а - вектор виду .

Зауважимо, що операція переходу від G до визначена завжди, в той час як перехід від G до (G) визначено тільки для числової мультимножини G.

При цьому для числових мультимножин будемо розрізняти представлення і , припускаючи, що вони складаються з елементів різної природи, між якими існує взаємно-однозначне відношення [3].

Далі буде використовуватися зворотня операція переходу від впорядкованих послідовностей і векторів до мультимножини їх координат. Для цього будемо використовувати позначення . Наприклад, якщо вектор виду , то сукупність його координат буде . На мультимножинах аналогічно з множинами можна ввести різні типи відношень. Розглянемо множину . Її опуклою оболонкою називається сукупність всіх опуклих комбінацій точок а аффінною оболонкою називається множина всіх комбінацій її точок. Опукла оболонка множини D позначуються conv D і af f D відповідно.

Опукла оболонка кінцевої множини D евклідового простору називається опуклим багатогранником, що було породжено множиною D [3].

Виходячи з цього визначення , опукла оболонка

|  |  |
| --- | --- |
|  | (1.7) |

Точкова конфігурація Е представляє з себе багатогранник. Розмірність (a dimension) багатогранника – це розмірність його аффінної оболонки:

|  |  |
| --- | --- |
|  | (1.8) |

тобто це число лінійно незалежних векторів в aff (P). Багатогранник, що задовольняє умові (1.5), називається d-мірним багатогранником (d-багатогранником, a d-polytope). Багатогранник називається повномірним (a full-dimensional polytope), якщо його розмірність співпадає з розмірністю простору, в котрому його задано. Наприклад, багатогранник – повномірний тоді й тільки тоді, коли .

Оскільки з припущення E – не порожня, P також непорожня множина, цьому для нього існує опорна гіперплощина H, тобто площина, що має загальні точки з P, і така, що весь багатогранник розміщено в одному з двох напівпросторів, що визначає гіперплощина H [4].

Створене в результаті непорожня множина називається граню багатогранника P, що була породжена опорною гіперплощею H, а i = dim F – розмірністю грані многогранника. Грані розмірності і називають і-гранями багатогранника і в залежності від величини для позначення i-граней P використовують наступну термінологію: 0-грані – це вершини P, 1-грані – його ребра, . . . , d − 1-грані – гіперграні.

Нехай - множина i-граней F , а – їх кількість , , H – множина опорних площин до гіперграней P. Величини є компонентами f-вектору багатогранника P, такого, що .

Множина вершин багатогранника P прийнято позначати або

V (P), ребер P або E(P), гіперграней - P чи F(P), тобто

|  |  |
| --- | --- |
|  | (1.9) |

відповідно,

|  |  |
| --- | --- |
|  | (1.10) |

Для множини вершин багатогранника P будемо використовувати позначення V , а для його елементів – , тобто

|  |  |
| --- | --- |
|  | (1.11) |

де . Аналогічно будемо використовувати позначення F[.] для гіперграней P і – для відповідних їм опірних гіперплощин:

|  |  |
| --- | --- |
|  | (1.12) |

Розглянемо так назване V - і H-представлення багатогранника P [5].

V-представлення (V-presentation, параметрична форма завдання) багатогранника P – це його завдання опуклої оболонки множини його вершин:

|  |  |
| --- | --- |
|  | (1.13) |

Зазначимо, що представлення (1.7) і V -представлення (1.8) є окремим випадком представлень P виду:

|  |  |
| --- | --- |
|  | (1.14) |

Відмінною особливістю V-представлення в сімействі (1.13) є те, що воно є мінімальним в тому сенсі, що виключення будь-якого елементу із V і наступне формування опуклої оболонки веде до утворення багатогранника, відмінного від P:

|  |  |
| --- | --- |
|  |  |

Відповідно, воно мінімально по кількості точок в породжуючий Р множині, інакше кажучи, кількість елементів V -представлення багатогранника [5]. Випадок, коли сімейство (1.13) включає єдине представлення P, що можливо лише тоді, коли

|  |  |
| --- | --- |
|  |  |

Множину E будемо називати вершино-розташованою (a vertex located set), якщо воно співпадає з множиною вершин своєї опуклої оболонки, тобто

|  |  |
| --- | --- |
|  | (1.15) |

З урахуванням (1.7), умова (1.10) можна переписати у вигляді виразу (1.1), що буде використано в подальшому як умова вершинної розташованості E.

Якщо множина E не є вершино-розташованою, то для неї виконано умову E = E\V 6= ∅, відповідно, воно містить внутрішні точки багатогранника P чи його граней [6].

Перевірка умови (1.8), а у випадку її невиконання, виявлення ознак (необхідних і достатніх) того, чи є довільна x ∈ E вершиною P – це задача побудування V-представлення багатогранника P, якщо сам від заданий у формі (1.9). Рішення цієї задачі для вершино-розташованих множин тривіальне – належність E є необхідною і достатньою умовою того, що точка х була вершиною Р. Одначе для довільної E перевірка виконання умови (1.8), відділення множини E з E і, відповідно, задача побудування V -представлення може викликати значні труднощі.

Данна задача вирішена для вершино-розташованих множин, якщо вони самі задані множиною своїх елементів. Окрім цього, ці множини мають ряд специфічних особливостей, котрі будуть розглянуті далі[7]. Усіляка не вершино-розташована множина Е дозволяє провести декомпозицію (1.14), (1.16) на вершино розташованих H-представлень. H-представленнях (H-presentation, аналітична форма завдання) багатогранника P, а саме, представлення за допомогою лінійної системи обмежень:

|  |  |
| --- | --- |
| , | (1.16) |

## 1.2 Поняття точкової конфігурації і багатогранника на такій конфігурації

Для вершино розташованих множин існує опуклі поліедрально-сферичні представлення. З іншої сторони, згідно з теоремою, для будь-якої функції, що була задана на вершино розташованій множині, можна побудувати опукле (строго/сильно опукле) продовження. Ці особливості дозволяють при рішенні розглянутих класів задач комплексно використовувати властивості множин, їх опуклих оболонок, описаних строго опуклих поверхонь та опуклих функцій, та будувати їх [7].

Для довільних кінцевих точкових конфігурацій це не вірно. Так, наприклад, для множин, котрі не є вершино розташованими, опуклих функціональних представлень й опуклих продовжень з них не існує [7, 8].

Встановимо взаємозв’язок довільної кінцевої точкової конфігурації з вершино розташованими множинами. Розглянемо наступні два підходи.

Перший представляє з себе вибір строго опуклої функції h(x) та побудування розкладання множини E по сімейству строго опуклих поверхонь, наприклад, сферично розташованих. Множини (1.16), утворені в результаті такого розкладання, будуть поверхнево-розташованими, а тому, і вершино розташованими згідно з теореми 2.3. Таким чином, здійснено перехід до розглядання сукупності вершино розташованих множин.

Другий спосіб базується на переході в розширений простір , розмірність якого буде суттєво залежати від кількості значень координат, що приймаються точками E, тобто від числа рівнів E по координатам і від конкретних значень [8].

Зауважимо, що якщо E не є вершино розташованою, то число перевершує два, оскільки:

|  |  |
| --- | --- |
| . | (1.17) |

При цьому для довільної не вершино розташованої множини існує розширене поліедрально-сферичне представлення. Дійсно, по будуванню множин виду, координати довільних точок x ∈ E приймають кінцеве число значень із цих множин, а саме,

|  |  |
| --- | --- |
|  | (1.18) |

так як дискретні величини представлено за допомогою булевих змінних

|  |  |
| --- | --- |
|  | (1.19)  (1.20) |

Утворена дискретна множина задовольняє умовам (1.19), (1.20) буде вершино розташованою як підмножина вершино розташованої булевої множини. До неї може бути застосовано поліедрально-сферичне представлення [8].

Розглянемо деякі операції над кінцевими множинами точок евклідова простору та дослідимо властивості отриманих кінцевих точок конфігурацій та відповідних комбінаторних багатогранників.

Введемо до розглядання набір кінцевих точкових множин, що задано в евклідовому просторі розмірності не вище n, і будемо формувати з них множину E виду (1.11) як результат деяких теоретико-множинних операцій над цими множинами [8].

Нехай n розбито на L доданків та задано множини

|  |  |
| --- | --- |
|  | (1.21) |

Елементи множини , а в координатній формі –

Для характеристик множин (1.21) та пов'язаних з ними понять, таких як опукла оболонка, описана поверхня, описана гіперсфера, H-представлення та множина гіперграней багатогранника і т.д. буде використано позначення, аналогічні тим, що було прийнято для множини E у пункті 1.1. При цьому до введених раніше позначень будемо додавати верхній індекс, наприклад:

|  |  |
| --- | --- |
|  | (1.22) |

це багатогранники, що відповідають множинам (1.21). Для кожного фіксованого система лінійних обмежень:

|  |  |
| --- | --- |
| ,  . | (1.23)  (1.24)  (1.25) |

H-представлення багатогранника . і – множина його гіперграней та відповідних опірних гіперплощин, а – вершин ( ). Ступінь вершини багатогранника будемо позначати . У випадку, якщо всі вершини – регулярні, то через будемо позначати ступінь регулярності його вершин. Нехай також буде утворюючою множиною для , , – числом рівнів в цілому і по координатам відповідно, а – числом рівні багатогранника . Розмірність багатогранника позначимо так – , а його f-вектор – [8].

В окремих випадках множини (1.22) можуть володіти специфічними властивостями. Ці множини вершино розташовані (1.26) та розташовані на деяких поверхнях (1.27).

|  |  |
| --- | --- |
| . | (1.26)  (1.27) |

Ця умова означає, що множини (1.22) лежать на поверхнях, заданих вказаними рівняннями. Представимо це у вигляді:

|  |  |
| --- | --- |
|  | (1.28)  (1.29) |

Вписані у поверхні (1.29), що припускають виконання умов (1.28):

|  |  |
| --- | --- |
|  | (1.28)  (1.29) |

поверхньо-розташовані множини, що передбачає виконання трьох умов – (1.28), (1.29),

|  |  |
| --- | --- |
| – строго опукла, . | (1.30) |

Множини (1.22) як сферично-розташовані множини. Тут передбачається, що виконана умова (2.35) та . Для мінімальної сфери, що описано навколо , будемо використовувати позначення [8].

Еліпсоїдально-розташовані множини передбачають що виконується умова (2.35), а також:

|  |
| --- |
| ,  . |

Дослідимо, в яких випадках сконструйована із множин (1.22) нова множина E успадкує їх властивості, такі як вершинна і сферична розташованості и т.п., а також як пов’язані властивості багатогранника разом із специфічними особливостями багатогранників (2.31), що взято за основу побудування P. При цьому для f-вектору багатогранника будемо використовувати позначення .

## 1.3 Поліедрально-поверхневі представлення

З огляду на те, що PSR (1.55) представляє E перетином строго опуклої поверхні S та багатогранника (1.55), (E.PSR) включає рівняння:

|  |  |
| --- | --- |
|  | (1.31) |

цієї поверхні та H-представлення (1.30) багатогранника P. Це нелінійне f-представлення виду (1.25), (1.31), що представлено у вигляді (1.23) наступним чином:

|  |  |
| --- | --- |
| , | (1.32) |

де строга частина має вигляд тобто нумерація компонент строгої частини починається з нуля. (1.32) – це змішане f-представлення з параметрами:

|  |  |
| --- | --- |
| . | (1.33) |

Оскільки всі його обмеження, за виключенням (1.31), лінійні, вид (E.PSR) повністю визначається видом функції f(x). Так якщо вона квадратична – в цілому це буде квадратичне f-представлення, якщо біквадратна – біквадратне і т.д. Також, за рахунок обмеженості S, (E.PSR) буде завжди обмеженою [9].

Відповідно з класифікацією PSR, що представлено в п. 1.2, введемо декілька класів (E.PSR) залежно від виду описаної поверхні. Якщо (2.32) – рівняння гіперсфери, (E.PSR) назвемо поліедрально-сферичним f-представлення (a polyhedral-hyper spherical representation, (E.PSpR)); якщо рівняння (1.31) задає еліпсоїд, (E.PSR) будемо називати поліедрально-еліпсоїдним f-представленням (a polyhedral-ellipsoidal representation, (E.PER)); у випадку, якщо (1.31) – це рівняння суперсфери, то (E.PSR) будемо називати поліедрально-суперсферичним f-представленням (a polyhedral-super spherical representation, (E.PSsR)).

Необхідною умовою того, щоб (E.PSR) була незбитковою, є використання у якості (1.31) незвідного H-представлення P. Гарантувати те, що при цьому (E.PSR) є незбитковою, можна тільки для простих багатогранників [9].

Також зауважимо, що додавання будь-якого числа обмежень в систему (1.32) все ще буде поліедрально-поверхневим f-представленням деякої дискретної множини, порядок якої відрізняється від (1.34) в більшу сторону. З іншої сторони, якщо замість P взяти релаксаційний багатогранник , що задовольняє умові

|  |  |
| --- | --- |
|  | (1.34) |

E буде допускати (E.PSR) з участю H-представлення . В підсумку її порядок буде відрізнятися від (2.34) в меншу сторону. Зрозуміло, що E допускає (E.PSpR), якщо вона є поліедрально-сферичною; (E.PER), якщо вона поліедрально-еліпсоїдна; (E.PSsR), якщо вона поліедрально-суперсферична. Виникає питання, для яких FPC існують поліедрально-поверхневі представлення [9].

Для того, щоб множина виду (1.9) допускала побудування (E.PSR) необхідно, щоб вона була вершино-розташовано. Доведемо це від противного:

Положимо, E має (E.PSR) у формі (1.33), при цьому E – не вершино-розташована. Тоді існує , для якої, з однієї сторони, , оскільки E – PSS. З іншої сторони, можна представити опуклою лінійною комбінацією точок V. Але, оскільки за умовою, – строго опукла, звідси випливає, що , отримали суперечність, що доказує наше припущення про вершину розташованість E.

Виходячи із зауваження 1.5, для конкретної множини E, щоб вирішити задачу побудування (E.PSR), достатньо знайти строго опуклу функцію , що приймає E нульове значення, а також H-представлення багатогранника P [10].

Для довільної вершино-розташованої множини питання, чи допускає вона PSR, залишається відкритим. Відокремлення окремих класів кінцевих точкових конфігурацій, що допускають PSR, основується на дослідженні їх геометричних особливостей, а також властивостей функцій на них. Одним із способів встановити це є пошук строго опуклої поверхні S, що описана навколо E (далі (PSR.Scheme 1)). Одночасно встановлюється вершина розташованість цієї множини.

Перелічимо деякі із класів кінцевих точкових конфігурацій, для яких задача побудування PSR завідомо вирішується. Для цих класів побудування конкретного (E.PSR) зводиться до пошуку (P.HR) і рівняння S [10].

Так, наприклад, відомо, що довільні n + 1 – на точка в деякій гіперсфері, цьому множина E міцності, не перевищує n + 1, завідомо є PSpS, і в якості (1.33) у її (E.PSpR) може бути взято рівняння гіперсфери, що може бути знайдено в результаті рішення лінійної системи. Якщо E - C – множина перестановок (**поки що поняття не введене**), що індуковано G, її точки мають однакову -норму:

|  |  |
| --- | --- |
|  | (1.35) |

(1.29) говорить про те, що E - SLS, навколо якої можна описати суперсферу з коефіцієнтом деформації для будь-якого .

Таким чином, кожне з рівнянь цього сімейства задає описану навколо Е суперсферу, серед яких є гіперсфера. Таким чином, E дозволяє побудування не тільки (E.PSsR), але й (E.PSpR). Як виявляється, дворівнева по координатах множина E також дозволяє побудування (E.PSpR), оскільки, як не важко побачити з (1.16), всі її точки рівновіддалені від точки:

|  |  |
| --- | --- |
|  | (1.36) |

тобто лежать на гіперсфері з центром у . Ще один клас множин, що дозволяє PSR – це дворівневі множини[10].

На закінчення розглянемо ще один спосіб побудування (E.PSR) вершино-розташованої множини, що задано в формі (1.9) (далі (PSR.Scheme2)). Він оснований на виборі в якості кусочно-лінійної функції, що задає ∂ P.

По побудуванню ця функція буде задовільнять умові , звідки випливає (1.29). Зафіксуємо параметр та побудуємо сильно опукле продовженя з параметром функції з E на . Продовжимо на зберіганням виразу функції та отримаємо , сильно опуклу, а значить строго опуклу, на P. Якщо ця функція залишається строго опуклою в межах області

|  |  |
| --- | --- |
|  | (1.37) |

E допускає PSR з участю строго опуклої поверхні . Якщо строга опуклість функції не розповсюджується на всю область C, для E має місце (1.55), тобто (P.HR), в сукупності з рівнянням , ткаод визначають E, тобто є (E.FR) [10].

## 1.4 Базові С-множини

Нехай G буде мультимножиною для C-множини , при цьому E об’єднує e-конфігурації. Мультимножина – в математиці, це множина в якій для кожного елемента запам'ятовується не лише його входження, але й кількість входжень. Назвемо її базовою С-множиною (Cb-множиною) для е-конфігурацій даного виду.

Подання безлічі E за допомогою функціональних залежностей:

|  |
| --- |
| , |

Записи вище називаються функціонально-аналітичним або f - уявлення безлічі E.

Перший запис система вгорі називається суворою частиною f - уявлення, а запис нижче – нестрога частина, кількість обмежень – це його порядок. Так, m буде порядком f-уявлення, а , – порядком його суворої і не суворої частин відповідно.

Класифікація f - уявлень може бути проведена в залежності від виду функції, а також порядку суворої і не суворої його частин і f - уявлення в цілому. Так, з вигляду функцій вище, f - уявлення безлічі E можуть бути лінійні і нелінійні, безперервні, диференціюються, гладкі, опуклі, поліноміальні, тригонометричні і т.п.

Введемо для позначення Е множин, а також для Ec-множин, що є їх прообразами при відображенні .

Базові множини спеціальних е-конфігурацій перестановок та розміщень будемо називати спеціальними C-множинами перестановок та розміщень відповідно [11]. Cb-множиною назвемо C-множину булевих перестановок і позначимо , C-множина булевих розміщень. – це вектор булевих значень із 0 та 1, де n вказує на кількість 0 або 1 у множині. В класі відокремимо ще деякі спеціальні підкласи: якщо E індуковано для прообразів вище приведених Cb-множин перестановок введемо позначення:

|  |
| --- |
|  |

Для прообразів вказаних Cb-множин розміщень:

|  |
| --- |
| . |

## Усі класи, що розглядаються, є поліедрально-сферичні з наступними параметрами описаної сфери: і центром у точці .

## Висновки до розділу 1

В цьому розділі було розглянуто базові теоретичні відомості природи квадратичної задачі оптимізації на поліедрально-сферичних конфігураціях. Розкрито сутність деяких основних математичних термінів.

# 2 МАТЕМАТИЧНА МОДЕЛЬ ЗАДАЧІ

## 2.1 Опис вхідних даних до дипломного проекту

Щоб сформувати пункт записки 2.2 «Побудова математичної моделі», необхідно спочатку сформування систему з вхідними даними до дипломного проекту відповідно встановленої теми, а це – «Квадратична оптимізація на булевих поліедрально-сферичних конфігураціях».

Раніше зазначено деякі загальні слова щодо вмісту пояснювальної записки, котрі кажуть нам, що в цьому дипломному проекті розглянуто задачі квадратичні і вони мають обмеження спеціального виду, а саме поліедрально-сферичні. Поліедрально-сферичні обмеження – це обмеження системи, котрі виконуються на перетині одиничного гіперкуба та гіперсфери описаної навколо одиничного гіперкуба**,** відповідно, вектор для завдання повинен бути сформований відповідно певних правил та вимог щодо постановки задачі. Такий вектор буде складатися з декількох матриць різних розмірностей, на перетині строк і стовпців, котрих будуть знаходитися певні числові коефіцієнти. Розглянемо, що ж за інформацію необхідно сформувати перш за все, на вхід до дипломного проекту [11].

Для більш простого представлення інформації об'єднаємо і опишемо загальні характеристики у вигляді векторів із даними. Ці вектори будуть узагальненням певних наборів із даними з описанням.

Зазначені до розв’язання завдання містять квадратичну природу, для нас це означає, що ми можемо не вище другого ступеня. Це означає, що допустимо лише піднесення до степеню 2 членів формули, а також, не допускається перемноження більш, ніж двох аргументів функції першого ступеня, якщо ж, наприклад, ми маємо три аргументи у першому степені, котрі об’єднані операцією добутку, то це вже буде поліном третього порядку, а задачі такого класу в цьому обмежені на розгляд темою диплома.

Як було зазначено, розглядається лише клас задач із степенем полінома не вище другого. Для того, щоб згенерувати такий поліном, на ділі потрібно створити двовимірний масив даних з коефіцієнтами, ці коефіцієнти генеруються випадковим чином. Зазвичай така матриця великих розмірів і може мати до 10000, або і більше елементів [11].

Наступний вектор – це вектор правих частин, котрий ми задаємо у вигляді матриці-стовпця. Цей вектор має таку кількість елементів, як і кількість строк у матриці коефіцієнтів. Вище описаний вектор використовується для запису квадратичної цільової функції, в котрій присутня матриця і вектор, у поліедрально-сферичному уявлені булевої множини, а рівняння гіперсфери і обмеження на змінні разом із лінійною системою додаткових обмежень утворює матриця лівих частин і вектор правих.

Для початку ітераційного процесу пошуку рішення, потрібно згенерувати початкову точку **,** він же початковий набір даних, що ми будемо передавати до оптимізаційного пакету. Цей вектор-значень має розмірність кількості стовпців у матриці коефіцієнтів і включає в себе, в даному випадку, значення, котрі будуть отримані випадковим шляхом на проміжку від 0 до 1, тобто це може буди абсолютно любе випадкове число, так як, розглянутий в цьому дипломному проекті клас задач обмежується квадратичним випадком, тобто, задачею випуклої оптимізації, це буде означати, що поставлена задача у будь-якому разі матиме однакове одне й те саме оптимальне рішення, що дозволяє нам не зволікати увагу на підготовчі методи генерації початкової точки [12].

Наступним вхідним параметром до дипломного проекту є система з обмеженнями. Це як правило, певна кількість рівнянь, котрі тим, чи іншим чином обмежують поводження оптимізаційного алгоритму на етапі спуску до рішення, котре є глобальним. Загальна кількість обмежень системи можна обрахувати за формулою:

|  |
| --- |
|  |

Символьний вираз вище описує кількість лінійних обмежень в залежності від n – кількість координат простору, а 1 – це обмеження на потрапляння на контур гіперсфери.

Система обмежень формується за допомогою генератора даних, котрий особливим чином, на основі введених на попередніх кроках даних, а саме, матриці коефіцієнтів і стовпцях початкових і правих частин, допомагає згенерувати низку гнучких обмежень, котрі призводять до формування сумісної системи обмежень (цей комплекс заходів по підготовці обмежень було самостійно спроектовано і реалізовано в нашому програмному продукті).

Після того, як було описано основні необхідні дані, зробимо загальну декомпозицію за завданнями і виділимо необхідні специфічні параметри для передачі на вхід кожного із завдань.

Для вирішення загального завдання без обмежень необхідно передати тільки вище описані параметри без останнього пункту, це означає, що така задача є найменш складної в обробці та з приводу пошуку рішення, так як, кількість параметрів впливає на загальну систему обмежень прямо пропорційно кількості змінних та кількості обмежень [13].

Розглянемо, що потрібно додатково передати на вхід при вирішенні задачі с додатковими лінійними обмеженнями. При такій постановці, об'єкт формування функції мети залишається тим же самим, а систему обмежень, котра буде зберігатися у вигляді двовимірної матриці, цю матрицю необхідно ввести вручну, або генерувати за бажанням (в нашому випадку, інтелектуальна система згенерує всі обмеження за вас). Матриця обмежень не симетрична відносно головної діагоналі, двовимірна і містить коефіцієнти, котрі задають певні числа. Матриця ж для задання функції мети симетрична.

Складаючи постановку завдань, було запропоновано використовувати тільки лінійні обмеження, конкретно при вирішенні завдання із квадратичної оптимізації і мінімізацією функції мети. При максимізації ж цього показника, ймовірно, що весь час було б отримано одне рішення із одиниць. Ми ж обираємо шлях змінних характеристик і векторів параметрів генерованих параметрів, щоб мінімізувати значення цільової функції в заданій конфігурації.

Таке завдання є дуже складним в питанні машинної завантаженості і призводить до виходу за обмеження кількості ітерацій, заданих попередньо для оптимізаційного пакету. Початкова точка, як було сказано вище, може формуватися за допомогою генератора псевдовипадкових чисел, а може бути складена вручну, або ж, можна отримати ці обмеження за допомогою складних математичних алгоритмів комбінування логіки машинного інтелекту разом із людським. Такій задачі на вхід необхідно передавати додатковий вектор з константних величин, ці величини є тим важливим аспектом, котрий забезпечує точність розрахунків штучного інтелекту і певною мірою сприяє на швидкість та точність отриманих результатів в силу загально-отриманої системи обмежень матриці коефіцієнтів, котра була складена особливим чином [14].

Найближче до навколишнього нас простору – тривимірний евклідів простір, тобто, в ньому виконаються всі необхідні нам для вирішення аксіоми геометрії в евклідовому просторі.

Наступною групою параметрів обов'язкових для початкового вказівки, котрі були описані раніше, є обмеження системи в цілому. Для кожного параметра потрібно виписати m перерахованих обмежень, в сукупності це буде, система з усіх обмежень, яка залежить від розмірності задачі. У нашій системі введені такі обмеження: лінійні обмеження генеровані штучним інтелектом та обмеження на поліедрально-сферичних конфігураціях.

Набори даних задаються у вигляді чисел з плаваючою комою або як цілі числа в десятковій системі числення.

## 2.2 Побудова математичної моделі.

Встановлюється взаємозв’язок аз множинами комбінаторних конфігурацій та евклідово-комбінаторних множин. Представлено класифікацію множин на основі аналізу їх геометричних особливостей та специфіки їх формування.

Описано підходи до моделювання множин за допомогою неперервних функціональних представлень, котрий базується на аналізі геометричних особливостей та мультимножини що їх індукують. Для вершино-розташованих множин викладено специфіка, методи оптимізації та підходи до побудування опуклих продовжень.

Евклідові комбінаційні множини представляють з себе клас множин , що дозволяють відображення в арифметичний евклідовий простір та розглядання їх образів, що називаються s-множинами. Відокремлення класу е-множин було пов’язано з пошуком нових можливостей для вирішення комбінаторних задач методами неперервного та дискретного програмування [15].

Елементами е-множини П є впорядковані вибірки однакового об’єму , де – об’єкти довільної природи. При цьому, оскільки в множинах відсутні однакові елементи, з’являється можливість відображення П у , наприклад, по правилу:

|  |  |
| --- | --- |
|  | (2.1) |

де таке, що

|  |  |
| --- | --- |
|  | (2.2) |

При цьому отримана множина Е називається спеціальною комбінаторною множиною або s-множиною [15].

Перехід від е-множини до розглядання s-множин дозволить досліджувати та застосовувати в оптимізації алгебро-топологічні та тополого-метричні їх властивості, що і поклало початок активним дослідженням в даному напрямку, що називається сьогодні евклідово-комбінаційною оптимізацією (ЕКО).

Правило (1.1) далеко не єдиний спосіб занурення е-множин у . Якщо

|  |  |
| --- | --- |
| , | (2.3) |

то по цьому принципу можна здійснити занурення у , N=n\*m, що забезпечує виконання умови (2.2), наприклад, так:

|  |  |
| --- | --- |
| . | (2.4) |

Використання правила (2.4) відкриває нові перспективи в ЕКО, пов’язані з можливістю використання всієї числової інформації про об’єкти, так як збільшується клас комбінаторних задач, що дозволяють формулювання у вигляді задач дискретного програмування, з іншої боку, розширяє клас е-множин. Саме цьому серед s-множин було відокремлено підклас множин евклідових комбінаційних конфігурацій, що представляють з себе образи множин комбінаторних конфігурацій в сенсі Бержа в арифметичний евклідовий простір. Е – множини є підкласом е-множин, що утворюють в результаті відображення однієї кінцевої множини в іншу [16].

У результаті С-множина Е – це образ в не просто е-множини, а Е множини П, для якої виконано умову (2.3).

Класифікацію С-множин проведемо по двом основним напрямках, перше із яких враховує, що , друге – що це образ П. Так було відокремлено класи вершино- та не вершино-розташованих, поверхнево-розташованих, поліедрально-сферичних, багаторівневих С-множин та інше.

Аналіз індукуючої мультимножини А та утворюючої множини А. Так були виділені класи е-конфігурацій розміщення, перестановок та перестановок із знаком, з повторюваннями та без повторювань, спеціальних, булевих, бінарних, трійчастих і т.д. Що об’єднано з вказаних ознак множин е-конфігурацій, що індуковано А, утворюють базові С-множини такі як С-множини перестановок без повторювань, розміщень з повторюваннями, перестановок із знаком та інших [17].

Основні комбінаційні властивості С-множин, що було досліджено, це потужність множини, розмірність і Н-представлення, в тому числі незвідне, багатогранника , критерій його порожнечі, вершини і суміжність вершин, симетрія, комбінаторна еквівалентність багатогранників та інше.

В рамках дослідження геометричних властивостей базових множин С-множини розглядались наступні задачі: пошук описаних поверхонь, в тому числі опуклих та гладких, розкладання по сімействам площин та строго опуклих поверхонь, декомпозиція Е на попарні С-множини меншої розмірності, що не перетинаються, декомпозиція не вершино-розташованих Е на вершино-розташовані С-множини і т.д.

Інтерес до вершино-розташованим С-множинам, викликано такою особливістю як можливість опуклого продовження довільної функції, що задано на ній, на її опуклу оболонку. А це в свою чергу, дозволяє рахувати, що як цільова функція, так і обмеження в задачах оптимізації (ЗО) задано опуклими функціями.

Вершино-розташовані множини також дозволяють представлення виду , що називається поліедрально-поверхневим представленням, де опукла поверхня обмежує тіло. Знаходження такої поверхні дозволяє при вирішенні ЗО заміняти умову та розглядати як традиційну поліедральну релаксацію, що складається в оптимізації опуклої функції на опуклій поверхні. Комбінація цих двох релаксацій лежить в основі групи поліедрально-сферичних методів. Серед них наближені методи з оцінкою точності, а також точні типи методів гілок та границь. Останні використовують декомпозицію С-множин на попарно-непересічні підмножини цього ж класу меншої розмірності [18].

Представлене рівняння описаної поверхні S та H-представленням багатогранника Р, поверхні S та H-представленням Е.

Представлення множини Е за допомогою функціональних залежностей:

|  |  |
| --- | --- |
|  | (2.5)  (2.6) |

Система (2.5) називається строгою частиною f-представлення, (2.6) – нестрогою частиною, а кількість обмежень – її порядком. Так, m буде порядком f-представлення (2.5), (2.6), а – порядком його строгої та нестрогої частин відповідно. Система (2.5), (2.6) називається строгим f-представленням Е, якщо в ньому є тільки строга частина, інакше – нестрогим.

Усі множини, що ви розглядаєте поліедрально-сферичні та мають нестрогі квадратичні f-представлення, але порядок їх не є поліноміальним. Пошук їх строгих f-представлень, порядок котрих не перевищує n, засновано на властивостях С-множин як образів Е-множин, у тому числі, на аналізі А. При цьому для вершино-розташованих С-множин строгі f-представлення будуються в початковому просторі, а для не вершино-розташованих С-множин строгих f-представлень два, тому, цікавою задачею є пошук тих із них, що визначаються парою гладких функцій, у результаті чого Е утворюється в дотичній точці двох поверхонь, а f-представлення називається дотичним. Дотичні представлення дозволяють комбінувати дві поверхневі релаксації – на S і на опуклій поверхні S' [18].

Методи оптимізації, що дозволяють застосування строгих f-представлень С-множин, включають метод штрафних функцій, метод Лагранжевих релаксацій, ньютонівські методи, метод послідовного під’єднання компонент f-представлень та інші. Особливе місце серед них займає група методів оптимізації на вершино-розташованих С-множинах, в яких комбінується опуклі та строгі продовження цільової функції.

Має сенс також виділити методи оптимізації на так названих добре описаних множинах, що дозволяють ефективно вирішувати поліедральні релаксації. В тому числі, якщо оптимізація проводиться на добре описаному PSS, передбачаються підходи до оптимізації на основі застосування методу штрафних функцій, проектування на PSS та рішення поліедральної релаксації [18].

Як, приклад, нижче показано який вигляд приймає функція цілі. Було використано наступну квадратичну функцію (2.7):

|  |  |
| --- | --- |
|  | (2.7) |

Обмеження на приналежність розв’язку точкам одиничного гіперкубу та гіперсфері описаній навколо нього, було охарактеризовано наступною формулою (2.8):

|  |  |
| --- | --- |
|  | (2.8) |

Система лінійних обмежень, як приклад, вводиться не надто складна (2.9):

|  |  |
| --- | --- |
|  | (2.9) |

## 2.3 Область застосування

Розглянемо сфери застосування квадратичних задач оптимізації на булевих поліедрально-сферичних конфігураціях.

Описані вище завдання, в сьогоднішньому світі знаходять широке застосування в наступних напрямках: оборонна промисловість, теорія графів, VLSI-дизайн, комп'ютерний зір, паралельне програмування, обробка зображень, наукові обчислення, завдання балансування, завдання прокладання оптимальних маршрутів переміщення, в задачах комбінаторики, проблеми балансування, пов'язаних з дизайном чіпів, завантаженням судна, оснащення літаків, геометричний дизайн, проблеми розміщення об'єктів.

Своє застосування такі завдання знаходять також в: теорії розкладів, квадратичних задачах про призначення, в задачах компоновки, дизайні чіпів, плануванні, міських комунікаціях, балансуванню турбін, ергономіці і т.д.

Як не складно здогадатися, судячи з двох попередніх абзаців, область застосування подібного класу задач дуже велика і потреба в нових методах пошуку рішень, в проведених всебічних експериментах ніколи не буде осторонь.

 Всі описані приклади використання можна умовно розділити в залежності від класу аналізованих обмежень виду (2.10) і типів розглянутих множин.

|  |  |
| --- | --- |
|  | (2.10) |

Центральне місце дослідження займають С-множини і описані поверхні [21].

Серед виділених класів С-множин, для яких такі уявлення будуть вершино-розташованими С-множини. Важливе місце в дослідженнях займають аналітичні уявлення С-множин системами нелінійних рівнянь (строгі f-вистави). В цьому класі особливий інтерес представляють дотичні f-вистави представляють С-безлічі як безлічі точок дотику гладких поверхонь. Класифікація нелінійних задач запропонована на рисунку 2.1:

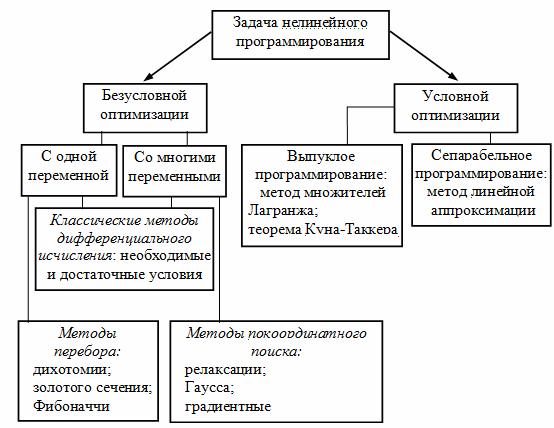


Рисунок 3.5 – Класифікація нелінійних задач

Багатьма науковими співробітниками отримані цікаві результати в області дослідження С-множин, які достовірно-перевірені і, також, як перевірки результатів були отримані при моделюванні множин e-конфігурацій за допомогою теоретико-множинних операцій над іншими С-множинами [21].

Для аналізу такого пласту теоретичної частини запропоновано ряд методів моделювання С-множин, комплексно використовують властивості цих множин як сукупностей точок евклідового арифметичного простору і як образів множин С-конфігурацій.

У контексті вирішення завдань оптимізації, інтерес представляють виділені нові класи вершино-розташованих С-множин і побудова опуклих продовжень на базі знайдених f-представлення. Для тих з них, які допускають поліедрально-сферичні f-представлення, пропонується група поліедрально-сферичних методів, заснована на методах поліедрально-сферичних і екстремальних властивостях цих множин.

Для чисельного застосування пропонується ряд математичних моделей практичних завдань, як задач оптимізації на множинах e-конфігурацій, а також виділяють інші області застосування отриманих результатів.

## Висновки до розділу 2

Відносно прочитаної наукової літератури в цьому розділі було сформовано набір вхідних даних до задачі. Було формалізовано та описано математичну модель стосовно постановки задачі до дипломного проекту.

# 3 АНАЛІЗ РОЗВИВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕНЯ ЩО ВИКОРИСТОВУЄТЬСЯ ДЛЯ РІШЕННЯ ПОСТАВЛЕНОЇ ЗАДАЧІ

## 3.1 Оптимізаційні можливості різноманітних математичних пакетів і вбудованих функцій у мови програмування для рішення задач квадратичної оптимізації.

Розглянемо в цьому розділі пакети-cолвери, котрі можуть допомогти в процесі програмування алгоритму та одночасно, було б легко його імпортувати на платформу програмування .NET, мова програмування C#.

Було проаналізовано багато електронних джерел інформації з математичною теорією, розглянуто деякі, запропоновані до використання математичні пакети.

Nloptr є R-інтерфейсом для NLopt, бібліотеки для вільного розповсюдження та необмежного використання джерел для нелінійної оптимізації, запущених системою. Стівен Джонсон забезпечує загальний інтерфейс для ряду різних безкоштовних процедур оптимізації, котрі доступні в режимі онлайн, а також оригінальні реалізації різних інших алгоритмів. NLopt Бібліотека доступна за GNU Lesser General Public License (LGPL), і авторські права на неї належить різним авторам. Більше інформації на з веб-сайті NLopt, де більш детально можна прочитати про цей пакет для отримання додаткової інформації про доступні алгоритми [19].

Вирішіть задачі оптимізації, використовуючи R-інтерфейс для NLopt і ви не будете витрачати багато часу на підключення інших ресурсів. Це бібліотека для відкритих джерел нелінійної оптимізації, яка забезпечує загальний інтерфейс для багатьох різних безкоштовних процедур оптимізації, а також оригінальних реалізацій різних інших алгоритмів. Під час встановлення nloptr на Unix-системах користувач має перевірити, чи встановлена бібліотека NLopt в системі. Якщо бібліотека NLopt не може бути знайдена, код складається з використанням джерела NLopt, що входить до пакету nloptr.

Пакет Quadprog – ця бібліотека спрямована на написання двох функцій, що вирішують класифікацію двох класів, які можна розділитипроблемою з лінійними векторними машинами підтримки (SVM) в різнихситуаціях: первинний подвійний, без і з шумом. Щоб зробити це працювати, ви повинні мати CVX**.** Вільно використовується в середовищі Matlab.Функція доступна в пакеті оптимізації MATLAB.

Osqp дозволяє розв'язувати параметричну проблему, наприклад, теплі старти між оновленнями**,** параметр, c.f. приклади. Об'єкт, який повертає osqp, містить декілька методів, які можуть бутивикористані для оновлення отриманих деталей проблеми, зміни налаштувань оптимізації або спроби вирішеннязадачі.

Пакет Netlib, котрий доступний для вільного завантаження для впровадження у власний програмний засіб та програмний код. Пакет було створено ще в 1980 році, на мовах С++ і Fortran. Включає в собі безліч алгоритмів вирішення, котрі можна опціонально вмикати або вимикати, його мінусом є старі оптимізаційні реалізації алгоритмів, що, часом призводе до довготривалих розрахунків. Пакет не так багато має функціоналу, і деякі функції не дозволяють працювати з обмеженнями. Складність портування на платформу .NET, також проблема [19].

Пакет TRON за посиланням http://www-unix.mcs.anl.gov/~more/tron/. Це метод Ньютона для вирішення великих завдань обмеженою оптимізації. Використовується метод проекції градієнта для генерації кроку Коші, зумовленого методу сполученого градієнта з неповною факторизаціїєю Холецкого для генерації напрямки і прогнозованого пошуку для обчислення кроку. Пакет обмежений кількістю обмежень, при вирішенні задач великих розмірностей припиняє обчислення, цей пакет є «умовно» безкоштовним, так як, має обмежений термін використання безкоштовної версії.

Оптимізація Toolbox надає функції для пошуку параметрів, які мінімізують або максимізують мети при задоволенні обмежень. Можна використовувати вирішувачі для пошуку оптимальних рішень безперервних і дискретних задач, виконання компромісних аналізів і включення методів оптимізації в алгоритми і програми.

Солвер LOQO для задач гладкої умовної оптимізації, заснований на методі внутрішньої точки, що застосовується для послідовно-квадратичних наближень. У вимогах до пакету, зазначено, що функції мають бути гладкими (в точках, оцінених алгоритмом), LOQO має лінійний та нелінійний солвери для розв’язку різних завдань, опуклих або не опуклих, з обмеженнями або без обмежень. Для задач LOQO в залежності від вхідних параметрів самостійно обирає необхідний тип солвера і продовжує відповідні розрахунки; в іншому випадку він ітерації з даної початкової точки, щоб знайти локально оптимальне рішення [19].

Підтримувані типи проблем:

1. лінійні,
2. квадратичні
3. гладко нелінійні задачі
4. обмеження безперервних змінних.

Розв’язувані завдання можуть бути лінійними або нелінійними, обмеженими або без обмежень. Єдине реальне обмеження полягає в тому, що функції, що визначають завдання, є гладкими (в точках, оцінених алгоритмом). Якщо проблема опукла LOQO знаходить глобальне оптимальне рішення. В іншому випадку він знаходить локальний екстремум поруч із заданою початковою точкою. Розглянуті бібліотеки ефективні при реалізації алгоритмів, мають ряд переваг на ряду.

Наведемо інформаційні відомості щодо IPOPT (http://www-124.ibm.com/developerworks/opensource/coin/Ipopt) – використовує алгоритм внутрішньої точки для задач великої розмірності. Має відкритий вихідний код і не вимагає зобов'язань при створенні комерційних програм.

IBM Research інвестувала в перевидання з відкритим вихідним кодом IPOPT на C++. За допомогою Карла Лайда, який прийшов до відділу математичних наук в IBM Research в якості річного стажиста в 2004. Код був повторно запроваджено з нуля.

Нова версія коду оптимізації IPOPT була збережена в IBM Research. Розробка версії Fortran припинилася. Пакет-бібліотека дозволяє застосувати для розв'язання оптимізаційних задач розміщення методи оптимізації IPOPT імпортовані на всі мови платформи .NET, реалізовані на С++, відповідно застосуємо рефлексію з використанням C# для динамічного завантаження DLL. Солвер має безліч налаштувань роботи алгоритму, включає безліч вбудованих методів, які не потребують повторної реалізації.

Після зваженого аналізу ринку і сильних сторін кожного з солверів було вибрано IPOPT з алгоритмом внутрішньої точки як метод отримання локального вирішення завдань дипломного проекту.

Профіль оптимізатора інтерполяції (IPOPT). У профілі оптимізатора IPOPT використовується програмний вирішувач IPOPT (Interior Point OPTimizer) для мінімізації або максимізації певної функції витрат при наявності обмежень. IPOPT здатний обробляти розріджені проблеми оптимізації з мільйонами змінних і обмежень [19].

IPOPT – це надійний, ефективний, великомасштабний, нелінійний програмний вирішувач, який реалізує метод фільтра інтерполяції, лінійного пошуку для вирішення опуклих і неопуклих, обмежених завдань оптимізації. Ітерації IPOPT включають в себе рішення розріджених, симетричних, нескінченних лінійних систем, що вимагають інтеграції з розрідженим рішенням лінійної алгебри. Astrogator використовує MUMPS, програмний пакет для багатофронтального рішення великих, рідкісних, лінійних систем для цієї мети. Для підвищення продуктивності MUMPS використовуються ефективні алгоритми впорядкування матриці, доступні в METIS. Всі похідні першого порядку, необхідні IPOPT, обчислюються за допомогою чисельного звичайно-диференціювання, тоді як похідні другого порядку отримують методом квазі-Ньютона з обмеженою пам'яттю (L-BFGS).

На рисунку 3.1 можна побачити структуру викликів всередині IPOPT:

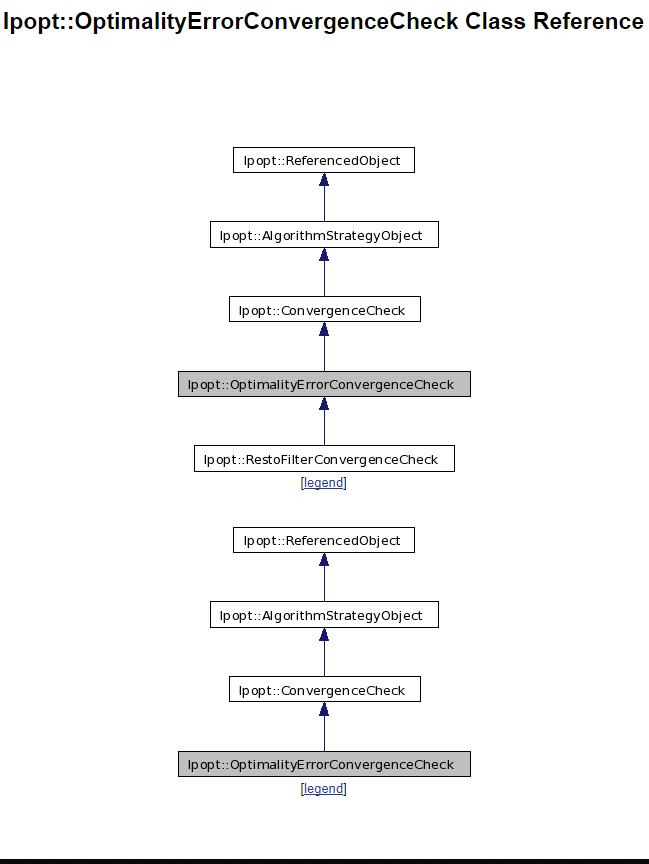


Рисунок 3.1 – Структура викликів всередині IPOPT

Нижче показано наслідування класів у IPOPT (рис. 3.2):

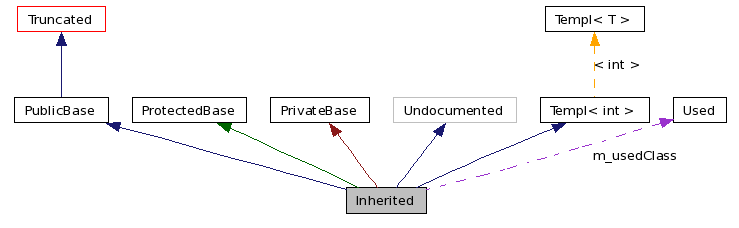


Рисунок 3.2 – Наслідування класів у IPOPT

## 3.2 Короткий опис алгоритму пошуку рішення в задачі квадратичної оптимізації на булевих поліедрально-сферичних конфігураціях.

З огляду на виписану готову математичну модель алгоритм пошуку розв’язку, не займатиме багато часу для опису, обмежимося базовими кроками при пошуку рішення.

Перший етап, це підготовка даних. На цьому етапі ми формуємо функцію цілі та генеруємо набори правих частих. Маючи всі дані на руках, можна описати відсутні обмеження системи і перейти до другого етапу.

Другий етап полягає у формуванні логічних ланцюгів дій по обробці і аналізу отриманих цільовій функції, обмеженням та іншим коефіцієнтам.

Третій етап - це підготовка даних. Дані необхідно підготувати на вхід оптимізаційних пакету, щоб він зміг спуститися до локального мінімуму для початкового завдання. Тут виникають свої особливості, і, в залежності від конкретного пакета оптимізації, буде різнитися і методика підготовки цих величин на вхід.

Четвертий етап – це безпосередня запуск усіх обмежень та розрахунок екстремальної точки. Аналіз отриманих даних на предмет математичного контексту і залежностей. Мається на увазі, знайти якісь залежності зростання показників при зміні інших величин [20].

Останній, п'ятий етап, це дати заключний висновок виходячи із знайдених екстремальних рішень (рис. 3.3).

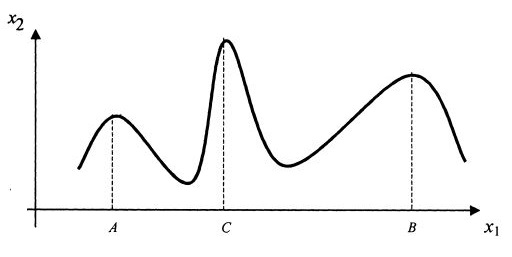


Рисунок 3.3 – Локальні екстремуми

## Висновки до розділу 3

У цьому розділі було проведено детальний аналітичний огляд оптимізаційних пакетів. Розглянуто їх особливості підключення, вказані можливі недоліки в зв'язку з постійним прогресом в світі програмування і збільшення продуктивності алгоритмів що розроблюються.

Коротко описано послідовність алгоритмічних кроків пошуку екстремалів, на підставі який, було вироблено побудова проектування програмного забезпечення.

Описано область застосування даного класу задач з практичними прикладами, відповідно до джерел інформації.

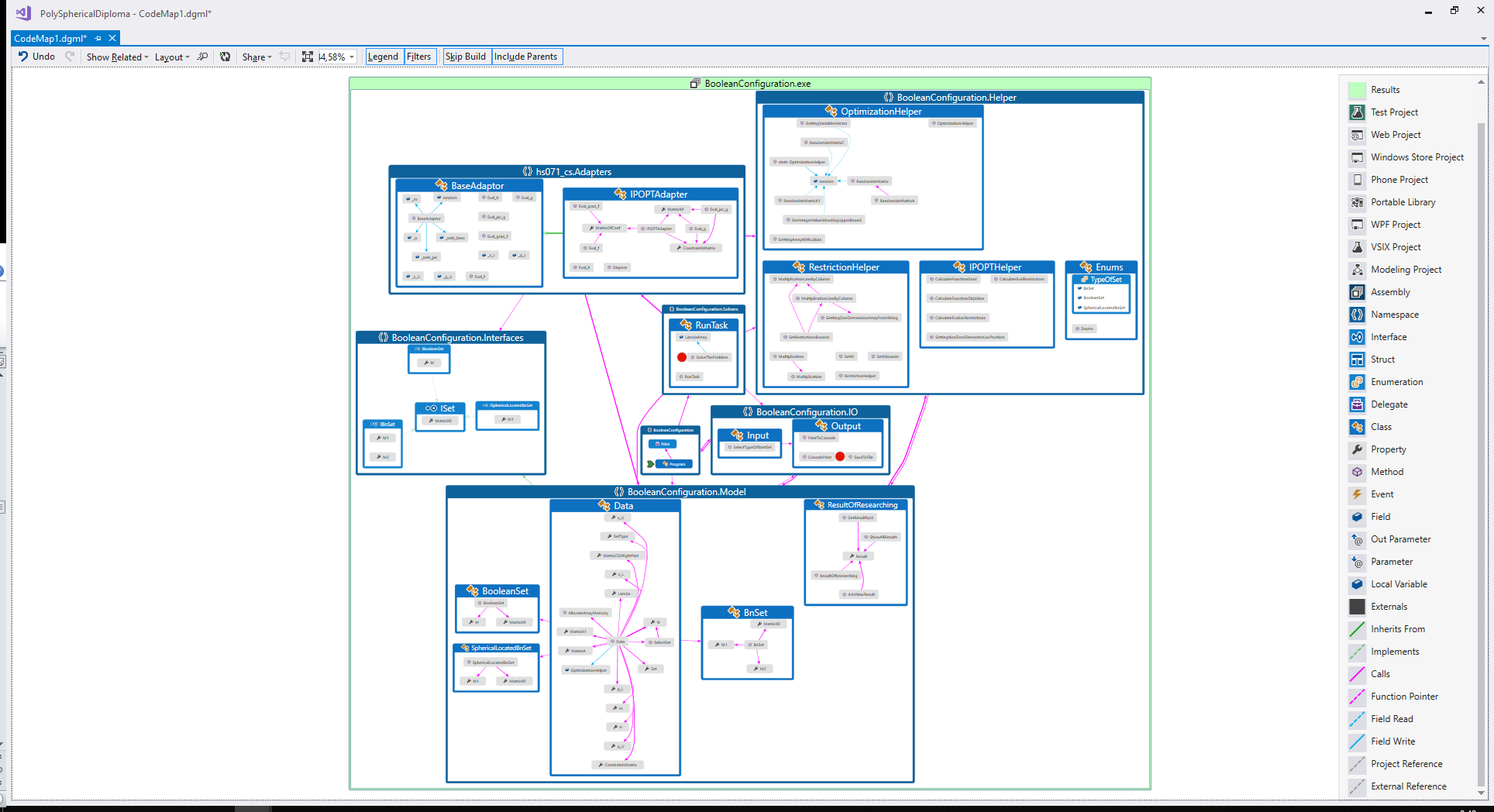


Рисунок 4.19 – Повна діаграма класів

На рисунку 4.46 цифрами показано: 1 – кількість змінних в векторі, 2 – поточна лямбда, 3 – значення цільової функції для отриманого локального рішення, 4 – час рахунку для цього завдання, 5 – значення отриманої оптимальної точки.

## 4.6 Результати проведеного експерименту

Логічним завершенням цього розділу буде проведення пізнавального метричного експерименту. Що ж це буде за експеримент? Опишемо, що ми досліджуватиме.

У цьому невеликому дослідженні будуть проводитися заміри параметрів. Параметрів, які нам не доведеться обчислювати вручну або робити додаткові кроки, все, що нам потрібно ми можемо отримати за допомогою нашого програмного забезпечення.

Якщо уважно подивитися на граф варіантів використання, то можна побачити, що користувач може сформувати кінцевий текстовий звіт. У цьому звіті зберігається багато корисної для нас інформації, щодо розв'язуваної нами задачі. Відкривши його і проаналізувавши, за відповідними підписами (рис. 4.46), можна відразу зрозуміти, що це за дані і як їх можна застосувати в нашому експерименті.

У цьому дипломному проекті розглядаються завдання, які можуть мати негативні значення в якості коефіцієнтів при діагональних елементах, для отримання позитивних величин на діагоналі проводиться операція овипукленія, для цього особливим чином проводиться підрахунок значення Λ (лямбда) і потім це число додається до всіх елементів матриці коефіцієнтів [25].

Наше дослідження полягає в тому, що ми з'ясувати яким чином значення параметра Λ впливає на швидкість рахунки програми, на значення цільової функції, на те, чи буде знайдено рішення.

Функція була нами взята квадратична для випробування, заповнивши матрицю коефіцієнтів числами, задаємо кількість ітерацій або іншими словами, розмірність завдань і натискаємо запуск.

Розмірність вирішуваної задачі нами було вибрано спочатку 300, але з огляду на те, що значення генерувалися рандомно в зазначеному діапазоні, отримуємо повторення для коефіцієнта лямбда, проводимо угруповання за значенням лямбда, а за іншими параметрами зробимо усереднення даних.

На рисунку 4.48 можна побачити деякі з отриманих нами даних:

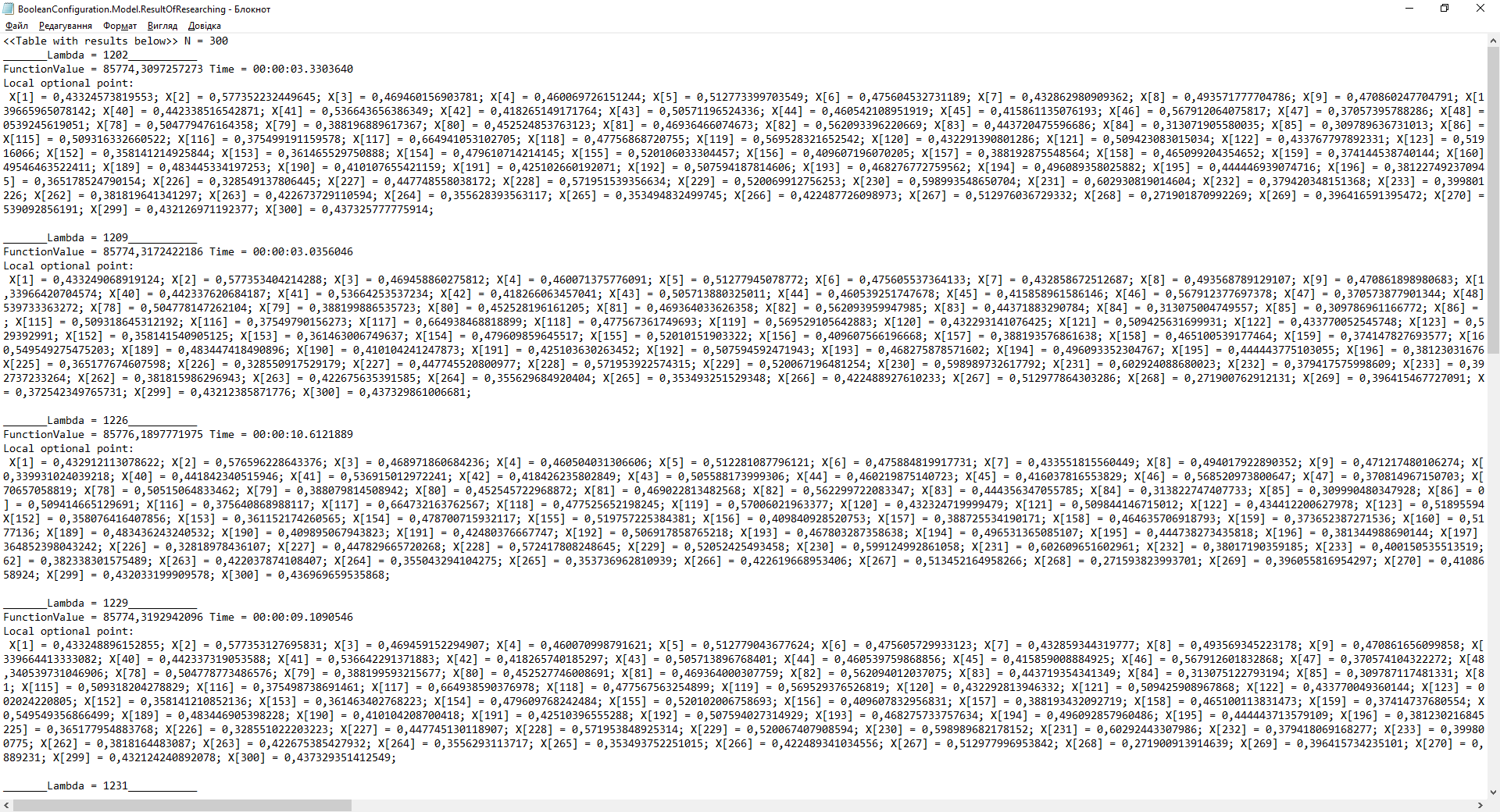


Рисунок 4.48 – Текстовий файл з фрагментом отриманих даних

Таблиця 4.1 – Згруповані отримані дані по лямбда за розмірності 300

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Λ | Function value | Time | Optional Solution Found |
| 1 | 1202 | 85774 | 12.3303640 | + |
| 2 | 1209 | 85774 | 11.0356046 | + |
| 3 | 1226 | 85776 | 10.6121889 | + |
| 4 | 1229 | 85774 | 09.1090546 | + |
| 5 | 1231 | 85774 | 07.8640318 | + |
| 6 | 1232 | 85774 | 08.0438261 | + |
| 7 | 1233 | 85774 | 06.3952917 | + |
| 8 | 1234 | 85774 | 04.2682409 | + |
| 9 | 1235 | 85774 | 03.2349695 | + |
| 10 | 1236 | 85774 | 02.0518446 | + |
| 11 | 1237 | 85774 | 01.9937101 | + |

Проаналізувавши дані, які були виписані в таблицю, чітко можна побачити, що зі збільшенням лямбда (найбільше значення - це провідна лямбда), отримуємо все краще значення цільової функції, ніж на попередньому кроці, бувають, звичайно і виключення, так як, задана деяка точність зупинки комп'ютерних обчислень, також, було отримано більше прийнятний час рахунку при збільшенні лямбда-коефіцієнта [25].

Збільшуємо розмірність задачі до 400 і заносимо дані до таблиці 4.2

Таблиця 4.1 – Згруповані отримані дані по лямбда за розмірності 400

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Λ | Function value | Time | Optional Solution Found |
| 1 | 1578 | 94554 | 7.679 | + |
| 2 | 1601 | 94554 | 6.595 | + |
| 3 | 1604 | 94554 | 6.368 | + |
| 4 | 1605 | 94554 | 7.003 | + |
| 5 | 1608 | 94554 | 5.403 | + |
| 6 | 1611 | 94554 | 5.761 | + |

Шляхом порівняння таблиць 4.1 та 4.2 можемо сказати, що залежність росту часу від кількості змінних дуже помітна.

## Висновки до розділу 4

Було зроблено величезний пласт робіт по розробці програмного продукту, побудовано різні діаграми, діаграми класів, граф варіантів використання. Описано порядок роботи з одиницею нашого продукту програмного забезпечення з повним описом всіх кроків. Був організований аналітичний експеримент з тестування розв'язуваної нами задачі під час якого було з’ясовано, що збільшення кількості змінних у задачі призводить до лінійного росту кривої залежності часу.

# ВИСНОВКИ

В результаті написання магістерського дипломного проекту на тему «Квадратична оптимізація на булевих поліедрально-сферичних конфігураціях» було власноруч розроблено програмний засіб з належним чином оформленою супутньою документацією у вигляді пояснювальної записки. Складовою частиною описової бази є: об’єкт, предмет, мета та постановка задачі.

Представлений пояснювальний документ є математично-описовою базою стосовно зазначеної теми, котрий повністю розкриває зміст виконаних робіт у проекті, використаних у роботі термінів, описує математичну модель, а окремими розділами більш детально розкрито цілі теми, тим самим продемонстровано, що було здійснено ґрунтовну пошуково-дослідницьку роботу з підготовки відповідної темі літератури, а реалізований програмний продукт, написаний по робочій математичній моделі засвідчує правильність отриманих за допомогою програмного засобу розрахунків.

Описовий документ містить відомості про проведений експеримент, в котрому було досліджено залежності часу роботи програми від складності цільової функції, від розмірності задачі, від наявності системи обмежень та їх складності обчислення.

Загальним результатом проведеного експерименту є те, що: при збільшенні числа змінних величин у цільовій функції призводить до лінійного росту кривої часу, тобто, чим складніше задача, тим більше часу знадобиться на пошук певного рішення для неї.

В експерименті також, розглядалося два випадки стосовно обмежень системи. В першому, проводився пошук рішення без обмежень, у другому – з лінійними обмеженнями. Проводилися певні аналітичні спостереження відносно функції цілі, замірявся час, кількість необхідних ітерацій для досягнення рішення, після чого застосовувалася система обмежень с покроковим ускладненням. Спочатку бралася невелика кількість лінійних обмежень і проводився пошук рішення, після чого бралося більш складне обмеження, і, також, проводилися заміри. Було встановлено. Що по мірі ускладнення обмежень крива часу росла лінійно.

Щоб виконаний дипломний проект охопив і максимально розкрив поставлену задачу з описанням запропонованого рішення, всю роботу було розбито на 5 великих розділів, додатково представлено: введення, загальні висновки, список використаних джерел він же список літератури. Кожен з розділів описує свою частку матеріалу і не випадково великим розділом є «Опис програмного продукту», саме в ньому було проведено один цікавий, з точки зору реалізації та отриманих результатів, в світі математичної теорії експеримент, що був запропонований до проведення дипломним керівником з огляду на математичну теорію та чисельні публікації наукових діячів стосовно цієї тематики.

У першому розділі нами було представлено до розгляду базові поняття. Щоб вільно плавати в тематиці дипломного проекту, необхідно знати і розумітися в перерахованих в тексті цього розділу поняттях.

Задля складання другого розділу, побудовано та представлено з описанням, готову до використання математичну модель задачі стосовно постановки задачі. Саме за цією моделлю було виконано аналітичні математичні підрахунки та підготовчі міркування стосовно запропонованого у третьому розділі алгоритму пошуку локального рішення. Третій розділ насичений формулами.

Задля проведення побудови програмного забезпечення необхідно визначитися з мовою програмування та яким чином пропонується спускатися до локального екстремуму. Саме тому, в третьому розділі було проведено огляд існуючих мов програмування, пакетів, розумних калькуляторів щодо можливості використання їх у якості допоміжного застосунку чи солвера.

Опісля проведення аналізу існуючих пакетів для нелінійної та нелінійної оптимізації серед великої купи оптимізаційних пакетів було обрано IPOPT, як допоміжний інструмент для пошуку локального рішення. За допомогою якого і було запропоновано ітераційний метод поетапного пошуку локального мінімуму в розробленому нами програмному засобі, опис якого проведено у розділі 4.

Розділ 4 повністю присвячено розгляду програмного продукту. Перш за все описуються вхідні дані, потім, ми будуємо діаграми: варіантів використання, класів, взаємозв’язків програмних компонентів. Обґрунтовуємо свій вибір мови програмування та демонструємо роботу програми.

Заключним, п’ятим розділом, приводиться економічний підрахунок усіх витрат, прибутку, податків, затрачених ресурсів робітників та інших показників.

Опісля підрахунку усіх економічних показників, заключним словом буде те, що вартість розробки програмного засобу на тему «Квадратична оптимізація на булевих поліедрально-сферичних конфігураціях» складає 72,715.86 гривні, а мінімальна критична кількість одиниць програмного засобу до продажу складає 364 одиниць. Було визначено розробку даного програмного продукту доцільною. Очікується прибуток власника на рівні 8209.02 гривні.

# СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

Пичугина, О. С. Комбинаторные конфигурации: подходы к моделированию и применение [Текст] : статья / О. С. Пичугина – Кропивницкий. : Комбінаторні конфігурації та їх застосування, 2018. – С. 94-100.

Методы линейного программирования [Электронный ресурс] / Воронеж. Институт Менеджмента, Маркетинга и Финансов – Режим доступа : \www/ URL: http://math.immf.ru/lections/302.html/ – 08.12.2016 г. – Загл. с экрана.

Yakovlev, S. V. Optimization problem on polyhedral-spherical configurations [Текст] : статья / S. V. Yakovlev, O. V.Yarovaya. – Кишинёв. : Evrica, 2018.. – C. 232 – 237.

Стоян, Ю. Г. Математические модели и оптимизационные методы геометрического проектирования [Текст] : учеб. / Ю. Г. Стоян, С. В. Яковлев. – К. : Академия Наук УССР Институт проблем машиностроения, 1986. – С. 202 – 227.

Yakovlev, S. V. On problems of optimization in the configuration space of spherical objects with variable radius [Текст]: статья / S. V. Yakovlev T. E., S. Shekhovtsov, B. Y. Skrypka – Netherlands., 2017. – 80 p.

Pichugina, O. S. Continuous Representations and Functional Extensions in Combinatorial Optimization [Текст] : учеб. / O. S. Pichugina, S. V. Yakovlev. – M.: Cybernetics and Systems Analysis, 2016. – С. 921 – 930.

Стоян, Ю. Г. Решение некоторых многоэкстремальных задач методом сужающихся окрестностей [Текст] : статья / Ю. Г. Стоян, В. З. Соколовский. – К.: Наук. думка, 1980. – 256 с.

Емеличев, В. А. Многогранники, графы, оптимизация (комбинаторная теория многогранников) [Текст] : учеб. пособие / В. А. Емеличев, М. М. Ковалев, М. К. Кравцов. – М. : Наука, 1981. – 344 с.

Шевченко, В. Н. Линейное и целочисленное программирование [Текст] : учеб. Пособие / В. Н. Шевченко, Н. Ю. Золотых. – Нижний Новгород : Университет, 2004. – 154 с.

Kartashov, O. Some problems of optimization in the configuration space of spherical objects [Текст]: статья / O. Kartashov, K. Korobchynskyi, B. Skrypka – K., 2017. – 168 p.

Нестеров, Ю. Е. Методы выпуклой оптимизации [Текст]: пособие / Ю. Е. Нестеров. – М. : МЦНМО, 2010. – 281 с.

Berge, C. Н. Principles of Combinatorics [Текст] : учеб. пособие / C. Berge. – Toronto: Academic Press, 2012. – 154 p.

Стоян, Ю. Г. Теорія і методі евклідової комбнаторної оптимізації. [Текст] : учеб. пособие / Ю. Г. Стоян, О. О. Єиець. – К. : Ін-т системн. дослідж. освіти, 1993. – 544 с.

Стоян, Ю. Г. Евклидовы комбинаторные конфигурации. [Текст] : монография / Ю. Г. Стоян, С. В. Яковлев. – Х. : Константа, 2017. – 544 с.

Пичугина, О. С. Непрерывные представления в задачах дискретной оптимизации [Текст] : монография / О. С. Пичугина, С. В. Яковлев. – Х. : Университет, 2017. – 357 с.

Яровая, О. В. Полиэдрально-сферические конфигурации и их свойства [Текст] : статья / О. В. Яровая. – Кропивницкий. : Комбінаторні конфігурації та їх застосування, 2018. – С. 155-159.

Яковлев, С. В. Свойства задач комбинаторной оптимизации на полиэдрально-сферических множествах [Текст] : статья / С. В. Яковлев, О. С. Пичугина. – Х. : Кибернетика и системный анализ, 2018. – № 1, – С 97-109.

Pichugina, O.S. Optimization on Polyhedral-Spherical Sets: theory and applications. [Текст] : статья / O. S.Pichugina, S. V. Yakovlev. – P. : IEEE(UKRCON), 2017. – P 1167-1175.

Yakovlev, S. V. Optimization problem on pohydral-spherical configurations. [Текст] : статья / S. V. Yakovlev, O. V. Yarovaya. –Kishinev : Evrica, 2018. – P 237-243.

Yakovlev, S. V. Theory of convex extensions in combinatorial optimization problems [Текст] : статья / S. V. Yakovlev. – K: Report of NAS of Ukraine, 2017. – № 8, – P 20-26.

Яковлев, С. В. Задачи оптимизации на евклидовых комбинаторных конфигурациях и их свойства [Текст] : уч. пособие / С. В. Яковлев, О. С. Пичугина. – Х. : АРК, 2017. – № 17, – С 263-278.

Pichugina, O.S. Continuous Approaches to the Unconstrained Binary Quadratic Problems. [Текст] : статья / O. S.Pichugina, S. V. Yakovlev. – Switzerland : Springer, 2016. – P 689-700.

Pichugina, O.S. Convex extensions and continuous functional representations in optimization. [Текст] : статья / O. S.Pichugina, S. V. Yakovlev. – Montréal : Coupled Syst. Multiscale Dyn., 2016. – № 2(4), – P 129-152.

Pichugina, O.S. Continuous representation techniques in combinatorial optimization. [Текст] : статья / O. S.Pichugina, S. V. Yakovlev. –V. : IOSR Journal of Mathematics, 2017. – № 2(13), – P 12-25.

Пичугина, О.С. Оптимизация на общем множестве перестановок со знаком. [Текст] : статья / О. С. Пичугина. – Х. : Сист. досл. та інф. техн. , 2017. – № 4, – С 74-96.