

Міністерство освіти і науки України
Національний університет «Одеська морська академія»

Кваліфікаційна наукова праця
на правах рукопису

Каменев Кирило Ігорович

УДК 629.123.03

ДИСЕРТАЦІЯ

РОЗРОБКА МЕТОДУ РОЗВ'ЯЗАННЯ ЗАДАЧІ СКЛАДАННЯ ВАНТАЖНОГО ПЛАНУ КОНТЕЙНЕРОВОЗУ З УРАХУВАННЯМ СТРУКТУРНИХ ТА ОПЕРАЦІЙНИХ ОБМЕЖЕНЬ

Спеціальність 271 – Річковий та морський транспорт

Галузь знань 27 – Транспорт

Подається на здобуття наукового ступеня доктора філософії

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


(підпис)

К. І. Каменев

Науковий керівник:

Цимбал Микола Миколайович,
доктор технічних наук, професор

Одеса – 2022

АНОТАЦІЯ

Каменєв К. І. Розробка методу розв'язання задачі складання вантажного плану контейнеровозу з урахуванням структурних та операційних обмежень. – Кваліфікаційна наукова праця на правах рукопису. Дисертація на здобуття наукового ступеня доктора філософії за спеціальністю 271 – Річковий та морський транспорт (галузь знань 27 – Транспорт). – Національний університет «Одеська морська академія» Міністерства освіти і науки України, Одеса, 2022.

Масова контейнеризація перевезень є однією з основних тенденцій, що враховуються національною транспортною стратегією України на період до 2030 року. Однією з її цілей є збільшення частки контейнерних перевезень та стимулювання розвитку контейнерних перевезень.

В той час, коли світовий контейнерний флот продовжує збільшуватися в розмірах і місткості, самі судна витрачають менше часу в портах на обробку контейнерів. Нещодавна пандемія змусила судноплавні компанії застосовувати політику скорочення витрат, яка включає зменшення об'ємів перевезень, призупинення послуг, перенаправлення суден тощо. Такі заходи накладають додатковий тиск на управління вантажними операціями з урахуванням типових щільних графіків контейнерних суден.

Дане дослідження присвячене підвищенню ефективності завантаження контейнерних суден при забезпеченні обмежень щодо морехідної безпеки на етапі підготовки попереднього вантажного плану.

Головною задачею дослідження є розробка методу підготовки вантажного плану контейнерних суден за певних обмежень.

Для її розв'язання були визначені такі **допоміжні задачі**:

1. Розробити математичну модель, яка би враховувала обрані критерії.
2. Обрати та модифікувати алгоритми укладання контейнерів відповідно до обмежень моделі.

3. Оптимізувати отримані розв'язки задачі з розміщення вантажів у трюмі щодо параметрів мореплавства.

Задача розв'язується за допомогою булевої математичної моделі у поєднанні з евристичними та метаевристичними методами, математична модель перевіряється за допомогою використання точного методу.

Наукова новизна дослідження визначається вищезазначеною математичною моделлю та методами, які враховують параметри, не охоплені раніше.

Робота складається із змісту, списку скорочень, вступу, шести розділів, висновку, списку використаної літератури та двох додатків.

У **розділу 1** розглядаються існуючі дослідження щодо морської безпеки та загальні напрямки їх розвитку. Далі розділ містить більш глибокий аналіз математичних моделей та методів, що використовуються дослідниками для розв'язання проблеми розподілу контейнерів на борту, та висвітлює їх переваги та недоліки. Ця проблема має назву MBPP (Master bay plan problem – основна проблема бей-плану), і в даний час є актуальною та часто розглядуваною темою. Існуючі дослідження можна розділити на дві категорії: ті, які намагаються повністю вирішити проблему, та ті, що розділяють її на складові частини.

В дослідженнях першої групи використовуються математичні моделі та алгоритми, що розгалужуються, для досягнення їх цілей. В дослідженнях другої групи задача зазвичай поділяється на дві частини. В першій частині контейнери розподіляються по певних кластерах, а в другій визначаються конкретні позиції контейнерів у таких кластерах.

Розділ 2 присвячений визначенню методології дослідження. В ньому визначається вибір напрямку дослідження, формулюється основне завдання та розподіл його на допоміжні завдання. У розділу 2 також перелічені використовувані методи.

Розділ 3 присвячений вищезазначеній математичній моделі. В ньому аналізуються структурні та експлуатаційні обмеження суден та контейнерів, а також вимоги до сегрегації небезпечних вантажів. У цьому розділі представлена

гнучка булева математична модель, яка складається з восьми нерівностей, що охоплюють визначені операційні та структурні обмеження і забезпечують наступне:

1. Усі необхідні контейнери завантажуються з урахуванням їх видів та типів вантажів.
2. Жодні два контейнери не займають одне і те ж місце.
3. FEU (forty-foot equivalent unit) контейнери займають два TEU (twenty-foot equivalent unit) слоти у вантажному просторі.
4. TEU контейнери не можна завантажувати поверх FEU контейнерів.
5. FEU контейнери не можна завантажувати поверх двох стеків TEU різної висоти.
6. Як TEU, так і FEU контейнери, можуть бути розміщені тільки над іншими контейнерами, або на палубі судна; вони не можуть парити у повітрі.
7. Контейнери, які містять небезпечні вантажі, охоплені Кодексом IMDG (International maritime dangerous goods code), належним чином сегреговані відповідно до правил.

Після визначення моделі пропонується метод перетворення обмежень моделі в матричну форму. Метод заснований на модифікованому алгоритмі сортувальної станції Е. Дейкстри та враховує конкретні значення параметрів моделі, такі як розміри судна, кількість контейнерів та ін.

Заключна частина розділу присвячена перевірці моделі за допомогою точного методу, заснованого на методі гілок та границь. Чисельний експеримент навмисно ускладняється шляхом штучного вибору ситуації з жорсткими обмеженнями, яка вимагає дуже лімітовані позиції розміщення контейнерів.

У розділу 4 представлений метод розв'язання розглядуваної задачі виконання обмежень. Після оцінювання існуючих алгоритмів для виконання завдання було обрано генетичний алгоритм.

Запропоновано спосіб кодування розв'язків для ефективного виконання схрещення та селекції.

Пропонується фітнес-функція, яка враховує загальну кількість неправильно розміщених контейнерів для кожної нерівності моделі.

Також запропоновано функцію схрещення, яка використовує повний набір координат розв'язків, що через специфіку задачі дозволяє уникати розв'язків, що не мають практичного сенсу.

Використання турнірного відбору зумовлено недоліками інших методів при розв'язанні поставленої задачі.

Враховуючи всі вищезазначені функції, пропонується генетичний алгоритм стаціонарного стану для розв'язання поставленої задачі виконання обмежень.

Розділ 5 присвячений аналізу морехідних властивостей судна, який враховує вимоги до параметрів остійності, передбачені ІМО (Міжнародна морська організація), місцевої та поздовжньої міцності, диференту та крену.

У цьому розділі представлено розроблений метод, заснований на генетичному алгоритмі, який дозволяє оптимізувати параметри морехідності. Для того, щоб розв'язок залишався незмінним щодо обмежень математичної моделі, алгоритм виконує лише обмін контейнерів з однаковими параметрами. Пропонується гнучка функція допасованості для вибору найбільш придатних розв'язків у залежності від конкретних вимог рейсу.

З метою мінімізації витрат часу, необхідного для розміщення контейнерів за алгоритмом, описаним у розділі 4, був розроблений метод початкового розміщення контейнерів. З цією метою була розроблена функція оцінки придатності доступних позицій для кожного нерозташованого контейнеру, яка враховує як обмеження моделі, так і параметри мореплавства.

Розділ 6 присвячений імітаційному моделюванню для аналізу результатів попередніх розділів. Імітаційне моделювання виконується з використанням реальних судових даних, що дозволяє перевірити практичне застосування розробленої моделі та розроблених методів.

Спочатку виконується послідовне завантаження 4 партій контейнерів, призначених для 4 портів, оптимізуючи кожен крок для досягнення бажаних

параметрів мореплавства. Експеримент показав ефективність методів, представлених у розділі 5.

Далі було обрано конкретний випадок, що вимагав використання алгоритму, описаного в розділі 4 даної роботи. Використання розробленого алгоритму показало задовільні (відповідно до математичної моделі) результати, що підтверджують придатність моделі та методу.

Нарешті, новий розв'язок було оптимізовано, надалі підтверджуючи ефективність алгоритму, представленого в розділі 5.

Розв'язання основних та допоміжних задач призвело до наукової позиції, яку можна сформулювати таким чином: підготовка плану укладання контейнерних суден може бути вдосконалена за допомогою уточненої математичної моделі.

Досягнуто наступні наукові результати:

- Розроблено вдосконалену булеву математичну модель, що охоплює структурні та експлуатаційні обмеження;
- Запропоновано метод розв'язання розглядуваної задачі виконання обмежень щодо розміщення контейнерів на борту судна;
- Розроблено метод оптимізації розміщення контейнерів щодо параметрів мореплавства, який використовує вагу, порт призначення, тип і клас небезпечного вантажу кожного окремого контейнера та відрізняється можливістю використовувати гнучку цільову функцію, яка може модифікуватися щодо окремих параметрів мореплавства у залежності від особливостей конкретного рейсу;
- Запропоновано метод початкового розміщення контейнерів з урахуванням структурних та експлуатаційних обмежень, а також параметрів мореплавства.

Практична цінність дослідження полягає у потенційному використанні результатів для складання вантажного плану контейнерних суден, при розробці програмного забезпечення для завантаження контейнерних суден, у навчальному процесі та для підвищення кваліфікації штурманів.

Ключові слова: контейнерне судно, вантажний план, оптимізація вантажного плану, булева математична модель, небезпечні вантажі, генетичний алгоритм.

Список публікацій за темою

Статті у наукових виданнях, включених до переліку наукових фахових видань України:

1. Камєнєва А. В., Камєнєв К. І. Розробка вантажної програми для контейнерного судна з урахуванням вимог Міжнародного морського кодексу з небезпечних вантажів // Наукові праці: науково-методичний журнал. - Вип. 296. Том 308 «Комп'ютерні технології». – Миколаїв: Вид-во ЧНУ ім. П. Могили, 2017, с. 94-99.
2. Камєнєва А. В., Камєнєв К. І. Використання адитивного алгоритму для розміщення небезпечних вантажів на контейнерному судні // Судовождение: сборник научных трудов. – 2018. – Вып. 28. – Одеса: НУ «ОМА», центр «Видавінформ», - с. 70-77.
3. Камєнєв К. І., Камєнєва А.В., Цимбал М. М. Розробка математичної моделі для задачі бей-плану з урахуванням черговості завантаження контейнерів // Судноводіння: науково-технічний збірник. – 2021. – Вып. 32. – Одеса: НУ «ОМА», центр «Видавінформ», – с. 34-45. DOI: 10.31653/2306-5761.32.2021.34-45

Статті у виданнях, проіндексованих у міжнародних наукометричних базах даних SCOPUS та Web of Science:

4. Kamieniev K. Construction of a mathematical model and a method for arranging hazardous cargoes on a containership / K. Kamieniev, A. Kamienieva, M. Tsymbal // Eastern-European Journal of Enterprise Technologies. – 2019. – Vol. 6. – № 3 (102). – p. 20-27. DOI: 10.15587/1729-4061.2019.183385.
5. M. Tsymbal, K. Kamieniev. Modified Integer Model for Solving the Master Bay Problem // the International Journal on Marine Navigation and Safety of Sea Transportation. – 2021. – Vol. 15 No. 4. – Akademia Morska w Gdyni, – p. 749-753. DOI: 10.12716/1001.15.04.05.

Статті у інших іноземних виданнях:

6. Kamieniev K. Automation of constraints creation for master bay plan problem / K. Kamieniev, A. Kamienieva // Slovak international scientific journal. – 2020. – Vol. 2 (45). – p. 19-23.

7. Каменев К.И. Подготовка входных данных для решения основной проблемы бэй-плана. // Science and Education a New Dimension. Natural and Technical Sciences, VIII (29), Issue: 238, 2020. - с. 71 - 73.

Статті, опубліковані у збірниках матеріалів наукових конференцій:

8. Kamieniev K. Relative container layout constrictions for automated stowage planning / K. Kamieniev, E. Strakhov // 9th International conference “Science and society”. Hamilton, Canada – 2019. – p. 1048-1052.

ABSTRACT

Kamieniev K. I. Method development for solving the task of preparing a containership stowage plan considering structural and operational constraints. – Manuscript.

Thesis for a Philosophy doctor degree (PhD) in specialty 271 – River and sea transport (branch of knowledge 27 – Transport) – National University “Odessa Maritime Academy” of the Ministry of Education and Science of Ukraine, Odesa, 2022.

Global containerization of transportations is one of the main tendencies taken into account by the Ukrainian national transport strategy for the period until 2030. One of its aims is to increase container transportation volumes and stimulate development of container transport.

As the world's container fleet continues to grow in size and capacity, the ships themselves spend less time in ports per container. The recent pandemic caused shipping companies to apply cost cutting policies, which include reducing capacity, suspending services, re-routing vessels and similar. Such measures apply additional pressure to cargo operations management considering the typical tight schedules of containerships.

The research is devoted to improving maritime shipping efficiency while satisfying safety constraints via optimizing the preloading stowage plan preparation.

The main task of the research is to develop a method for preparing a containership stowage plan under specified constraints. In order to solve it, it has been broken into the following **auxiliary tasks**:

1. To develop a mathematical model that would take into account chosen criteria.
2. To select and modify algorithms for stowing containers under the model's constraints.
3. To optimize the model's solutions for seaworthiness parameters.

The task is accomplished using a Boolean mathematical model in conjunction with heuristic and metaheuristic methods, the model is then checked using an exact method.

The **scientific novelty** in the research is determined by the mathematical model and methods mentioned above which consider several parameters not covered before.

The thesis consists of a table of contents, list of abbreviations, introduction, six chapters, conclusion, list of references and two annexes.

Chapter 1 examines existing research regarding maritime safety and general directions of its development. Furthermore, it contains a deeper analysis of the mathematical models and methods used by researchers for solving the problem of allocating containers onboard and highlights their advantages and flaws. The problem has been named MBPP (Master bay plan problem) and it is currently highly studied topic. Existing studies can be divided into two categories: those trying to solve the problem in full and those dividing it into smaller tasks.

The first category of studies uses mathematical models and branching algorithms to reach their targets. The second category studies are normally separated into two groups. The first group allocates sets of containers inside certain clusters while the second determines specific container positions within such clusters.

Chapter 2 focuses on determining the research's methodology. It includes selecting the research direction, formulating the main task and dividing it into auxiliary tasks. It also lists the methods used.

Chapter 3 is devoted to the abovementioned mathematical model. It analyses structural and operational constraints of vessels and containers and also segregation requirements for dangerous goods. The chapter presents a flexible Boolean mathematical model consisting of eight inequalities covering those operational and structural constraints that ensure the following:

1. All the necessary containers have been loaded considering their types and kind of goods.
2. No two containers occupy the same spot.
3. FEU (forty-foot equivalent unit) containers occupy two TEU (twenty-foot equivalent unit) slots in the cargo space.
4. TEU containers cannot be loaded on top of FEU containers.
5. FEU containers cannot be loaded on top of two TEU stacks of different heights.

6. Both TEU and FEU containers can only be located either on top of other containers or the ship's deck; they can't be suspended in air.
7. Containers containing dangerous goods covered by IMDG Code are properly segregated according to regulations.

After specifying the model, a method for transforming it to a matrix form is proposed. The method is based on a modified shunting-yard algorithm and considers specific values of the model's parameters, such as a ship's dimensions or containers numbers.

The final part of the chapter is devoted to checking the model using an exact method based on a branch-and-bound algorithm. The numerical experiment is specifically complicated by artificially selecting a situation with strict constraints dictating very limited container placement options.

Chapter 4 presents a method for solving the constraint satisfaction problem. After weighing available options, a modified genetic algorithm was selected to perform the task.

A solution coding method is presented which allows to effectively perform crossover and selection.

A fitness function is proposed that takes into account the total number of wrongly placed containers per inequality.

A crossover function is proposed that uses full sets of solutions' coordinates which, because of the task specifics, cuts nonsensical solutions.

A tournament selection is used due to other methods' disadvantages for the task.

Considering all of the abovementioned functions a steady-state genetic algorithm is presented for solving the constraint satisfaction problem.

Chapter 5 is devoted to analyzing a vessel's seaworthiness. It considers stability parameters requirements mandated by the IMO (International Maritime Organization), local and longitudinal strength, trim and heel.

A method based on a genetic algorithm is developed allowing to optimize a solution's seaworthiness parameters. In order to keep the solution unchanged regarding the mathematical model, the algorithm operates via only swapping containers with the

same parameters. A flexible fitness function is proposed for selecting the fittest solutions based on specific voyage requirements.

In order to minimize the time needed for placing the containers by the algorithm described in Chapter 4, a method of initial container placement was developed. For that purpose, a function assessing available positions' fitness per unlocated container was developed which takes into account both the model's requirements and seaworthiness parameters.

Chapter 6 is devoted to simulation modelling for analyzing the previous chapters' results. It is performed using a real ship's data which checks the developed model's and methods' practical application.

First a consecutive loading of 4 container consignments destined to 4 ports is performed, optimizing each step to achieve the desired seaworthiness parameters. The experiment showed effectiveness of the methods presented in Chapter 5.

After, a specific case was selected requiring usage of algorithm described in Chapter 4 of the thesis. Using the algorithm showed satisfactory (as per mathematical model) results confirming the model's and the methods' applicability.

Finally, the effectiveness of the algorithm presented in Chapter 5 was further confirmed by optimizing the new solution.

Solving the main and auxiliary tasks has resulted in a **scientific position** which can be formulated as the following: preparing a containership stowage plan can be improved using a refined mathematical model.

The following **scientific results** have been achieved:

- The development of an improved Boolean mathematical model covering structural and operational constraints
- A method for solving the container placement constraint satisfaction problem has been proposed
- A proposal of a method for further container placement optimization considering seaworthiness parameters has been developed
- A method for initial containers placement considering structural and operational restrictions as well as seaworthiness parameters has been proposed

The **practical value** of the study consists of its results' potential usage in stowage planning optimization, developing containership loading software, educational process and for advanced training of navigators.

Keywords: containership, stowage planning, stowage plan optimization, Boolean mathematical model, dangerous goods, genetic algorithm.

List of published works on the topic

Articles in scientific professional publications, listed in the list of scientific professional publications of Ukraine:

1. Камєнєва А.В., Камєнєв К. І. Розробка вантажної програми для контейнерного судна з урахуванням вимог Міжнародного морського кодексу з небезпечних вантажів // Наукові праці: науково-методичний журнал. - Вип. 296. Том 308 «Комп'ютерні технології». – Миколаїв: Вид-во ЧНУ ім. П. Могили, 2017, с. 94-99.
2. Камєнєва А.В., Камєнєв К. І. Використання адитивного алгоритму для розміщення небезпечних вантажів на контейнерному судні // Судовождение: сборник научных трудов. – 2018. – Вып. 28. – Одеса: НУ «ОМА», центр «Видавінформ», - с. 70-77.
3. Камєнєв К. І., Камєнєва А.В., Цимбал М. М. Розробка математичної моделі для задачі бей-плану з урахуванням черговості завантаження контейнерів // Судноводіння: науково-технічний збірник. – 2021. – Вып. 32. – Одеса: НУ «ОМА», центр «Видавінформ», – с. 34-45. DOI: 10.31653/2306-5761.32.2021.34-45.

Articles in publications indexed in SCOPUS and Web of Science databases:

4. Kamieniev K. Construction of a mathematical model and a method for arranging hazardous cargoes on a containership / K. Kamieniev, A. Kamienieva, M. Tsymbal // Eastern-European Journal of Enterprise Technologies. – 2019. – Vol. 6. – № 3 (102). – p. 20-27. doi: 10.15587/1729-4061.2019.183385.
5. M. Tsymbal, K. Kamieniev. Modified Integer Model for Solving the Master Bay Problem // the International Journal on Marine Navigation and Safety of Sea Transportation. – 2021. – Vol. 15 No. 4. – Akademia Morska w Gdyni, – p. 749-753. DOI: 10.12716/1001.15.04.05.

Articles in other foreign publications:

6. Kamieniev K. Automation of constraints creation for master bay plan problem / K. Kamieniev, A. Kamienieva // Slovak international scientific journal. – 2020. – Vol. 2 (45). – p. 19-23.
7. Каменев К.И. Подготовка входных данных для решения основной проблемы бэй-плана. // Science and Education a New Dimension. Natural and Technical Sciences, VIII (29), Issue: 238, 2020. - с. 71 - 73.

Conference proceedings:

8. Kamieniev K. Relative container layout constrictions for automated stowage planning / K. Kamieniev, E. Strakhov // 9th International conference “Science and society”. Hamilton, Canada – 2019. – p. 1048-1052.

ЗМІСТ

ПЕРЕЛІК ПРИЙНЯТИХ СКОРОЧЕНЬ.....	17
ВСТУП.....	18
РОЗДІЛ 1. АНАЛІЗ ЛІТЕРАТУРНИХ ДЖЕРЕЛ З ПРОБЛЕМИ ЗАБЕЗПЕЧЕННЯ БЕЗПЕКИ МОРЕПЛАВСТВА	23
1.1 Основні напрямки підвищення безпеки судноплавства	23
1.2 Постановка задачі завантаження контейнеровозу та напрямки її розв’язання.....	27
1.3 Висновки за розділом 1	36
РОЗДІЛ 2. МЕТОДОЛОГІЧНЕ ЗАБЕЗПЕЧЕННЯ НАУКОВОГО ДОСЛІДЖЕННЯ	38
2.1 Вибір теми дослідження	38
2.2 Методи дисертаційного дослідження	39
2.3 Методика проведення дисертаційного дослідження.....	43
2.4 Висновки за розділом 2	44
РОЗДІЛ 3. РОЗРОБКА МАТЕМАТИЧНОЇ МОДЕЛІ ОСНОВНОЇ ПРОБЛЕМИ БЕЙ ПЛАНУ	45
3.1 Вимоги щодо розміщення контейнерів з небезпечними вантажами .	45
3.2 Математична модель.....	52
3.3 Формування обмежень	55
3.4 Перевірка адекватності моделі	60
3.5 Висновки за розділом 3	65
РОЗДІЛ 4. РОЗРОБКА МЕТОДУ РОЗВ’ЯЗАННЯ ЗАДАЧІ ВИКОНАННЯ ОБМЕЖЕНЬ	66
4.1 Вибір методу.....	66

4.2	Модифікація методу	68
4.3	Висновки за розділом 4	81
РОЗДІЛ 5. ВДОСКОНАЛЕННЯ РОЗТАШУВАННЯ КОНТЕЙНЕРІВ ЩОДО ПАРАМЕТРІВ МОРЕХІДНОСТІ.....		83
5.1	Остійність.....	83
5.2	Диферент та міцність	84
5.3	Крен	86
5.4	Модифікація розв’язку з урахуванням морехідних якостей	86
5.5	Початкове розташування контейнерів	95
5.6	Висновки за розділом 5	98
РОЗДІЛ 6. ОЦІНКА ЕФЕКТИВНОСТІ РОЗРОБЛЕНИХ МЕТОДІВ ЗА ДОПОМОГОЮ ІМІТАЦІЙНОГО МОДЕЛЮВАННЯ ЗАВАНТАЖЕННЯ.....		100
6.1	Початкове розташування та подальша оптимізація завантаження..	100
6.2	Окремий випадок перевірки моделі	112
6.3	Висновки за розділом 6	116
ВИСНОВКИ.....		118
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....		120
ДОДАТКИ.....		133

ПЕРЕЛІК ПРИЙНЯТИХ СКОРОЧЕНЬ

ACO – Ant Colony Optimization / мурашиний алгоритм

FEU – forty-foot equivalent unit / 40-футовий стандартний контейнер

GA – Genetic Algorithm / генетичний алгоритм

IMDG Code – The International Maritime Dangerous Goods Code / Міжнародний кодекс морського перевезення небезпечних вантажів

IMO – International maritime organization / Міжнародна морська організація

MBPP – Master Bay Plan Problem / основна проблема бей плану

PSO – Particle Swarm Optimization / метод рою часток

SA – Simulated Annealing / алгоритм імітації відпалу

SOLAS – Safety of life at sea / Міжнародна конвенція з охорони людського

життя

на морі

TEU – twenty-foot equivalent unit / 20-футовий стандартний контейнер

TS – Tabu Search / алгоритм табу пошуку

UNN – United Nations number / номер ООН для небезпечних вантажів

ВСТУП

Актуальність теми

Задача визначення розташування контейнерів різних типів у обмеженому вантажному просторі контейнерного судна при врахуванні операційних та структурних обмежень як контейнерів, так і самого судна отримала назву основної проблеми бей плану (MBPP – Master Bay Plan Problem) та є часто розглядуваною у сучасній літературі.

Складність задачі зумовлена великою кількістю параметрів, які залежать від кожного окремого контейнеру. При цьому розміщення контейнерів впливає не лише на безпеку судноплавства, але й на ефективність роботи суден. Уникнення перестановок контейнерів дозволяє зменшити витрати у портах, а оптимізація за диферентом – зменшити витрати палива.

Масова контейнеризація перевезень є однією з основних тенденцій, що враховуються національною транспортною стратегією України на період до 2030 року [1]. Однією з її цілей є збільшення частки контейнерних перевезень та стимулювання розвитку контейнерних перевезень.

В той час, коли світовий контейнерний флот продовжує збільшуватися в розмірах і місткості, самі судна проводять менше часу в портах на завантаження та вивантаження контейнерів. Нещодавна пандемія змусила судноплавні компанії застосовувати політику скорочення витрат, яка включає зменшення об'ємів перевезень, призупинення послуг, перенаправлення суден тощо. Такі заходи накладають додатковий тиск на управління вантажними операціями з урахуванням типових щільних графіків контейнерних суден.

Згідно з оглядом морського транспорту за 2020 рік організації об'єднаних націй (Review of maritime transport, 2020) [2] світовий комерційний флот зріс на 4.1%, що є найвищим показником росту з 2014 року. У той самий час розмір найбільшого контейнерного судна виріс на 10.9%. При цьому через пандемію оператори почали впроваджувати заходи по зменшенню об'ємів перевезень та

зменшення витрат для запобігання падіння фрахтових ставок. У квітні 2020 10% світового контейнерного флоту (за місткістю) були у простої.

У другій половині 2020 року об'єм перевезень почав зростати, але довгострокові перспективи не є ясними.

Постійне зростання розміру контейнерних суден та економічні заходи, які приймають компанії через пандемію, ускладнюють складання вантажного плану, який буде забезпечувати морехідність судна.

Звідси вибір оптимального завантаження контейнерних суден при забезпеченні обмежень щодо їх безпеки є **актуальною проблемою** у сучасних умовах.

Зв'язок роботи з науковими програмами, планами, темами.

Робота виконувалась у рамках наукових досліджень національного університету «Одеська морська академія» за держбюджетною темою «Розвиток інтегрованої системи функціонування спеціалізованого флоту в умовах нестабільності» (№ ДР0117U000314, 2017 - 2021рр.), в якій здобувачем було виконано окремий підрозділ.

Додатково, при виконанні роботи було використано положення Національної транспортної стратегії України на період до 2030 року (розп. КМУ № 430-р від 30 травня 2018 р.).

Метою дослідження є підвищення ефективності завантаження контейнерних суден при забезпеченні обмежень щодо їх морехідної безпеки.

Наукове завдання полягає у розробці методів та моделей складання вантажного плану контейнерного судна для підвищення ефективності їх завантаження при забезпеченні обмежень щодо морехідної безпеки.

Часткові завдання дослідження, які були вирішені для розв'язання головного:

- розроблено математичну модель судна з урахуванням обраних критеріїв;
- обґрунтовано вибір і модифікацію алгоритмів для розташування вантажів у відповідності до розробленої моделі;

- вдосконалення процедур розташування вантажів для покращення морехідних якостей.

Об'єктом дослідження роботи є процеси забезпечення морехідної безпеки контейнерних суден.

Предметом дослідження є методи та моделі завантаження контейнерних суден.

Методи дослідження:

- дедукції – при здійсненні інформаційного пошуку;
- системного аналізу – для визначення тематики дослідження та методологічного забезпечення;
- математичного моделювання – для розробки моделі та проведення імітаційного моделювання;
- математичного програмування – для оптимізації завантаження контейнерів на судно.

Наукова новизна дослідження визначається розробленою математичною моделлю та методами, які враховують параметри, не охоплені раніше.

У дисертаційній роботі:

- вперше розроблено вдосконалену булеву математичну модель, що охоплює структурні та експлуатаційні обмеження;
- вперше запропоновано метод розв'язання розглядуваної задачі виконання обмежень розміщення контейнерів на борту судна, який заснований на генетичному алгоритмі та враховує обмеження математичної моделі;
- отримали подальший розвиток методи оптимізації розміщення контейнерів щодо параметрів мореплавства при завантаженні, який використовує вагу, порт призначення, тип і клас небезпечного вантажу кожного окремого контейнера та відрізняється можливістю використовувати гнучку цільову функцію, яка може модифікуватися щодо окремих параметрів мореплавства у залежності від особливостей конкретного рейсу;

- вперше запропоновано метод початкового розміщення контейнерів з урахуванням структурних та експлуатаційних обмежень, а також параметрів мореплавства, який заснований на сортуванні списку контейнерів, які необхідно завантажити, за зазначеними параметрами та гнучкій функції, що по чергово оцінює допасованість окремого контейнеру для розміщення у кожному доступному слоті.

Практичне значення отриманих результатів: можливе використання результатів для складання вантажного плану контейнерних суден, розробленні програмного забезпечення для завантаження контейнерних суден, у навчальному процесі та для підвищення кваліфікації штурманів.

Матеріали приведеного дисертаційного дослідження було впроваджено у навчальний процес при викладанні дисципліни «Морські перевезення» (акт від 17.05.2021).

Особистий внесок здобувача: дисертаційна робота була виконана самостійно; був проведений інформаційний пошук та аналіз літературних джерел, виконана розробка математичної моделі, перевірка її адекватності, вибір та розробка методу розв'язання задачі виконання обмежень, вибір та розробка методів оптимізації щодо морехідних якостей та початкового розміщення, та проведення імітаційного моделювання завантаження судна для їх перевірки.

З наукових праць, опублікованих у співавторстві, використано виключно ті положення, що належать автору особисто: розробка математичної моделі та методу перевірки її адекватності [3], програмне забезпечення та методи перевірки сегрегації [4] та метод розташування контейнерів [5].

Апробація результатів роботи. Основні результати і положення дослідження доповідались на ряді конференцій, зокрема:

- науково-технічна конференція «Транспортні технології (морський та річковий флот): інфраструктура, судноплавство, перевезення, автоматизація» (Одеса, 16-17 листопада 2017 р.);

- міжнародна науково-технічна конференція «Суднові комп'ютерно-інтегровані технології», (Одеса, 08-09 листопада 2018 р.):
- 9th International conference «Science and society», (Hamilton, February 1st, 2019);
- international scientific and professional conference «Science without boundarie development in 21st century – 2020 », (Budapest, August 30th, 2020).

Публікації. За темою дисертаційного дослідження було опубліковано 8 наукових праць, з яких 3 – у наукових виданнях, включених до переліку наукових фахових видань України, 2 – у виданнях, включених до наукометричної бази SCOPUS, 3 – у інших іноземних виданнях (2 – у наукових журналах та 1 – у збірці матеріалів наукової конференції).

Структура роботи. Дисертація складається зі змісту, списку скорочень, вступу, шести розділів, висновку, списку використаних джерел (131 найменування) та двох додатків. Загальний обсяг роботи становить 143 сторінки, зокрема основний текст з анотацією – 120 сторінок, список використаних джерел – 13 сторінок, додатки – 10 сторінок; робота містить 33 рисунки та 12 таблиць.

РОЗДІЛ 1. АНАЛІЗ ЛІТЕРАТУРНИХ ДЖЕРЕЛ З ПРОБЛЕМИ ЗАБЕЗПЕЧЕННЯ БЕЗПЕКИ МОРЕПЛАВСТВА

1.1 Основні напрямки підвищення безпеки судноплавства

Проблема забезпечення та підвищення безпеки судноплавства розглядається багатьма вітчизняними та зарубіжними авторами. Для цього проводяться дослідження за різними напрямками, що розглянуті нижче.

У роботах [6]–[11] розглянуті основні питання щодо забезпечення остійності, міцності, плавучості та посадки суден.

Роботи [12], [13] більш детально розглядають остійність судна, а саме остійність при великих кутах нахилу та методи контролю остійності у залежності від особливостей процесу хвилювання.

У [14] досліджується залежність ДСО (діаграми статичної остійності) судна від форми його корпусу.

Стаття [15] розглядає альтернативні вимоги до остійності суден та розраховує критерій погоди щодо них.

Робота [16] пропонує метод розрахунку ймовірності збереження остійності пошкодженого судна.

У [17] проводиться аналіз впливу залежності остійності судна від циркуляції води у пошкодженому відсіку.

Робота [18] проводить аналіз впливу проектних характеристик судна на його параметри морехідності та міцності.

Більш детальний огляд місцевої міцності корпусу судна та метод розрахунку коефіцієнту запасу для неї представлений у [19].

Автори [20] пропонують метод оцінки місцевої міцності корпусу, що оснований на даних з листових конструкцій.

У [21] та [22] розглядаються питання нормування міцності суден.

Автор [23] розглядає скручування корпусів суден та розрахунок напруг і деформацій при ньому.

У статті [24] розглядається вплив циклічних навантажень на конструктивні вузли корпусу судна.

Робота [25] аналізує напруги при поперечному вигині та моделі корпусів суден, що можна використовувати при комп'ютерних обчисленнях.

Стаття [26] розглядає граничний згинаючий момент корпусу і його залежність від початкового прогину поздовжніх ребр та поперечного навантаження.

У роботі [27] розглядаються залишкові деформації ребр жорсткості палуби, які викликані використанням грейферів.

Автор [28] пропонує методи розрахунків конструкційних елементів судна в умовах арктичного плавання.

Публікація [29] проводить порівняльний аналіз чисельних методів, що використовуються для оцінки міцності судна.

Робота [30] презентує систему розрахунку напруг для великотоннажних танкерів.

Автори [31] розглядають міцність поздовжніх елементів корпусу судна та метод розрахунку діаграми навантажень відносно початкових деформацій.

Стаття [32] розглядає нормативні вимоги щодо контролю завантаження та пропонує спрощену методику визначення згинаючих моментів та перерізуючих сил.

У роботі [33] розглядається розрахунок згинаючого моменту пошкодженого судна та його перегин.

Автор [34] розглядає розрахунок непотоплюваності судна як нелінійну просторову задачу.

У [35] розглядається боротьба за непотоплюваність за допомогою динамічної моделі щодо зміни небезпечних чинників.

Стаття [36] розглядає остійність RO-RO (roll on – roll off, накатні) суден при пошкодженнях за допомогою симуляції.

Робота [37] розглядає кутові прискорення, що виникають при бортовій хитавиці.

Також питання хитавиці судна розглянуто у [38], де відбувається аналіз кутових прискорень, і у [39], де розглядається залежність сил інерції від моменту щодо повздовжньої осі.

Автори [40] пропонують метод розрахунку параметрів статичної посадки суден при хитавиці, що дозволяє підвищити точність результатів.

Робота [41] розглядає вплив хвильової хитавиці на згинаючий момент під час шторму.

У статті [42] пропонується система контролю за безпекою суден під час шторму, що враховує плавучість та остійність.

У роботі [43] приведено умови перекидання судна при попутній хитавиці в умовах параметричного резонансу.

Публікація [44] розглядає безпеку перевезення вантажів з урахуванням хитавиці, а [45] – вплив завантаження судна на величину сил інерції при хитавиці.

Робота [46] пропонує спосіб визначення максимального значення горизонтальної складової сил інерції, що під час хитавиці діє на вантаж.

У [47] розглянуто сили інерції, що виникають під час переходу, та їх вплив на завантаження судна.

Автор [48] аналізує фактори, що впливають на аварійність невеликих суден, та пропонує напрями зниження їх впливу.

У статті [49] розглядається проблема пошкодження суден при швартуванні.

У [50] проводиться статистичний аналіз пошкоджень суден при експлуатації.

Робота [51] присвячена ідентифікації та зниженню ризиків при проведенні зміни баласту у морі для існуючих суден на основі їх проектів.

Автори [52] представили алгоритм розрахунку навантажень на кріплення вантажів та рекомендації щодо видів завантаження.

У [53] розглядається визначення водотоннажності та осадок на великих диферентах за допомогою модифікування вантажної шкали, а саме доповнення її поправками на диферент.

Автор [54] розглядає вплив гнучкості корпусу суден при розрахунках посадки та вигину при виконанні попереднього завантаження за допомогою комп'ютерних програм.

Стаття [55] присвячена підвищенню точності при розрахунках кількості вантажу та водотоннажності за осадками, враховуючи вигин корпусу та крен.

Автор [56] розглядає математичні моделі для оптимізації розрахунків при проектуванні суден.

Робота [57] присвячена оптимізації завантаження транспортних засобів, що здійснюють комерційні перевозки.

Стаття [58] розглядає автоматизацію розрахунку оптимального вантажного плану та оцінку непотоплюваності суден на судових комп'ютерах.

У роботі [59] аналізуються стратегії проведення вантажних операцій для навальних суден та питання розрахунку їх множини, а у [60] формується оптимізаційна задача проведення вантажних операцій на таких суднах.

Стаття [61] також розглядає проведення вантажних операцій на навальних суднах та аналізує вибір оптимального варіанту для їх проведення.

Автор [62] розглядає окремі питання оптимізації вантажних операцій для навальних суден.

У роботах [63]–[67] розглядається проблема завантаження генеральних вантажів, розподілу таких вантажів по трюмах, їх укладання у межах одного трюму, вибір партій генеральних вантажів, тощо.

Стаття [68] пропонує стратегію завантаження генеральними вантажами, [69] – процедуру вибору їх типів при укладці до вантажних приміщень, а [70] присвячена моделюванню укладки генеральних вантажів до вантажних приміщень.

Автор [71] аналізує проблему забезпечення безпечного перевезення цементу насипом на навальних суднах.

У [72] пропонується метод формування розробки вантажної програми судна, беручи до уваги критерії морехідності, а у [73] – формування вантажної програми з використанням бази даних.

Стаття [74] описує метод введення характеристик судна до бази даних.

Робота [75] присвячена обробці даних танків судна для подальшого введення їх до бази даних.

У публікації [76] розглядається програмний опис стандартних судових приміщень, а [77] присвячена судовим приміщенням нестандартного типу.

Автор [78] приводить аналіз залежностей параметрів міцності, остійності та посадки від початкових даних судна.

Публікація [79] присвячена відображенню параметрів морехідності у вантажній програмі судна, а [80] – відображенню укладки генеральних вантажів по трюмах.

Стаття [81] розглядає імітаційне моделювання завантаження контейнеровозу.

Робота [82] присвячена завантаженню контейнерних суден з урахуванням ротації портів, а [83] – безпечному завантаженню, що буде задовольняти усім вимогам морехідності.

1.2 Постановка задачі завантаження контейнеровозу та напрямки її розв'язання

Детальний опис основної проблеми бей плану (MBPP) приведено Данієлою Амброзіно у статті «Stowing a containership: The master bay plan problem» [84]. Він полягає у визначенні розташування контейнерів різних типів у обмеженому вантажному просторі контейнерного судна при врахуванні структурних та операційних обмежень як контейнерів, так і самого судна.

Автор пропонує спрощену модель лінійного програмування, яка вирішується шляхом розподілення задачі на 3 етапи.

Перший етап, «Попереднє опрацювання», полягає у виключенні положень контейнерів, які не задовольняють строго заданих умов. Береться множина усіх можливих положень контейнерів, і із неї виключаються такі, в яких заданий контейнер однозначно не може бути розміщений. Наприклад, для реф-контейнерів виключаються такі положення, де не може бути забезпечення їх живлення; для контейнерів з деякими видами небезпечних вантажів – місця на палубі і т. д. Таким чином з початкової множини отримується підмножина, що менше першої.

Другий етап має назву «Процедура попереднього завантаження», і на ньому відбувається розподіл контейнерів згідно портів вивантаження. Зазначена модель поділяє вантажі, призначені у різні порти на окремі беї, при цьому порядок завантаження запланований таким чином, що контейнери, спрямовані до перших портів вивантаження, розташовуються ближче до центру судна. Наприкінці цього етапу із вищезазначеної підмножини отримується ще менша підмножина, що задовольняє умови по портах/беях.

На третьому етапі виконується розв'язання системи лінійних рівнянь з урахуванням підмножини, отриманої у результаті перших двох етапів.

Дана модель не враховує параметрів остійності або міцності, а замість них використовується система спрощень, таких як: більш легкі контейнери вантажаться поверх більш тяжких, вага контейнерів, розміщених у носовій частині судна повинен рівнятися вазі контейнерів, розміщених у кормовій частині (з урахуванням деякої похибки), вага контейнерів з лівого та правого борту має бути однаковою та ін. Очевидно, що дані припущення не охоплюють усі можливі випадки і можуть викликати невірні розв'язки.

У статті [85] «A 3D-BPP approach for optimising stowage plans and terminal productivity» судно розглядається як 3-D корзина, а завантаження поділяється на три етапи, при цьому вимоги щодо небезпечних або рефрижераторних контейнерів не перевіряються. При цьому автори намагаються забезпечити найшвидше завантаження/вивантаження контейнерів за допомогою розподілення контейнерів таким чином, щоб їх можна було паралельно вивантажувати декількома кранами.

На першому етапі пропонується розподілити контейнери між беями та береговими кранами, вважаючи, що мінімальна відстань між двома працюючими кранами становить два бею.

На другому етапі судно розподіляється на секції/корзини, що відділяють трюми, носову та кормову частини, головну палубу, а також окремі місця, куди контейнери завантажувати не можна.

На третьому етапі автори розподіляють контейнери за корзинами, у залежності від їх ваги; розглядаються легкі (до 15 тон), середні (до 25 тон) та важкі

(більш 25 тон) контейнери. Для цього використовується евристичний алгоритм, що спочатку завантажує найбільш важкі та маленькі контейнери.

Точних перевірок морехідних якостей судна, що було завантажено таким чином, не проводиться.

Надалі у статті [86] «A new three-step heuristic for the Master Bay Plan Problem» пропонується лінійна цілочисельна модель, та трьохетапний підхід для її розв'язання. Вказана модель використовується для мінімізації часу завантаження, вважаючи, що контейнери розподіляються на легкі, середні та важкі, і що важкіші контейнери не можна грузити на більш легкі, та беручи до уваги максимальну вагу одного стеку, неможливість завантаження TEU на FEU, наявність вантажних просторів під і над кришками та максимальний кренуючий момент. Запропоновані етапи розв'язання задачі складаються з наступних:

- Перший – розподіляє вантажний простір судна на конкретні підмножини для кожного порту призначення, розподіляючи навантаження рівномірно та максимізуючи кількість кранів, що можуть працювати паралельно;
- Другий – готує пробний розв'язок для одного порту, зневажаючи деякими вимогами;
- Третій – готує розв'язок для усього переходу, який задовольняє заданим вимогам та використовує табу пошук для мінімізації часу завантаження.

Питання остійності, загальної міцності та розміщення небезпечних вантажів не розглядаються.

Стаття [87] «An experimental comparison of different heuristics for the master bay plan problem» продовжує розглядати лінійну цілочисельну модель, та пропонує два додаткових підходи для її розв'язання. Перший підхід представляє собою просту евристику, що розділяє TEU та FEU, обирає припустимі множини беїв відповідно портів призначення, сортирує контейнери за портами (від останнього до першого), а потім за вагою та поміщає їх у відповідні беї.

Другий підхід використовує модифікований мурашиний алгоритм, що імітує поведінку справжніх мурах для розв'язання комбінаторних проблем. Автори

приводять ефективність мурашиного підходу для великої кількості вантажів, проте обмеження моделі у даній статті адресовані не були.

Стаття [88] «Using a Bin Packing Approach for Stowing Hazardous Containers into Containerships» розглядає судно як 3-D корзину та ділить його на вантажні місця/менші корзини у залежності від розташування бей та кришок. При цьому підході вантажі розподіляються по бейх у залежності від порту призначення, а більш легкі контейнери ставляться на більш важкі за евристичним алгоритмом. Також автори розглядають лише окремі випадки несумісних контейнерів та не розглядають загальну міцність судна.

Стаття [89] «Experimental evaluation of mixed integer programming models for the multi-port master bay plan problem» використовує дві цілочисельні моделі для порівняння ефективності їх розв'язання. При цьому контейнери поділяються на три типи за вагою: легкі, середні та важкі. Інші обмеження співпадають з [84]. Автори пропонують використовувати спеціально розроблені евристичні алгоритми, які використовуються для знаходження близьких до оптимальних результатів за приведеними моделями.

У статті «A Genetic Algorithm with a Compact Solution Encoding for the Container Ship Stowage Problem» [90] була приведена спрощена модель контейнерного судна, яке має один бей з R ярусів та C колонок, що перевозить вантажі у N портів.

Для розв'язання даної задачі використовується генетичний алгоритм з евристично підібраними параметрами, що намагається вирішити задачу, мінімізуючи кількість перестановок контейнерів. При цьому ураховується вплив ваги контейнерів на крен судна та використовується функція штрафу для його мінімізації. Проте перевірок інших обмежень не проводиться.

Автори статті «Container stowage pre-planning: Using search to generate solutions, a case study» [91] поділяють задачу на два етапи: стратегічне та тактичне планування.

На першому етапі використовується об'єктивна функція, яка мінімізує кількість руху кришок трюмів, перестановок кранів та максимізує кількість кранів,

що використовуються у кожному наступному порту вивантаження. Розв'язання цієї задачі виконується за допомогою методу гілок і меж, який повторюється для кожного порту вивантаження. Результат обирається за допомогою функції допасованості, що мінімізує небажані результати.

На другому етапі функція мінімізує кількість перестановок контейнерів та стеків контейнерів, призначених у різні порти та розміщає контейнери по вазі (найбільш важкі – знизу). Розв'язання даної функції виконується за допомогою алгоритму табу-пошуку. При цьому усі контейнери вважаються одного типу, не проводиться перевірок остійності, міцності або ін.

У статті «Automated stowage planning for large containerships with improved safety and stability» [92] пропонується модель судна, яка включає беї та баластні танки. Автори розглядають розміщення множини 20 та 40 футових контейнерів (TEU – Twenty-foot equivalent unit та FEU – Forty-foot equivalent unit) з урахуванням місцевої міцності, диференту, крену, видимості та остійності. Для виконання обмежень використовується евристичний алгоритм, що переміщує контейнери, які не задовольняють заданим умовам, на вільні місця. Для полегшення задачі також використовуються баластні танки, проте автори наповнюють їх попарно задля уникання крену.

Варто зазначити, що перевірки загальної міцності, розташування небезпечних вантажів, або взаємного розташування TEU та FEU не проводиться.

Автори статті [93] «Dangerous goods container allocation in ship stowage planning» пропонують метод розміщення небезпечних вантажів через задавання припустимих слотів для контейнерів, що їх містять, та розміщення такого контейнера випадковим чином у одному з них. При цьому автори не розглядають інші проблеми розміщення контейнерів.

У статті [94] «Integer Linear Programming Models for the Containership Stowage Problem» пропонуються чотири цілочисельні моделі для різних варіантів завантаження судна з урахуванням декількох портів вивантаження. Перша модель розглядає завантаження порожнього судна без кришок лише TEU, друга дає

можливість додати кришки вантажних приміщень, третя дозволяє вантажити частково заповнене судно, а четверта дозволяє використовувати різні розміри та типи контейнерів. Однак при цьому не враховуються місцева міцність або розміщення небезпечних вантажів. Також автори використовують готові рішення для розв'язання зазначених моделей.

Стаття [95] «A Rule-based Greedy Algorithm to Solve Stowage Planning Problem» розглядає судно як єдиний прямокутний бей, що містить N колонок та M рядів. Автори розглядають усі контейнери як одного типу та розміру та пропонують жадібний алгоритм, що, намагаючись скоротити кількість перестановок контейнерів, грузить контейнери за порядком спадання починаючи з терміналу, у який направляється найбільша кількість контейнерів. Ніякі інші фактори завантаження не розглядаються.

Автори статті [96] «Research on Decision-making Optimization of Container MBPP Based on Evolutionary Strategy Algorithm» розглядають завантаження судна з мінімізацією кількості перестановок контейнерів та кренуючого і диферентуючого моментів. Вони спрощують модель, не враховуючи остійності, міцності або характеристики небезпечних вантажів, а також припускаючи, що усі контейнери є TEU, пропорція їх ваг є заданою, перший порт є портом завантаження, а усі послідовні – портами вивантаження.

У статті [97] «Solution strategies for a multiport container ship stowage problem» пропонується лінійна цілочисельна модель, що враховує непродуктивні перестановки контейнерів, проте вона розглядає судно як один бей та не враховує ніякі інші параметри при завантаженні, такі як вага контейнерів, їх тип, характеристики небезпечних вантажів та ін. Хоча запропонований алгоритм вирішив задачу у такій постановці, автори признають його неефективність у разі доповнення моделі для розв'язання реальних задач.

Стаття [98] «Container ship stowage plan using steepest ascent hill climbing, genetic, and simulated annealing algorithms» також розглядає судно як один бей з заданою кількістю рядів та стовпців. При цьому робиться ряд припущень, а саме: судно має мати повну наповненість на протязі усього переходу, розглядаються

тільки FEU контейнери, рефрижераторні контейнери повинні розміщуватися у двох найвищих рядках вищезазначеного бею, остійність та вага контейнерів не враховуються. При розв'язанні задачі автори фокусуються на мінімальній кількості перестановок контейнерів у портах. Для розв'язання моделі використовуються декілька алгоритмів, а саме: генетичний, імітації відпалу та алгоритм сходження на вершину. Незважаючи на незначні відмінності результатів, для розв'язання задачі у поставленому авторами вигляді найбільш ефективним є алгоритм імітації відпалу за рахунок швидкості виконання.

Автори статті [99] «Solving the Container Stowage Problem (CSP) using Particle Swarm Optimization (PSO)» використовують модель, запропоновану Данієлою Амброзіно. Для розв'язання задачі використовується модифікований метод рою часток, який показав ефективність вище заданого евристичного методу.

У статті [100] «Container vessel stowage planning system using genetic algorithm» автори розподіляють задачу на дві фази, а саме розподілення контейнерів по бехах та розташування окремих контейнерів у межах заданого бею. Начальне розташування контейнерів виконується випадковим шляхом, а потім використовується генетичний алгоритм, який базується на штрафах за невиконання обмежень. Автори розглядають обмеження по розташуванню TEU та FEU, обмеження по розташуванню рефрижераторних контейнерів, обмеження по посадці і диференту, вітру (обмеження по кількості контейнерів у одному стеку) та лінії зору. Автори не розглядають обмеження остійності, міцності або небезпечних вантажів.

Стаття [101] «Multi-objective optimization based on improved genetic algorithm for containership stowage on full route» пропонує модель, що враховує кількість перестановок контейнерів, диферент, порядок портів та деякі параметри остійності, а саме метацентричну висоту GM, період хитавиці. Автори оптимізують початковий розв'язок, заснований на мінімізації кількості перестановок, щодо диференту та періоду хитавиці. Для розв'язання поставленої задачі вона поділяється на два етапи. На першому етапі здобувається початковий розв'язок за допомогою евристичного алгоритму. Другий етап використовує модифікований

генетичний алгоритм, що змінює початкове розташування контейнерів поки не буде отриманий результат, що задовольняє параметрам. При цьому не розглядаються вимоги щодо міцності або небезпечних вантажів.

Автори статті [102] «A deep Q-learning network for ship stowage planning problem» використовують алгоритм глибокого навчання з підкріпленням для оптимізації за такими критеріями, як мінімізація перестановок контейнерів, мінімізація руху берегових кранів та порядок розташування контейнерів. При цьому розглядаються вимоги щодо розташування контейнерів (не у повітрі), максимальної ваги, що можна загрузити у кожен контейнерну позицію та розташування більш важких контейнерів під легшими. Варто зазначити, що дані вимоги є дуже спрощеними і не можуть розглядатися для практичного завантаження.

У статті [103] «A GRASP algorithm for the container stowage slot planning problem» автори пропонують цілочисельну математичну модель, що враховує неможливість розташування контейнерів у повітрі, неможливість постановки TEU на FEU, розташування рефрижераторних контейнерів лише у попередньо заданих слотах, обмеження щодо максимальної ваги одного стеку та розміщення небезпечних вантажів. При цьому виконується максимізація кількості завантажених контейнерів, мінімізація перестановок і пустих слотів та уникнення завантаження звичайних контейнерів у рефрижераторні слоти. Запропонований алгоритм завантажує судно по стекам, і завантажує кожний стек повністю перед переходом на інший. Повторні ітерації з випадковим вибором контейнерів і стеків виконуються поки задані критерії не виконуються. При цьому не виконуються перевірки морехідних якостей, таких як параметри остійності або загальної міцності.

Автори статті [104] «Optimizing constraint test ordering for efficient automated stowage planning» пропонують двоетапний евристичний алгоритм, що використовує «розділяй та володарюй» парадигму, що полягає у розбитті головної задачі на підзадачі меншого розміру і комбінуванні їх розв'язків для отримання відповіді. На першому етапі виконується розташування контейнерів з мінімізацією

перестановок у портах. На другому етапі проводиться покращення параметрів безпеки за допомогою перестановки контейнерів та баластування. Варто зазначити, що вищевказана стаття розглядає лише перший етап розв'язання. Вводиться поняття ваги (або вартості) обмежень, які накладаються на розташування контейнерів. Автори пропонують переміщувати обмеження, що було порушено, на половину довжини списку наперед, що дозволяє перевіряти ті обмеження, що зустрічаються найчастіше, першими, і тим самим економити час розв'язання задачі.

У статті [105] «A Constraint Programming model for fast optimal stowage of container vessel bays» пропонуються дві математичні моделі для розв'язання задачі завантаження, а саме цілочисельного програмування та програмування обмеженнями. При цьому автори ураховують обмеження щодо розташування контейнерів у стеках, взаємне розташування TEU і FEU, обмеження за вагою та висотою контейнерів та намагаються мінімізувати кількість перестановок та використаних стеків, запобігти завантаження звичайних контейнерів у рефрижераторні слоти та завантаження контейнерів, направлених у різні порти, у один стек. Перевірок щодо виконання вимог щодо розташування небезпечних вантажів та остійності або загальної міцності не виконується.

Модель, запропонована авторами [106] «Randomized algorithm with tabu search for multi-objective optimization of large containership stowage plans», ставить своєю ціллю мінімізувати кількість перестановок контейнерів, тривалість найдовшого вивантаження одним краном, уникнути стеки, що перевищують припустиму вагу, мінімізувати кількість вільних слотів, різниці горизонтальних (що відповідають за крен) та поздовжніх (що відповідають за диферент) моментів. Перевірок щодо вимог остійності або загальної міцності не виконується. Для розв'язання задачі автори успішно використовують алгоритм табу пошуку.

1.3 Висновки за розділом 1

Після аналізу літературних джерел з питання підвищення безпеки судноплавства було встановлено, що ця проблема має наступні основні напрямки вирішення:

- 1) Підвищення остійності шляхом модифікації методів розрахунку її параметрів та їх нормування.
- 2) Більш детальний аналіз міцності при впливі різних типів навантажень та напруг; модифікація методів її оцінки.
- 3) Аналіз непотоплюваності суден у різних ситуаціях.
- 4) Аналіз хитавиці та вплив зовнішніх факторів на її параметри.
- 5) Зниження пошкоджень суден під час експлуатації.
- 6) Підвищення ефективності кріплення вантажів.
- 7) Формування, обробка та представлення даних для вантажних програм.
- 8) Автоматизація завантаження суден.

Останні пункти представляють особливий інтерес у зв'язку з сучасним розвитком обчислювальної техніки та можливості підвищення ефективності перевезень шляхом оптимізації розміщення вантажів. При розгляді робіт щодо автоматизації завантаження суден було помічено значну кількість спрощень, що не дозволяють застосовувати запропоновані методи у реальних задачах. Зокрема роботи, що розглядають завантаження контейнерних суден, мають суттєві недоліки, усунення яких вимагає більш детального аналізу цієї задачі.

Роботи, що відносяться до розв'язання проблеми завантаження контейнеровозів, або основної проблеми бей-плану, можна поділити на дві категорії: ті, що розглядають проблему ціликом та ті, що підрозділяють її на декілька підпроблем.

У роботах першої категорії використовуються математичні моделі та застосовуються алгоритми з розгалуженням або евристичні.

Роботи другої категорії, як правило, поділяють проблему на дві частини. Спочатку контейнери розподіляються по кластерам судна. Надалі відбувається розподіл контейнерів у межах таких кластерів.

При розгляданні даної задачі з урахуванням декількох портів, проблема розглядається як динамічна для завантаження та вивантаження судна. В останній час, двома основними підходами для її розв'язання є одно та багатофазний.

У однофазному підході задача розглядається цілком, обираються стратегії завантаження, що мінімізують перестановки контейнерів у портах, як правило, враховуючи багато портів призначення та велику кількість контейнерів, але нехтуючи вимогами до морехідних якостей та розглядаючи один вид контейнерів.

При багатофазному підході, задача розбивається на декілька оптимізаційних моделей, кожна з яких вирішується окремо. Такий підхід дозволяє включати більш реалістичні обмеження, такі як параметри морехідних якостей, обмеження берегових кранів, тощо.

РОЗДІЛ 2. МЕТОДОЛОГІЧНЕ ЗАБЕЗПЕЧЕННЯ НАУКОВОГО ДОСЛІДЖЕННЯ

2.1 Вибір теми дослідження

Безпека судноплавства є багатоаспектною проблемою, розв'язання якої потребує врахування великої кількості факторів, чимало з яких залежать від правильного завантаження судна. Аналіз літературних джерел встановив, що однією з найбільш актуальних та важливих проблем при забезпеченні безпеки судноплавства є автоматизація процесу завантаження суден, зокрема контейнерних, при забезпеченні обмежень щодо їх безпеки, що також дозволяє підвищити ефективність вантажних операцій. Тому основним напрямом дисертаційного дослідження є вдосконалення методів складання вантажного плану контейнеровозу з урахуванням як структурних, так і операційних обмежень.

Актуальність проблеми надалі підкреслюється великою кількістю наукових праць за даним напрямком, чим і обумовлюється вибір теми дисертації, яка має наступне формулювання: «Розробка методу розв'язання задачі складання вантажного плану контейнеровозу з урахуванням структурних та операційних обмежень».

Основною науковою гіпотезою дослідження є допущення про можливість вдосконалення підготовки вантажного плану контейнеровозу, використовуючи покращену математичну модель.

Розв'язання задачі потребує розгляду наступних питань:

- 1) Здійснити декомпозицію головної задачі на складові за допомогою методів системного підходу.
- 2) Розглянути обмеження, які необхідно враховувати при складанні вантажного плану.
- 3) Розробити математичну модель судна з урахуванням обраних обмежень.
- 4) Перевірити розроблену модель за допомогою точного методу.

- 5) Здійснити підготовку початкових даних для розв'язання отриманої задачі виконання обмежень.
- 6) Розробити метод розв'язання задачі у заданому вигляді.
- 7) Провести комп'ютерне моделювання завантаження за допомогою розробленого методу та проаналізувати отримані результати.

Актуальність тематики дисертаційного дослідження визначається необхідністю підвищення ефективності складання вантажного плану при забезпеченні безпеки судноплавства.

Розробка гнучкої математичної моделі, яка може бути використана для розв'язання задачі складання вантажного плану контейнерних суден і методів розподілу та оптимізації розміщення вантажів за заданими критеріями, можуть скласти наукову новизну дисертаційного дослідження.

Можливість реалізації запропонованого наукового дослідження забезпечуються розробкою математичної моделі і методів розподілу вантажів та перевіркою теоретичної частини за допомогою імітаційного моделювання.

Об'єктом дослідження роботи є процеси забезпечення морехідної безпеки контейнерних суден.

Предметом дослідження є математичні моделі та методи завантаження суден.

2.2 Методи дисертаційного дослідження

Технологічна карта методологічного забезпечення дисертаційного дослідження обґрунтовує послідовність розв'язання головної задачі дисертації методами системного підходу (Рис. 2.1).

В даний час проблеми морехідної безпеки суден, розробки сучасних методів завантаження суден та оптимізації вантажного плану представляються одними з найважливіших запитів практики.

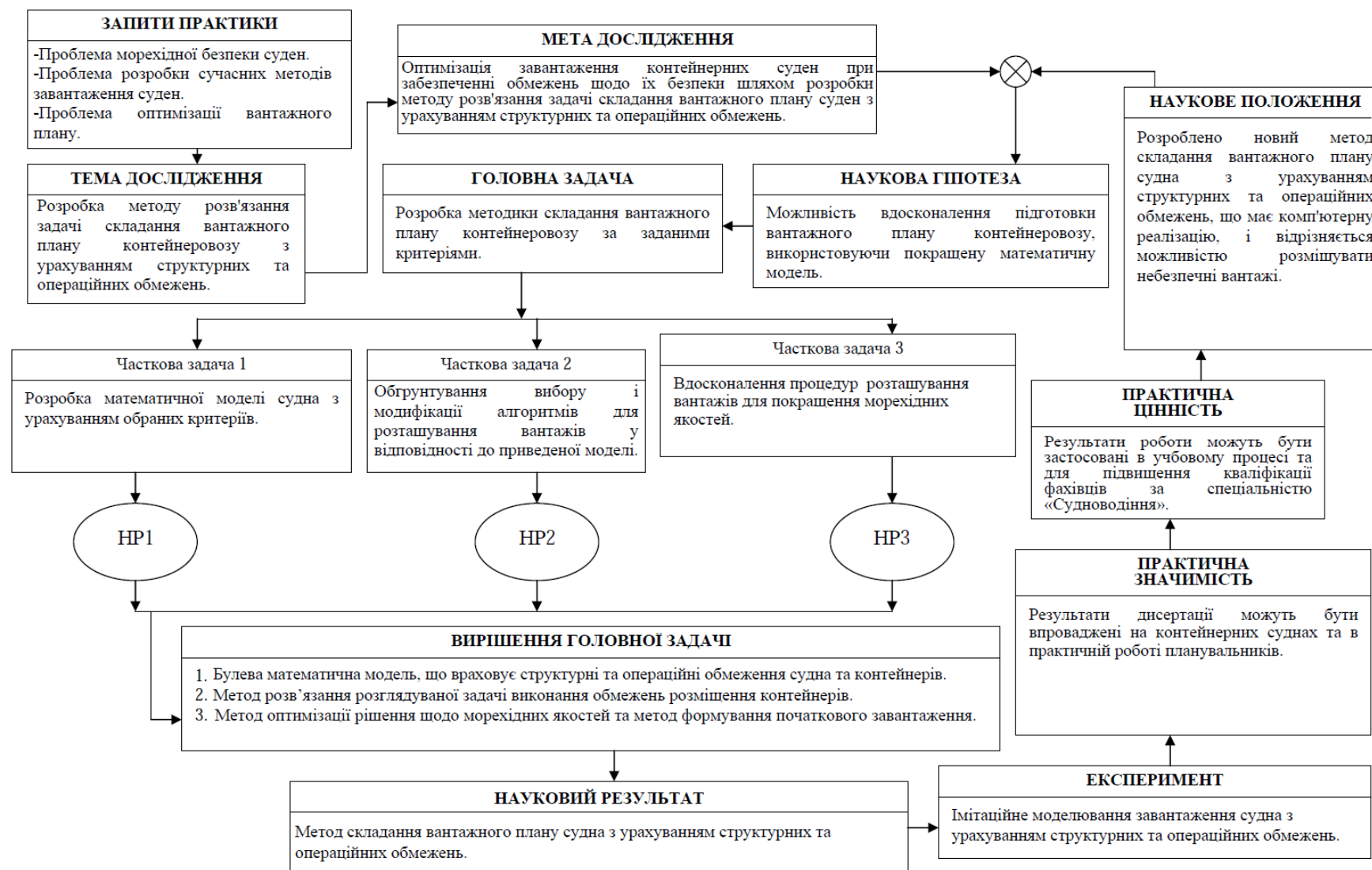


Рис. 2.1 Технологічна карта дисертації

Метою дисертаційного дослідження являється оптимізація завантаження контейнерних суден при забезпеченні обмежень щодо їх безпеки шляхом розробки методу розв'язання задачі складання вантажного плану контейнерних суден з урахуванням структурних та операційних обмежень.

Наукова гіпотеза дисертаційного дослідження полягає у допущенні про можливість вдосконалення підготовки вантажного плану контейнеровозу, використовуючи покращену математичну модель.

Друга складова передбачає два етапи. На першому етапі виконується аналіз та вибір методу розв'язання моделі. На другому етапі виконується модифікація обраного методу для конкретної моделі.

Третя складова полягає у оптимізації розв'язку, отриманого після застосування результуючого методу, таким чином, щоб отриманий розв'язок був припустимим щодо вимог до морехідних якостей судна. Окрім цього виконується розробка методу початкового розташування вантажів на судні.

У результаті розв'язання складових задач були отримані нові наукові результати, які на технологічній карті методологічного забезпечення дисертаційного дослідження позначені відповідно НР1, НР2 і НР3:

- булева математична модель, що враховує структурні та операційні обмеження судна та контейнерів, у загальному та чисельному виглядах являється науковим результатом НР1 першої складової задачі;
- метод для розв'язання розглядуваної задачі виконання обмежень щодо розміщення контейнерів на борту судна є науковим результатом НР2 другої складової задачі;
- метод оптимізації розміщення контейнерів щодо параметрів мореплавства, який використовує вагу, порт призначення, тип і клас небезпечного вантажу кожного окремого контейнера та відрізняється можливістю використовувати гнучку цільову функцію, яка може

модифікуватися щодо окремих параметрів мореплавства у залежності від особливостей конкретного рейсу, а також метод формування початкового завантаження судна контейнерами є науковим результатом НРЗ третьої складової задачі.

Проведено імітаційне моделювання перевірки результатів дисертаційного дослідження щодо коректності розробленого методу підготовки вантажного плану судна за заданими критеріями.

Наукова гіпотеза відносно можливості вдосконалення підготовки вантажного плану контейнеровозу, використовуючи покращену математичну модель, була підтверджена одержаними в дисертаційній роботі теоретичними результатами та імітаційним моделюванням.

Даному дисертаційному дослідженню притаманна практична значимість, яка полягає в збільшенні автоматичної складової в процесі складання вантажного плану. Результати дисертаційного дослідження можна упроваджувати у практичній роботі судноводіїв та суднохідних планувальників.

Практична цінність дослідження полягає у потенційному використанні результатів для складання вантажного плану контейнерних суден, розробленні програмного забезпечення для завантаження контейнерних суден, у навчальному процесі та для підвищення кваліфікації штурманів.

Наукові результати, одержані в дисертаційному дослідженні, і проведене імітаційне моделювання визначають його наукове положення, яке може бути сформульовано таким чином:

розроблено новий метод складання вантажного плану судна з урахуванням структурних та операційних обмежень, що має комп'ютерну реалізацію, і відрізняється можливістю розміщувати небезпечні вантажі.

2.3 Методика проведення дисертаційного дослідження

Стислий виклад методики, що використовувалась для проведення дисертаційного дослідження, приведений у цьому підрозділі.

По-перше, необхідно методами дедукції провести огляд поточних розробок в області забезпечення безпеки судноводіння, що зумовлює вибір теми дисертації.

Наступний етап виконання роботи – декомпозиція головної задачі дисертаційного дослідження на складові та забезпечення методологічного обґрунтування дисертаційного дослідження.

У зв'язку з наступним етапом роботи виконується розробка булевої математичної моделі контейнеровозу з урахуванням структурних та операційних обмежень. Надалі слід модифікувати точний класичний метод розв'язання отриманої задачі виконання обмежень для перевірки адекватності математичної моделі. Для перевірки необхідно провести чисельний експеримент з такими обмеженнями, які дозволяють сильно обмежити множину припустимих позицій для розміщення контейнерів.

Проведення чисельного експерименту потребує приведення математичної моделі до числового вигляду. Окремим завданням є модифікація алгоритму сортувальної станції Едсгера Дейкстри, заснованого на стеку, для цілей даної роботи.

Наступний етап роботи полягає у розробці методу для задачі розподілу вантажів на контейнеровозі.

В подальшому слід провести оптимізацію розв'язку щодо вимог до морехідних якостей та розробити метод початкового розміщення вантажів на судні.

Для аналізу результатів необхідно провести імітаційне моделювання, що дозволить перевірити коректність запропонованого методу складання вантажного плану судна за заданими критеріями.

2.4 Висновки за розділом 2

Розділ 2 дисертаційної роботи присвячений вибору теми дисертаційного дослідження та основним напрямкам цієї кваліфікаційної роботи. В даному розділі представлена технологічна карта дослідження, яка відображає методологічну структуру дисертаційної роботи, та приведено методологічне забезпечення дисертаційного дослідження.

В технологічній карті приведена мета дисертаційної роботи та її головна задача, яка представлена трьома складовими. Сформульована робоча гіпотеза наукового дисертаційного дослідження. Також продемонстровано, що в дисертаційній роботі одержані результати, яким властива наукова новизна.

У підрозділі 2 також вказано практичну цінність і значимість проведеного дисертаційного дослідження, що підтверджено можливістю застосування практичних результатів, одержаних в роботі. Також в цьому розділі наведено формулювання основного наукового положення дисертаційної роботи.

В підрозділі «Методика проведення дисертаційного дослідження» представлено стислу методику розв'язання складових задач дисертаційного дослідження, отриманих в дисертаційній роботі. В цьому підрозділі наведені послідовні етапи наукового дослідження згідно теми дисертації.

Методика вказує на застосування сучасних методів теоретичного аналізу, методів конструювання компіляторів, математичних, кількісних методів теорії дослідження операцій для обґрунтування отриманих рішень. Також запропонована нова булева математична модель контейнеровозу з урахуванням структурних та операційних обмежень та метод розв'язання задачі розподілу вантажів на контейнеровозі.

РОЗДІЛ 3. РОЗРОБКА МАТЕМАТИЧНОЇ МОДЕЛІ ОСНОВНОЇ ПРОБЛЕМИ БЕЙ-ПЛАНУ

3.1 Вимоги щодо розміщення контейнерів з небезпечними вантажами

При перевезенні небезпечних вантажів морем мають виконуватись вимоги національного і міжнародного законодавств щодо пакування, маркування, розміщення і оформлення вантажів.

На території України діє Закон України «Про перевезення небезпечних вантажів» від 06.04.2000 року {із змінами, внесеними згідно з законами N 586-VI (586-17) від 24.09.2008, ВВР, 2009, N 10-11, ст.137; N 2189-VI (2189-17) від 13.05.2010, ВВР, 2010, N 30, ст.394; N 5459-VI (5459-17) від 16.10.2012, ВВР, 2013, N 48, ст.682; N 5502-VI (5502-17) від 20.11.2012, ВВР, 2014, N 8, ст.88 }. Згідно з ним державне управління у сфері перевезення небезпечних вантажів здійснюють Кабінет Міністрів України, Рада міністрів Автономної Республіки Крим, центральний орган виконавчої влади, що забезпечує формування державної політики у сфері транспорту, місцеві державні адміністрації та органи місцевого самоврядування, а також інші спеціально уповноважені державні органи відповідно до їх компетенції. Вищевказаний Закон охоплює вимоги до відправника, перевізника і одержувача небезпечних вантажів. Щодо умов перевезення у Законі сказано: «Умови перевезення небезпечних вантажів визначаються нормативно-правовими актами, що регулюють діяльність транспорту. У разі відсутності таких актів небезпечний вантаж допускається до перевезення на умовах, встановлених за рішенням органів, зазначених у статтях 13, 15 та 16 цього Закону» [107].

Окрім вищевказаного закону в Україні прийняті інші нормативні документи, в котрих висуваються додаткові вимоги до транспорту і персоналу, а саме: Постанова Кабінету Міністрів України «Про затвердження Порядку і правил проведення

обов'язкового страхування відповідальності суб'єктів перевезення небезпечних вантажів на випадок настання негативних наслідків під час перевезення небезпечних вантажів» станом на 22-03-2017, Постанова Кабінету Міністрів України «Про затвердження Порядку проведення спеціального навчання працівників суб'єктів перевезення небезпечних вантажів» станом на 23-04-2015, Наказ Міністерства інфраструктури України 05.11.2014 № 565 «ЗМІНИ до Правил перевезення небезпечних вантажів» та ін.

Що стосується міжнародного законодавства, головним юридичним документом, який регламентує перевезення небезпечних вантажів морем, є IMDG Code (International maritime dangerous goods code, Міжнародний морський кодекс з перевезень небезпечних вантажів морем), який складається на основі офіційних документів IMO (International maritime organization, Міжнародна морська організація). Поправки до конвенції SOLAS-74 (Safety of life at sea, Міжнародна конвенція з охорони людського життя на морі), а саме глави VII (Перевезення небезпечних вантажів), які були прийняті у травні 2002 року, зробили кодекс обов'язковим з 1 січня 2004 року.

Згідно до кодексу небезпечні вантажі підрозділяються на 9 класів. Проте розділення лише на класи не є достатнім для забезпечення безпечного судноплавства, тому, відповідно до кодексу, деякі класи небезпечних вантажів додатково підрозділяються на підкласи що дозволяє точніше визначити специфіку таких вантажів.

Окрім розділення за підкласами, небезпечні речовини мають свій UNN (United Nations number, номер ООН) згідно з рамкою міжнародного транспорту. Цей номер міститься у томі 2 кодексу. Згідно цього номеру визначається приналежність вантажу до групи сумісності і категорії розташування. Проте UNN може позначати і декілька речовин відразу.

Існують наступні класи та підкласи небезпечних вантажів:

1. Вибухові матеріали
 - 1.1. Вибухові матеріали з небезпекою вибуху масою
 - 1.2. Вибухові матеріали, що не вибухають масою
 - 1.3. Вибухові матеріали, що є пожежонебезпечними, але не вибухають масою
 - 1.4. Субстанції і предмети, що не представляють значної небезпеки
 - 1.5. Дуже нечутливі матеріали з небезпекою вибуху масою
 - 1.6. Вироби з надзвичайно низькою чутливістю
2. Гази
 - 2.1. Легкозаймисті гази
 - 2.2. Незаймисті неотруйні гази
 - 2.3. Токсичні гази
3. Легкозаймисті рідини
4. Легкозаймисті тверді речовини; речовини, що можуть спонтанно запалитися, речовини, що виділяють легкозаймисті гази при контакті з водою
 - 4.1. Легкозаймисті тверді речовини
 - 4.2. Самозаймисті речовини
 - 4.3. Речовини, що виділяють займисті гази при взаємодії з водою
5. Окислюючі речовини та органічні перекиси
 - 5.1. Окислюючі речовини
 - 5.2. Органічні перекиси
6. Токсичні та інфекційні речовини
 - 6.1. Токсичні речовини
 - 6.2. Інфекційні речовини
7. Радіоактивні матеріали
8. Корозійні субстанції
9. Різні небезпечні субстанції та предмети.

Положення щодо розміщення і сегрегації небезпечних вантажів розміщені у розділі 7 Міжнародного кодексу з морського перевезення небезпечних вантажів, що має назву «Положення, що стосуються транспортних операцій».

Сегрегація – це процес розділення двох або більше речовин або предметів, що приймаються взаємно несумісними коли їх сумісна упаковка або розміщення може викликати надмірні ризики у випадку витоку, втрати або будь-якої іншої аварії [108].

Основні вимоги до сегрегації між різними класами небезпечних товарів вказані у кодексі (див. Таблиця 3.1), проте можуть висуватись додаткові вимоги у залежності від конкретних речовин [4].

Проте у будь-якому випадку правила сегрегації треба уточнювати у Списку небезпечних товарів за їх UNN.

Цифри і символи, що зазначені у таблиці, мають наступні значення:

1 – «на відстані від»

2 – «відділено від»

3 – «відділено цілим відсіком або трюмом від»

4 – «відділено поздовжньо проміжним повним відсіком або трюмом від»

X – необхідно звіритися зі Списком небезпечних товарів

* – необхідно звіритися з параграфом 7.2.7.1.

Контейнерне місце, зазначене у приведених таблицях, позначає дистанцію не менш 6 м поздовж судна та не менше 2.4 м поперек.

Окрім загальних вимог, до контейнерних суден висуваються додаткові вимоги при перевезеннях небезпечних вантажів. Ці вимоги вказані у параграфі 7.4 кодексу.

Згідно з параграфом 7.4.2.1 кодексу, небезпечні товари можуть перевозитися усередині або вертикально над трюмами контейнерних суден, що не мають трюмових кришок, якщо перевезення цих небезпечних вантажів дозволено під палубою згідно зі Списком небезпечних товарів, та якщо трюм такого судна повністю відповідає вимогам конвенції SOLAS-74.

Таблиця 3.1

Таблиця сегрегації небезпечних вантажів

КЛАС	1.1 1.2 1.5	1.3 1.6	1.4	2.1	2.2	2.3	3	4.1	4.2	4.3	5.1	5.2	6.1	6.2	7	8	9
1.1,1.2, 1.5	*	*	*	4	2	2	4	4	4	4	4	4	2	4	2	4	X
1.3, 1.6	*	*	*	4	2	2	4	3	3	4	4	4	2	4	2	2	X
1.4	*	*	*	2	1	1	2	2	2	2	2	2	X	4	2	2	X
2.1	4	4	2	X	X	X	2	1	2	X	2	2	X	4	2	1	X
2.2	2	2	1	X	X	X	1	X	1	X	X	1	X	2	1	X	X
2.3	2	2	1	X	X	X	2	X	2	X	X	2	X	2	1	X	X
3	4	4	2	2	1	2	X	X	2	1	2	2	X	3	2	X	X
4.1	4	3	2	1	X	X	X	X	1	X	1	2	X	3	2	1	X
4.2	4	3	2	2	1	2	2	1	X	1	2	2	1	3	2	1	X
4.3	4	4	2	X	X	X	1	X	1	X	2	2	X	2	2	1	X
5.1	4	4	2	2	X	X	2	1	2	2	X	2	1	3	1	2	X
5.2	4	4	2	2	1	2	2	2	2	2	2	X	1	3	2	2	X
6.1	2	2	X	X	X	X	X	X	1	X	1	1	X	1	X	X	X
6.2	4	4	4	4	2	2	3	3	3	2	3	3	1	X	3	3	X
7	2	2	2	2	1	1	2	2	2	2	1	2	X	3	X	2	X
8	4	2	2	1	X	X	X	1	1	1	2	2	X	3	2	X	X
9	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Вимоги до сегрегації на контейнерних суднах представлені у кодексі у вигляді таблиць сегрегації. Для цілей даної роботи розглядається сегрегація вантажів на суднах з відкритими трюмами (див. Таблиця 3.2 та Таблиця 3.3).

Таблиця 3.2

Таблиця сегрегації контейнерів на суднах з відкритими трюмами – част. 1

Вимоги до сегрегації	Вертикально			Горизонтально			
	Закритий з закритим	Закритий з відкритим	Відкритий з відкритим		Закритий з закритим		
					На палубі	Під палубою	
«На відстані від»	Дозволено один на одного	Дозволено відкритий на закритий. Інакше як для "відкритий на відкритий"	Не в одній вертикальній лінії	Вперед і назад	Без обмежень	Без обмежень	
				Поперек судна	Без обмежень	Без обмежень	
«Відділено від»	Не в одній вертикальній лінії	Так само, як для "відкритого з відкритим"		Вперед і назад	Одне контейнерне місце	Одне контейнерне місце або одна переборка	
				Поперек судна	Одне контейнерне місце	Одне контейнерне місце	
«Відділено цілим відсіком або трюмом від»				Вперед і назад	Одне контейнерне місце і не в або над одним трюмом	Одна переборка	
				Поперек судна	Два контейнерних місця і не в або над одним трюмом	Одна переборка	
«Відділено поздовжньо проміжним повним відсіком або трюмом від»*	Заборонено			Вперед і назад	Мінімальна горизонтальна відстань 24м і не в або над одним трюмом	Одна переборка та мінімальна горизонтальна відстань 24м	
				Поперек судна	Заборонено	Заборонено	

Таблиця 3.3

Таблиця сегрегації контейнерів на судах з відкритими трюмами – част. 2

Вимоги до сегрегації	Горизонтально				
		Закритий з відкритим		Відкритий з відкритим	
		На палубі	Під палубою	На палубі	Під палубою
«На відстані від»	Вперед і назад	Без обмежень	Без обмежень	Одне контейнерне місце	Одне контейнерне місце або одна переборка
	Поперек судна	Без обмежень	Без обмежень	Одне контейнерне місце	Одне контейнерне місце
«Відділено від»	Вперед і назад	Одне контейнерне місце	Одне контейнерне місце або одна переборка	Одне контейнерне місце і не в або над одним трюмом	Одна переборка
	Поперек судна	Два контейнерних місця	Два контейнерних місця	Два контейнерних місця і не в або над одним трюмом	Одна переборка
«Відділено цілим відсіком або трюмом від»	Вперед і назад	Одне контейнерне місце і не в або над одним трюмом	Одна переборка	Два контейнерних місця і не в або над одним трюмом	Дві переборки
	Поперек судна	Два контейнерних місця і не в або над одним трюмом	Одна переборка	Три контейнерних місця і не в або над одним трюмом	Дві переборки
«Відділено поздовжньо проміжним повним відсіком або трюмом від»	Вперед і назад	Мінімальна горизонтальна відстань 24м і не в або над одним трюмом	Дві переборки	Мінімальна горизонтальна відстань 24м і не в або над одним трюмом	Дві переборки
	Поперек судна	Заборонено	Заборонено	Заборонено	Заборонено

3.2 Математична модель

Зазвичай вантажний план контейнерного судна складається по баях, у яких можуть міститись TEU або FEU. При складанні математичної моделі усі беї судна розглядаються одного розміру, такого, що має ширину та висоту найбільшого бея судна. При цьому ті беї, що є меншими, матимуть відповідні контейнерні місця зайнятими.

Контейнер розміру t ($0 = \text{TEU}$, $1 = \text{FEU}$), класу IMDG c ($c = 0$ позначає безпечний вантаж), зберігається в позиції (i, j, k) , якщо $x_{tcijk} = 1$ [3].

Координата i відповідає за порядковий номер бею, починаючи з носу; координати j та k – за розташування контейнеру у відповідному беї (Рис. 3.1, Рис. 3.2).

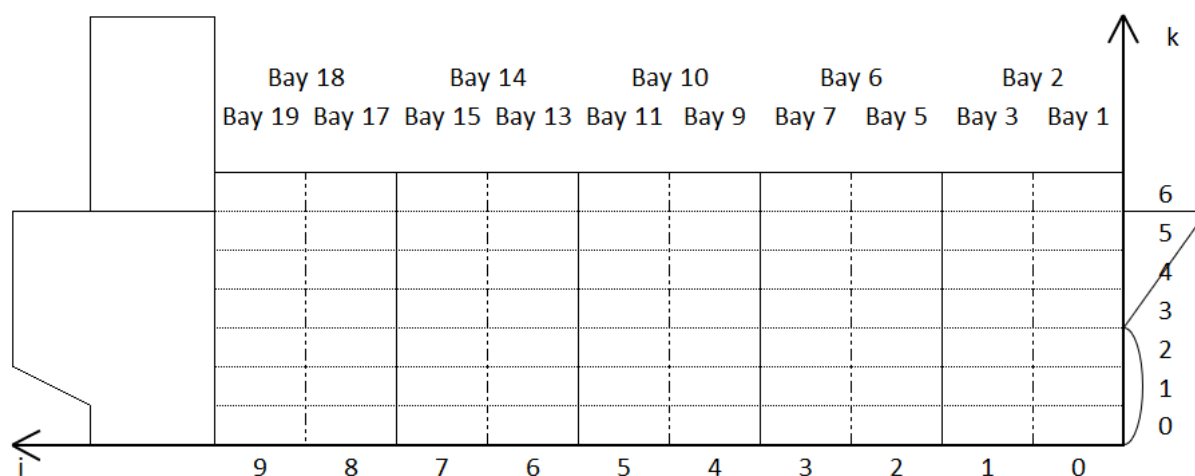


Рис. 3.1 Координати i, k

Змінна t відповідає за розмір контейнеру. Якщо $t = 0$, то контейнер є TEU, якщо $t = 1$ – FEU.

Змінна c відповідає за клас небезпечного вантажу, що знаходиться у контейнері. Вона може приймати 18 значень. Якщо $c = 0$, то контейнер не містить небезпечних

вантажів, якщо $c = 1$, то контейнер містить вантажі класу 1.1, 1.2 або 1.5; якщо $c = 2$, то контейнер містить вантажі класу 1.3 або 1.6 і до $c = 17$ (див. Таблиця 3.1).

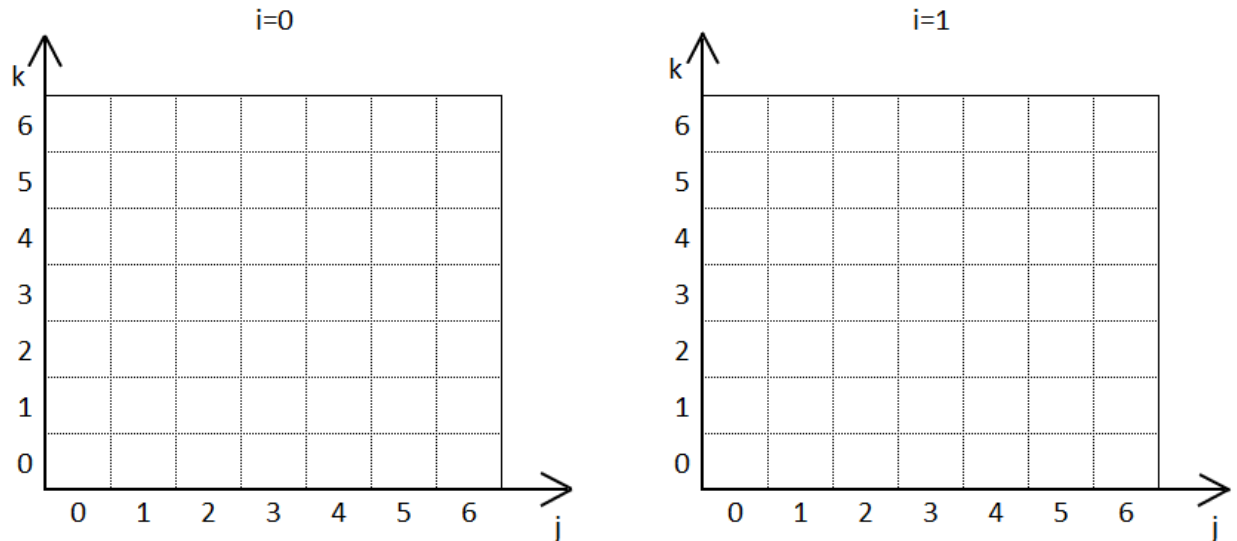


Рис. 3.2 Координати i, j

Перше з розглянутих обмежень – на кількість контейнерів. Для TEU

$\sum_i \sum_j \sum_k x_{0cijk} = n_0^c$ для кожного класу небезпечних вантажів c . Аналогічно для FEU $\sum_i \sum_j \sum_k x_{1cijk} = 2n_1^c$. Після приведення до стандартної форми задачі оптимізації, ці рівняння можна об'єднати:

$$-\left(\sum_i \sum_j \sum_k x_{tcijk}\right) \leq (-1 - t)n_t^c, \quad \forall t, c, i, j, k; \quad (1)$$

$i \in [0..i_{max}]$, де $i_{max} + 1$ – загальна кількість беїв у TEU;

$j \in [0..j_{max}]$, де $j_{max} + 1$ – загальна кількість беїв у TEU;

$k \in [0..k_{max}]$, де $k_{max} + 1$ – загальна кількість беїв у TEU;

FEU займають 2 TEU місця (по i). Таким чином, якщо $x_{1ci'jk} = 1$, то $x_{1c(i'+1)jk} = 1$, і навпаки, де $i' \in i$ – номери бейв під FEU. Стандартний вигляд цього обмеження для задачі оптимізації виглядає наступним чином:

$$|x_{1ci'jk} - x_{1c(i'+1)jk}| \leq 0, \quad \forall c, i', j, k; \quad (2)$$

У одній позиції трюму може знаходитись TEU, FEU, або вона може бути пустою. Тобто якщо $x_{0cijk} = 1$, то $x_{1cijk} = 0$, і навпаки. Після приведення до нерівності:

$$\sum_t \sum_c (x_{tcijk}) \leq 1, \quad \forall i, j, k; \quad (3)$$

Контейнери можуть знаходитися на палубі, кришці, або один поверх іншого, вони не можуть парити у повітрі. Це обмеження виглядає наступним чином:

$$k * \sum_t \sum_c (x_{tcijk}) - \sum_{k^*=0}^{k-1} \sum_t \sum_c (x_{tcijk^*}) \leq 0 \quad \forall i, j, k > 0; \quad (4)$$

Конструкція контейнерів дозволяє ставити FEU поверх TEU, але не дозволяє ставити TEU поверх FEU. Таким чином, якщо $x_{1cijk} = 1$, то $\sum_c \sum_{k^*>k} (x_{0cijk^*}) = 0$. Звідки:

$$2M \sum_c x_{1cijk} + \sum_c \sum_{k^*>k} (x_{0cijk^*}) \leq 2M, \quad \forall i, j, k; \quad (5)$$

У наведеній нерівності M – найбільша кількість контейнерів, що може знаходитися у одному стеку.

Також не можна грузити FEU на стеки TEU різної висоти.

$$\left\{ \left| \left[\sum_c \sum_{k^*=0}^k x_{0ci'jk^*} \right]_s - \left[\sum_c \sum_{k^*=0}^k x_{0c(i'+1)jk^*} \right]_s \right| \leq M(1 - w_s), \quad \forall i', j, k; \right. \\ \left. \left[\sum_c x_{1ci'j(k+1)} + \sum_c x_{1c(i'+1)j(k+1)} \right]_s \leq Mw_s, \quad \forall i', j, k; \right. \quad (6)$$

Для цієї нерівності $w_s \in \{0,1\}$ – додаткова змінна, яка вводиться для кожного обмеження: $s = i * (j_{max} + 1) * (k_{max} + 1) + j * (k_{max} + 1) + k$.

Вимоги до сегрегації небезпечних вантажів можна представити наступним чином:

$$\sum_{i^*=i''-l_{c1c2}}^{i''+l_{c1c2}} \sum_{j=j''-w_{c1c2}}^{j''+w_{c1c2}} \sum_{k^*=k''-h_{c1c2}}^{k''+h_{c1c2}} (x_{t,c1,i^*j^*k^*}) \leq R(1 - x_{tc2ijk}), \quad (7)$$

$\forall t, c1 \neq c2, i'', j'', k''$.

У наведеній нерівності $i'' \in [l_{c1c2} \cdot i_{max} - l_{c1c2}]$, $j'' \in [w_{c1c2} \cdot j_{max} - w_{c1c2}]$, $k'' \in [h_{c1c2} \cdot k_{max} - h_{c1c2}]$, де l_{c1c2} – вимога щодо поздовжнього інтервалу у TEU, h_{c1c2} – вимога щодо вертикального інтервалу, а w_{c1c2} – вимога щодо поперечного інтервалу поміж двома контейнерами IMDG класів $c1$ та $c2$; R – максимальна кількість контейнерів у зоні обмежень.

Отримана таким чином задача (1) – (7) є задачею виконання обмежень (constraint satisfaction problem).

На даному етапі не виконується перевірок щодо остійності або міцності судна.

3.3 Формування обмежень

Для розв'язання приведеної математичної моделі її спочатку необхідно привести у числовий вигляд. У залежності від розміру судна, до якого вона застосовується, кожна з нерівностей може розкриватися до сотень окремих випадків [109].

Для цілей даної роботи вказані нерівності було вирішено привести до вигляду двох матриць: коефіцієнтів при змінних та вільних членів. Тоді отримані дані можна опрацьовувати належним чином.

Для формування матриці коефіцієнтів використовується модифікований алгоритм сортувальної станції Едсгера Дейкстри, заснований на стеку. Модифікації

цього алгоритму широко використовуються у сучасній літературі [110]–[113], проте автору не вдалось знайти приклади його застосування для розв’язання подібних задач.

Алгоритм сортувальної станції використовується для перетворення математичних виразів, записаних у інфіксійній нотації (звичайна форма запису математичних виразів, наприклад, «3+4»), до постфіксної (або зворотньої польської, у такій формі запису «3+4» перетворюється на «3 4 +»). Основною перевагою такого запису є можливість аналізу виразу у процесі одноразового просмотру зліва направо. Для опису цього алгоритму необхідно ввести поняття токена. Токен – це об’єкт, що складається з лексеми у процесі лексичного аналізу; токен складається з імені та опціонального значення. Прикладами імен токенів є ідентифікатор, сепаратор, оператор, ключове слово та інші.

Для застосування алгоритму сортувальної станції математичний вираз розбивається на токени, після чого виконуються наступні дії [114]:

- Доки не всі токени оброблені:
 - Прочитати токен.
 - Якщо токен – число, тоді додати його до черги виводу.
 - Якщо токен – функція, додати його до стеку операторів.
 - Якщо токен – оператор $op1$, тоді:
 - поки існує оператор $op2$ на вершині стеку операторів, який має більший пріоритет, а при рівності пріоритетів $op1$ є лівоасоціативним:
 - перекласти $op2$ до черги виводу
 - покласти $op1$ до стеку.
 - Якщо токен – відкривальна скобка, покласти його до стеку.
 - Якщо токен – закривальна скобка:
 - Поки токен на вершині стеку не є відкривальною скобкою, переміщувати оператори зі стеку до черги виводу.

- Прибрати відкривальну скобку зі стеку, але не додавати до черги виводу.
- Якщо токен на вершині стеку є функцією, перемістити її до черги виводу.
- Якщо токенів не залишилось:
 - Перекласти оператор зі стеку до черги виводу

Для формування матриці коефіцієнтів нерівностей (див Таблиця 3.4), їх спочатку треба привести до наступної форми:

$$\begin{cases} a_{00} * x_{00000} + a_{01} * x_{00001} + \dots + a_{0n} * x_{TCIJK} \leq N_0, \\ \dots \\ a_{m0} * x_{00000} + a_{m1} * x_{00001} + \dots + a_{m,n} * x_{TCIJK} \leq N_m, \end{cases} \quad (8)$$

де $T=t_{\max}$, $C=c_{\max}$, $I=i_{\max}$, $J=j_{\max}$ та $K=k_{\max}$.

Таблиця 3.4

Матриця коефіцієнтів змінних

	$X_{0,0,0,0,0}$	$X_{0,0,0,0,1}$	$X_{0,0,0,0,2}$	$X_{0,0,0,0,3}$	$X_{0,0,0,0,4}$	$X_{0,0,0,0,5}$	...	$X_{T,C,I,J,K}$
$0,0,0,0,0$	$a_{0,0}$	$a_{0,1}$	$a_{0,2}$	$a_{0,3}$	$a_{0,4}$	$a_{0,5}$	...	$a_{0,n}$
$0,0,0,0,1$	$a_{1,0}$	$a_{1,1}$	$a_{1,2}$	$a_{1,3}$	$a_{1,4}$	$a_{1,5}$	...	$a_{1,n}$
$0,0,0,0,2$	$a_{2,0}$	$a_{2,1}$	$a_{2,2}$	$a_{2,3}$	$a_{2,4}$	$a_{2,5}$	...	$a_{2,n}$
...	...	...	...	...	...	...	...	...
T,C,I,J,K	$a_{m,0}$	$a_{m,1}$	$a_{m,2}$	$a_{m,3}$	$a_{m,4}$	$a_{m,5}$	...	$a_{m,n}$

Для приведення нерівностей (1) – (7) до форми (8), їх ліві частини переводяться до зворотнього польського запису, розкладаючи алегбраїчні суми на складові, відкриваючи скобки та задаючи значення відповідних коефіцієнтів t , c , i , j та k .

Розкриття скобок відбувається через подвійне перетворення: приведення нерівностей до постфіксної форми, а потім знову до інфіксної. При цьому невідомі залишаються незмінними, а коефіцієнти при них приводяться. Для застосування алгоритму використовуються наступні пріоритети операцій (див. Таблиця 3.5).

Таблиця 3.5

Пріоритети операцій

Операція	Пріоритет
+	1
—	1
*	2
/	2
^	3
# (унарний мінус)	4
Σ	5

Розкриття сум вимагає додаткових доробок. Для вирішення цієї проблеми було введено знак Σ як окремий символ операції з найвищим пріоритетом. При перетворенні Σ до постфіксної нотації результат записується наступним чином:

- параметр, за яким відбувається сумування;
- вираз;
- Σ .

Наприклад, для виразу $\sum_k x_{tcijk}$ перетворення відбувається наступним чином:

- k
- x_{tcijk}
- Σ

Для виразу $\sum_{k^*=k}^{k_{\max}} (x_{0cijk^*})$:

- $k^*=k..k_{\max}, 1$
- x_{0cijk^*}
- Σ

У виразі « $k^*=k..k_{\max}, 1$ » 1 – це крок змінної k^* .

Як видно з прикладів сумування може відбуватися двома шляхами. Перший – за усіма індексами, як у $\sum_k x_{tcijk}$, та другий – що залежить від конкретних поточних значень змінних, як у $\sum_{k^*=k}^{k_{\max}} (x_{0cijk^*})$.

$\sum_k x_{tcijk}$ розкривається як $x_{tcij0} + x_{tcij1} + x_{tcij2} + \dots + x_{tcijk_{\max}}$, підставляючи кожне можливе значення k при перетворенні з постфіксної форми запису назад до інфіксної. Після чого відбувається циклічна підстанова кожного значення кожного параметру та запис відповідного коефіцієнту до матриці.

$\sum_{k^*=k}^{k_{\max}} (x_{0cijk^*})$ розкривається для кожного окремого k .

При поєднанні двох випадків одразу, роботу алгоритму можна представити у наступній формі (Рис. 3.3).

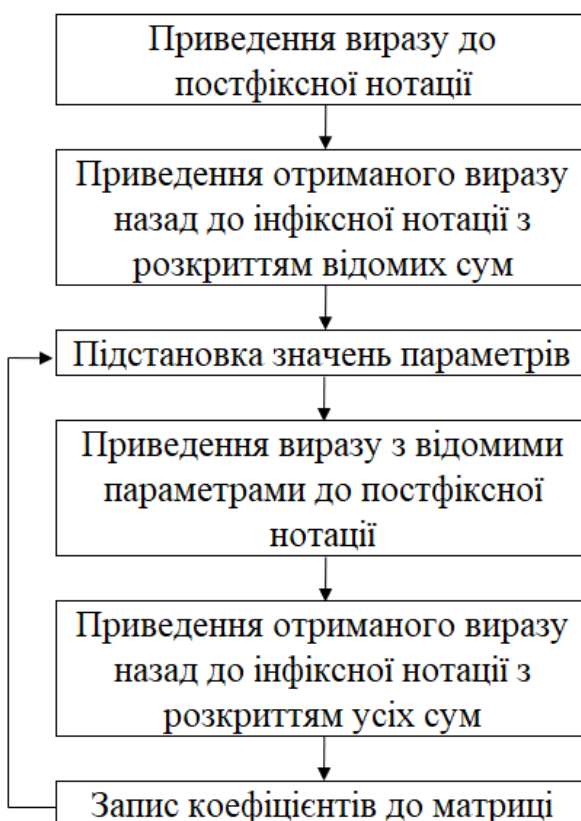


Рис. 3.3 Модифікований алгоритм сортувальної станції

Спочатку відбувається розкриття незалежних від поточних параметрів сум, після чого виконується підстановка параметрів та повторне розкриття сум для поточних значень параметрів. По завершенні усі коефіцієнти записуються до матриці.

Щодо матриці вільних членів, вона формується паралельно з матрицею коефіцієнтів для кожної розкритої нерівності.

3.4 Перевірка адекватності моделі

Для перевірки адекватності математичної моделі (1) – (7) було проведено чисельний експеримент з «жорсткими» обмеженнями, коли конструктивні обмеження контейнерів і судна, правила розміщення небезпечних вантажів дозволяють різні види контейнерів розташувати у вкрай обмеженій кількості позицій. Для перевірки адекватності математичної моделі в якості алгоритму для розв'язання задачі був обраний метод гілок і меж, який є точним (не наближеним) методом розв'язання.

Дана задача виконання обмежень відноситься до булевого програмування; що враховується при модифікації обраного алгоритму. У розглянутому алгоритмі обчислювальні операції обмежуються тільки складанням і відніманням. Тому алгоритм розв'язання задач з булевими змінними іноді називають адитивним методом. Цей алгоритм спирається на ідеї загального методу гілок і меж [115].

Для реалізації адитивного алгоритму задача повинна мати такий вигляд [5].

Мінімізувати $z = \sum_{j=1}^n c_j x_j$, де $c_j \geq 0$, при обмеженнях
 $\sum_{j=1}^n a_{ij} x_j \leq b_i$, де $x_j = 0$ або 1 , $i = 1, 2, \dots, m$; $j = 1, 2, \dots, n$.

Симплекс-таблиця даної задачі визначає припустимий розв'язок, коли всі $c_j \geq 0$.

Для розв'язання задачі адитивним алгоритмом обмеження записуються у вигляді:

$$\sum_{j=1}^n a_{ij}x_j + S_i = b_i, \quad (9)$$

де $S_i \geq 0$ – додаткова змінна, яка відповідає i -му обмеженню, $i = 1, 2, \dots, m$.

Початкова задача розміщення контейнерів була сформульовано таким чином: мінімізувати функцію

$$z = \sum_{t \ c \ i \ j \ k} x_{tcijk} \quad (10)$$

при зазначених обмеженнях (1) – (7), які записуються у вигляді (9).

Основна ідея адитивного алгоритму полягає в переборі 2^n можливих розв'язків початкової задачі. Процедура перебору здійснюється спеціальним чином, що дозволяє відкидати деякі розв'язки без безпосередньої перевірки. В кінцевому підсумку, реалізація адитивного алгоритму вимагає безпосереднього розгляду тільки частини з 2^n можливих розв'язків.

На першому кроці всі початкові змінні приймаються рівними 0. Це виконується через те, що у цільовій функції усі $c_j \geq 0$. Отриманий розв'язок (жодного контейнера не завантажено) є неприпустимим, тому що деякі $S_i < 0$. Тому деяким вхідним змінним необхідно присвоїти значення 1. Метою цієї процедури є виконання умови $S_i \geq 0 \ \forall i = 1, 2, \dots, m$, т. є. забезпечення припустимості розв'язку. На кожному кроці алгоритму визначаються змінні, яким необхідно присвоїти значення 1 і 0. Зазначений вибір здійснюється за допомогою чотирьох класичних тестів [115]. При реалізації алгоритму багатовимірний масив x_{tcijk} і одновимірний масив w_s було перетворено у одновимірний масив.

У зв'язку з тим, що математична модель для задачі завантаження контейнерного судна має певний вигляд, класичний адитивний алгоритм був доповнений ще 4-ма додатковими тестами. Ці тести дозволяють присвоювати нульові значення невідомим змінним без додаткових перевірок, що дозволяє зменшити кількість переборів можливих розв'язків початкової задачі.

Перший з додаткових тестів перевіряє, що на FEU контейнер неможливо поставити TEU контейнер, тому, якщо в процесі розв'язання було отримано значення змінної $x(1, c_0, i_0, j_0, k_0) = 1$, то $x(0, c, i_0, j_0, k) = 0$ для $\forall k > k_0$ і $\forall c \in \{0, 1, \dots, 17\}$.

Другий тест забезпечує те, що у одну позицію може бути завантажено лише один контейнер, або вона може бути порожньою, тобто Якщо в процесі розв'язання було отримано значення змінної $x(1, c_0, i_0, j_0, k_0) = 1$, то $x(0, c, i_0, j_0, k_0) = 0$ для $\forall c \in \{0, 1, \dots, 17\}$ і $x(1, c, i_0, j_0, k_0) = 0$ для $\forall c \in \{0, 1, \dots, 17\}$, $c \neq c_0$. Аналогічно, якщо $x(0, c_0, i_0, j_0, k_0) = 1$, то $x(0, c, i_0, j_0, k_0) = 0$ для $\forall c \in \{0, 1, \dots, 17\}$, $c \neq c_0$ і $x(1, c, i_0, j_0, k_0) = 0$ для $\forall c \in \{0, 1, \dots, 17\}$.

Третій додатковий тест виключає змінні, що не задовольняють вимогам щодо небезпечних вантажів, які представлені у вигляді (7). Якщо в деякому обмеженні присутні тільки невід'ємні коефіцієнти, і права частина цього обмеження дорівнює 0, то відповідному коефіцієнту змінної (елемент масиву) присвоюється 0:

$$\sum_{j=1}^n a_{i_0 j} = \sum_{j=1}^n |a_{i_0 j}| > 0 \quad \text{і} \quad b_{i_0} = 0,$$

Четвертий тест дозволяє виключити такі змінні, які викликають значення цільової функції більше, ніж $\sum_{c=0}^{17} (n_{TEU}^c + n_{FEU}^c)$, при присвоєнні їм значення 1. Тобто такі змінні повинні залишатися нулями.

Варто зазначити, що практичне використання програм для ЕОМ (електронно обчислювальних машин), призначених для розв'язання цілочислових задач, зазвичай передбачає активне втручання оператора в процес обчислень [115].

Як приклад для підтвердження адекватності математичної моделі розглядалася спрощена ситуація, де 40 контейнерів (5 FEU і 35 TEU), серед яких 3 FEU і 1 TEU з небезпечним вантажем класу 4.1, розташовані у одному прямокутному трюмі (Рис. 3.4). Потрібно додатково розмістити 6 контейнерів (2 FEU і 4 TEU), серед яких 1 FEU і 1 TEU з небезпечним вантажем класу 5.2.

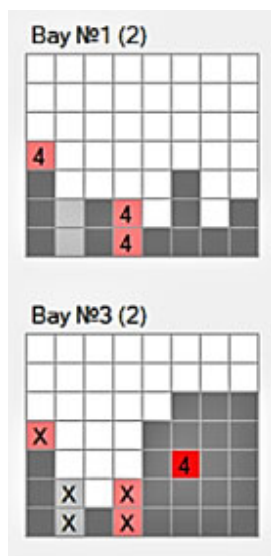


Рис. 3.4 Початкове розміщення контейнерів у трюмі

На рисунку ■ – *TEU* контейнер з вантажем класу 0; ■ – *FEU* контейнер з вантажем класу 0; 4 – *FEU* контейнер з вантажем класу 4.1; 4 – *TEU* контейнер з вантажем класу 4.1

Як впливає з обмеження (7), контейнери з небезпечними вантажами класу 5.2 можна встановлювати тільки в позиціях (0, 7, 2), (0, 7, 3), (0, 7, 4), (0, 7, 5), (0, 7, 6), (1, 7, 5), (1, 7, 6), тобто тільки в останньому стовпці бейв №1 і №3. Що стосується *FEU* контейнерів, то *FEU* контейнер з вантажем класу 5.2 можна встановити тільки в 2-х місцях: (0, 7, 5), (1, 7, 5) і (0, 7, 6), (1, 7, 6) і тільки за умови, що під ними не буде вільних місць. Такі «жорсткі» умови обрані для того, щоб перевірити і продемонструвати адекватність моделі і працездатність методу. Зазвичай, на практиці, для розміщення контейнерів з небезпечними вантажами прийнятними є значно більша кількість позицій.

Нижче (Рис. 3.5) наведено один з розв'язків, отриманих за допомогою модифікованого адитивного алгоритму.

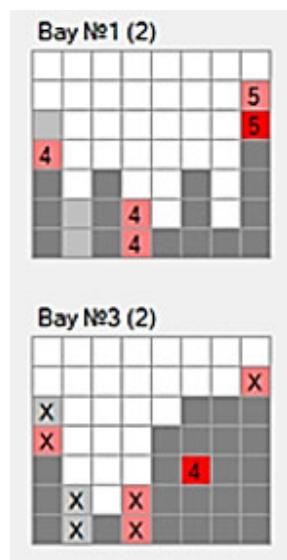


Рис. 3.5 Отримане розміщення контейнерів у трюмі

На рисунку **5** – FEU контейнер з вантажем класу 5.2; **5** – TEU контейнер з вантажем класу 5.2

Таким чином, отримано припустиме розміщення контейнерів, а саме:

- 1) Розміщені всі початкові контейнери відповідно до умови (1);
- 2) FEU контейнери займають по 2 місця, і не розміщені на стеках TEU контейнерів різної висоти відповідно до умов (2) і (6);
- 3) Позиції, зайняті TEU і FEU, не перетинаються відповідно до умови (3);
- 4) TEU контейнери не розміщені на FEU контейнерах відповідно до умови (5);
- 5) Сегрегація небезпечних вантажів дотримана відповідно до умови (7).

В результаті проведеного чисельного експерименту були отримані припустимі розташування контейнерів, одне з яких приведено. У якості вхідних даних, як вказувалося, навмисно були обрані такі, які дозволяють різні види контейнерів розташувати у вкрай обмеженій кількості позицій. Це може свідчити про адекватність запропонованої математичної моделі для задачі бей-плану, так як результат розрахунку з використанням даної моделі за допомогою точного методу відповідає припустимому розташуванню контейнерів у трюмі.

3.5 Висновки за розділом 3

Розташування контейнерів у вантажному просторі контейнеровозу вимагає врахування ряду обмежень, невиконання яких робить вантажний план неприпустимим. До таких обмежень відносяться структурні обмеження самих контейнерів (наприклад, TEU контейнери не можна ставити поверх FEU), обмеження щодо сегрегації небезпечних вантажів, тощо.

Запропоновано математичну модель, яка дозволяє забезпечувати виконання таких обмежень для виконання перевірки та виконання обмежень завантаження.

Приведено метод перетворення обмежень математичної моделі до двовимірної матриці за допомогою алгоритму сортувальної станції, що дозволяє облегшити опрацювання окремих випадків.

Приведений метод перетворення було застосовано для перевірки адекватності моделі. Був проведений чисельний експеримент з «жорсткими» обмеженнями, коли конструктивні обмеження контейнерів і судна та правила розміщення небезпечних вантажів дозволяють розташувати різні види контейнерів у вкрай обмеженій кількості позицій. Результат, отриманий внаслідок проведення вищезазначеного експерименту, відповідав припустимому розташуванню контейнерів у трюмі.

РОЗДІЛ 4. РОЗРОБКА МЕТОДУ РОЗВ'ЯЗАННЯ ЗАДАЧІ ВИКОНАННЯ ОБМЕЖЕНЬ

4.1 Вибір методу

Задача складання вантажного плану контейнерного судна є NP повною [116], тобто вона може бути розв'язана шляхом повного перебору усіх можливих розв'язків.

Зазвичай, повний перебір не є практичним для знаходження необхідного розв'язку через надзвичайну велику множину можливих розв'язків. Наприклад, для контейнерного судна місткістю C TEU існує P_r можливих розташувань r контейнерів.

$$P_r = \frac{C!}{(C - r)!} \quad (11)$$

Навіть для невеликого контейнеровозу місткістю 2000 TEU ($C = 2000$) повне завантаження має $2000!$ варіантів. Очевидно, що повний перебір усіх можливих розв'язків не є практичним, тому для розв'язання подібних NP проблем використовуються інші підходи, а саме:

- апроксимація;
- рандомізація;
- обмеження;
- параметризація;
- евристичний.

Для випадків, коли оптимальний розв'язок невідомий, простір розв'язків є надзвичайно великим, складання евристичного алгоритму не є можливим, проте оцінка окремого розв'язку може бути проведена, найчастіше використовуються методи метаевристики. Поставлена задача підходить під даний опис, так як визначити оптимальний розв'язок не є можливим (у залежності від особливостей рейсу, оптимальні розв'язки можуть спиратися на різні параметри, такі як: кількість завантажених контейнерів, максимальна посадка, максимальна кількість контейнерів деякого типу, тощо), простір усіх можливих розв'язків охопити неможливо, як

показано вище, та складання евристичного алгоритму послідовного завантаження вимагає урахування дуже великої кількості факторів, які також будуть змінюватись у залежності від конкретного рейсу. Проте оцінити припустимість окремого розв'язку можливо за деякими параметрами (кількість невірно розташованих контейнерів, параметри остійності, міцності і т.п.).

До найбільш розповсюджених метаевристичних алгоритмів оптимізації, відносяться наступні: генетичні (Genetic Algorithms / GA), імітації відпалу (Simulated Annealing / SA), табу пошуку (Tabu Search / TS), рою часток (Particle Swarm Optimization / PSO), мурашиний (Ant Colony Optimization / ACO), тощо.

При виборі алгоритму головним чином розглядались ті, які застосовувались для MBPP або задачі пакування рюкзака, яка є більш загальним випадком розглядуваної задачі.

Основним призначенням алгоритму імітації відпалу (SA) є пошук глобального мінімуму або максимуму функції, яка має декілька локальних мінімумів або максимумів [117]. SA алгоритми також відрізняються можливістю застосування для нелінійних моделей та шумних даних [118], проте при їх застосуванні виникає багато небажаних та можлива втрата оптимальних розв'язків [119].

Табу пошук (TS) базується на жадібному алгоритмі локального пошуку, що перевіряє простір розв'язків поза локальним оптимумом та використовує «пам'ять», що дозволяє використовувати попередні результати для покращення наступних [120]. Головним недоліком TS алгоритмів є сильна залежність від якості початкового рішення [121].

Алгоритми рою часток (PSO) оснований на поведінці зграй птах. Кожна частка має позицію та швидкість у просторі рішень; зберігає якість розв'язку, який вона представляє, найкращий розв'язок, що вона мала у минулому та найкращий розв'язок, що мали сусідні частки [122]. Основною областю застосування таких алгоритмів є пошук у безперервному просторі, хоча вони також застосовуються для пошуку

дискретних розв'язків. При застосуванні PSO алгоритмів конвергенція значно сповільнюється при збільшенні розміру задачі [123].

Мурашиний алгоритм (ACO) заснований на спілкуванні штучних мурах за допомогою феромонів, які працюють як ймовірнісні функції для пошуку або модифікації розв'язків [124]. Хоча модифікації ACO алгоритму мають багато переваг, вони обмежені повільною конвергенцією, яка виникає через те, що феромони, які використовуються для керування процесом пошуку, задаються випадковим чином на початку роботи алгоритму [125].

У поставленій задачі припустимість розміщення залежить від конкретних положень окремих контейнерів, але зміна їх положень для виконання вимог за одним критерієм може викликати невиконання інших, і обрати з таких розв'язків один для подальшої модифікації не є тривіальною задачею. Тому у межах даної роботи було вирішено використовувати такий метод, що зберігає декілька розв'язків, і виконує модифікацію такої множини. Такі методи називаються популяційними, і двома найбільш розповсюдженими з них є генетичний алгоритм, та еволюційна стратегія.

Для розв'язання математичної моделі, представленої у попередньому розділі було обрано генетичний алгоритм (GA). Вибір зумовлено тим, що такі алгоритми мають високі показники успіху для задач, що мають велику область розв'язків та при відсутності необхідності знаходження глобального оптимуму [126] та є дуже гнучкими. Окрім цього GA були успішно застосовані багатьма авторами для розв'язання подібних задач [90], [98], [101].

4.2 Модифікація методу

Генетичний алгоритм відноситься до еволюційних алгоритмів та оснований на процесі природного добору. Для описання генетичних алгоритмів використовуються наступні поняття:

- Хромосома – один з можливих розв’язків проблеми, закодований деяким чином.
- Ген – одинарний біт або короткий блок хромосоми.
- Допасованість – якість розв’язку. Функція допасованості оцінює придатність розв’язку щодо поставлених умов.

Такі алгоритми зазвичай використовуються для розв’язання оптимізаційних та пошукових проблем, використовуючі спеціальні операції мутації, схрещення та селекції.

За допомогою селекції з популяції обираються хромосоми для схрещення. Чим вища допасованість окремої хромосоми, тим вищий в неї шанс породити нового нащадка.

Схрещення – операція, що змішує елементи батьківських хромосом для отримання нащадків.

Мутація випадковим чином змінює елементи батьківських хромосом.

Стандартний генетичний алгоритм може бути описаний наступним чином:

- 1) Створити випадкову початкову популяцію з n хромосом.
- 2) Обчислити допасованість кожної хромосоми у популяції.
- 3) Повторювати наступні кроки, поки не створено n нащадків:
 - a. Обрати пару батьківських хромосом з наявної популяції, ймовірність вибору батьків має бути пропорційною до їх допасованості.
 - b. З деякою ймовірністю p_c схрестити елементи батьків.
 - c. Мутувати дві хромосоми-нащадка з деякою ймовірністю p_m , та розмістити отримані хромосоми у новій популяції.
- 4) Замінити початкову популяцію новою.
- 5) Повернутися до кроку 2.

Кожна ітерація цього процесу називається поколінням.

4.2.1 Селекція

На відміну від інших еволюційних алгоритмів, які обирають декілька найбільш допасованих особин, генетичний алгоритм є ітераційним, що дозволяє декілька варіантів відбору.

Оригінальною технікою селекції для генетичного алгоритму був пропорційний відбір. У цьому алгоритмі якщо допасованість хромосоми вища, вона обирається більш часто. Для цього особини «масштабуються» пропорційно їх допасованості (Рис. 4.1).

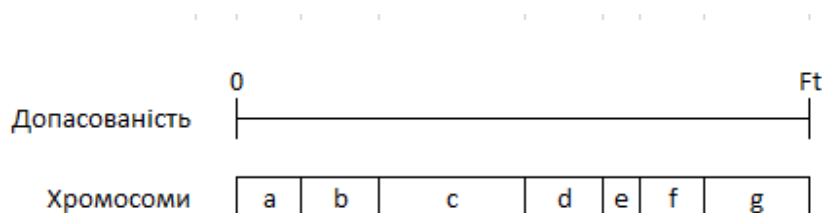


Рис. 4.1 Масив хромосом при пропорційному відборі

Якщо F_t – сума допасованостей усіх елементів масиву, то випадкове число, що знаходиться між 0 та F_t знаходиться у межах одного з елементів, який обирається. Головним недоліком такої селекції являється те, що через випадкову природу відбору хромосома з найвищою допасованістю може бути не обрана.

Цей недолік усувається іншим варіантом селекції, що має назву стохастичного універсального відбору. Вона виконується через випадкове перемішування особин у векторі, після чого відбувається відбір n особин, через рівні проміжки F_t/n (Рис. 4.2).

Як видно з рисунку, при такій селекції особини, що мають допасованість більше деякого значення, завжди обираються для схрещення, а при достатньо великій допасованості, можуть навіть обиратися декілька разів.

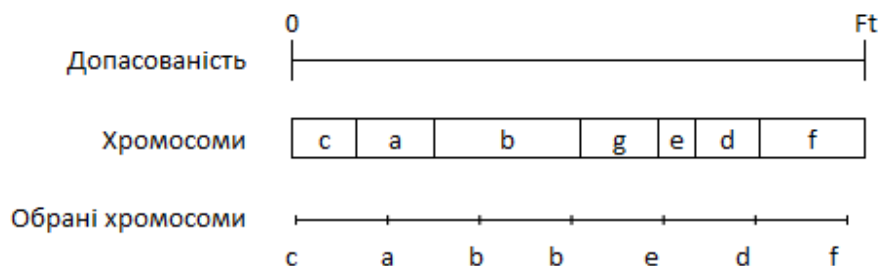


Рис. 4.2 Массив хромосом при стохастичному універсальному відборі

Проте обидва способи, вказані вище, мають один суттєвий недолік – вони не гарантують вибір найбільш допасованої хромосоми для схрещення. При близьких значеннях допасованості, наприклад 101.1, 101.2 та 101.3 ймовірність вибору будь якої особини приблизно однакова. Таким чином алгоритм може видавати майже повністю випадкові значення.

Для усунення цього недоліку найчастіше використовується турнірний відбір. У цьому алгоритмі з початкової популяції випадково обирається t хромосом, допасованість яких потім порівнюється для визначення найкращої особини. Цей алгоритм можна представити наступним чином:

- Для популяції P
- Для розміру турніру $t, t \geq 1$
- V – найкраща особина, обрана випадково
- Повторити t разів:
 - N – інша особина, обрана випадково.
 - Якщо допасованість N більше допасованості V , тоді
 - $V = N$

Цей алгоритм має ряд переваг. По перше, він не залежить від особливостей функції допасованості. По друге, він простий для розуміння та впровадження. По третє, він дає можливість налаштування: змінюючи розмір турніру t , змінюється його

селективність. При $t=0$ алгоритм видає випадкове значення, при t набагато більшому за розмір популяції, ймовірність вибору найкращої особини наближається до 100%. Через це даний алгоритм був обраний для розв'язання поставленої задачі.

4.2.2 Кодування хромосом та функція допасованості

Для отримання припустимого варіанту завантаження судна за допомогою генетичного алгоритму кодування хромосом є одним з найважливіших факторів, від яких залежить успішність результату. Існує декілька розповсюджених варіантів кодування хромосом [126]:

- Бінарний – найбільш розповсюджений варіант, хромосома представлена як вектор фіксованої довжини, що містить булеві значення.
- Багатознаковий та дійсний – застосовуються декілька знаків алфавіту або дійсні числа.
- За допомогою дерев – застосовуються дерева (як структури даних), що дозволяє не обмежувати область пошуку.

Застосування булевої моделі для розв'язання поставленої задачі робить використання одновимірної булевої матриці фіксованої довжини найбільш доцільним для її перевірки.

У розробленій моделі контейнер розміру t , класу IMDG c , зберігається в позиції (i, j, k) , якщо $x_{tcijk} = 1$, інакше $x_{tcijk} = 0$. Також використовується додаткова змінна $w_s \in \{0,1\}$, яка вводиться для кожного обмеження: $s = i * (j_{max} + 1) * (k_{max} + 1) + j * (k_{max} + 1) + k$.

Таким чином матриця розв'язку, або особина, має довжину lv ,
 $lv = ((t_{max} + 1) * (c_{max} + 1) + 1) * (i_{max} + 1) * (j_{max} + 1) * (k_{max} + 1)$.

Нехай Mi – матриця коефіцієнтів змінних нерівностей (див. Таблиця 3.4), Mc – матриця вільних членів, а Mx – матриця розв'язку.

$$Mx = (x_{00000}, x_{00001}, \dots, x_{t_{max}c_{max}i_{max}j_{max}k_{max}}, \dots, w_{000}, w_{001}, \dots, w_{i_{max}j_{max}k_{max}})$$

Коли на судно не завантажено жодного контейнеру, усі елементи матриці Mx є нулями. Після завантаження кожного контейнеру, відповідний елемент Mx замінюється на одиницю. Як тільки всі контейнери завантажено, можна перевірити його правильність згідно математичної моделі наступним чином:

- 1) Обчислити одновимірну матрицю $Mt = Mi * Mx$.
- 2) Порівняти отриману матрицю Mt з Mc . Так як кожна нерівність моделі має вигляд $\sum_{n'=1}^n x_{m',n'} \leq b_{m'}$, то завантаження є вірним при $Mt \leq Mc$, тобто $Mt_{m'} \leq Mc_{m'} \forall ni$.

При невиконанні цієї нерівності розв'язок не є припустимим та має бути виправленим.

Опираючись на вищесказане, можна отримати непридатність розв'язку ug (функція, зворотна допасованості) як:

$$ug = \sum_{m'=0}^m ui_{m'}, \text{ де}$$

$$ui_{m'} = \begin{cases} 0, & \text{якщо } Mt_{m'} \leq Mc_{m'}; \\ 1, & \text{якщо } Mt_{m'} > Mc_{m'}. \end{cases}$$

Звідси функція допасованості $fg = \frac{1}{1+ug}$. Така функція рахує кількість нерівностей, що не було задоволено, та видає результат на проміжку $(0,1]$, що визначає допасованість розв'язку. Проте такий підхід має суттєвий недолік, а саме: кожна невиконана нерівність оцінюється однаково. Таким чином може виникнути ситуація, де один розв'язок містить один невірно розміщений контейнер, а інший – п'ять, і при цьому їх допасованості будуть рівними.

Щоб уникнути такої ситуації, функцію допасованості було модифіковано наступним чином:

$$fg = \frac{1}{1 + ug}$$

$$ug = \sum_{m'=0}^m ui_{m'}, \text{ де}$$

$$ui_{m'} = \begin{cases} 0, \text{ якщо } Mt_{m'} \leq Mc_{m'}; \\ \sum_{n'=1}^n Mi_{m'n'} \wedge Mx_{n'}, \text{ якщо } Mt_{m'} > Mc_{m'}, \end{cases}$$

де операція $\wedge$ позначає кон'юнкцію, або перетин множин.

Таким чином функція допасованості враховує не лише кількість нерівностей, які не задовольняють умовам, а також і кількість невірно розташованих контейнерів для кожної такої нерівності.

4.2.3 Схрещення та мутація

Схрещення є основною особливістю генетичних алгоритмів. Воно включає відбір і змішування окремих частин батьківських особин для отримання нащадків. Як правило, у залежності від задачі, обирається один з наступних видів схрещення:

- одноточкове,
- двоточкове,
- однорідне.

Одноточкове схрещення векторів замінює усі елементи одного батька на елементи іншого, починаючи з деякої позиції. Наприклад, маючи булеві вектори $\bar{a}$ і $\bar{b}$ довжиною lv і позицію lc , при їх схрещенні елементи отриманих векторів $\bar{c1}$ та $\bar{c2}$, будуть складатися з перших lc елементів вектору $\bar{a}$, та останніх, починаючи з lc , елементів вектору $\bar{b}$ та навпаки (Рис. 4.3).

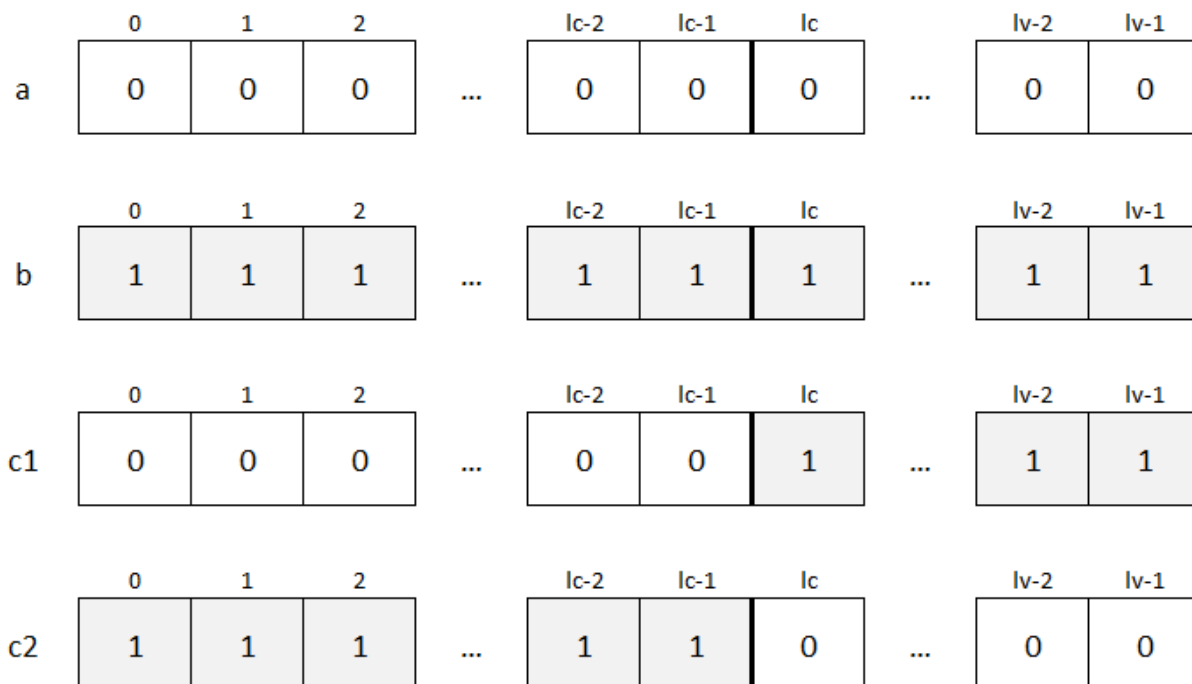


Рис. 4.3 Одноточкове схрещення

Якщо $lc = 0$, схрещення не відбувається, вектори $\bar{c}1$ та $\bar{c}2$ будуть рівними векторам $\bar{a}$ і $\bar{b}$. Недоліком такого схрещення є можливе зчеплення (або епістаз) між елементами векторів. Він полягає у тому, що шанс розриву будь яких двох послідовних елементів векторів є дуже малим, та зростає з віддаленістю елементів. Наприклад, якщо для підвищення допасованості тільки один з елементів $\bar{a}_{lc-2}$ та $\bar{a}_{lc-1}$ має бути у результуючих векторах, то ймовірність такого результату є досить малою. І навпаки, якщо $\bar{a}_0$ та $\bar{a}_{lv-1}$ покращують допасованість разом, то існує велика ймовірність погіршення результату при такому схрещенні, так як воне викличе їх розрив.

Для боротьби з даним недоліком часто використовується двоточкове схрещення. Воно полягає у тому, що обираються дві позиції, між якими відбувається заміна елементів одного вектору на елементи іншого. Наприклад, маючи булеві вектори $\bar{a}$ і $\bar{b}$ довжиною lv і позиції $lc1$ та $lc2$, при їх схрещенні елементи отриманих

векторів $\overline{c1}$ та $\overline{c2}$, будуть складатися з елементів вектору $\overline{a}$, окрім позицій між $lc1$ та $lc2$, і навпаки (Рис. 4.4).

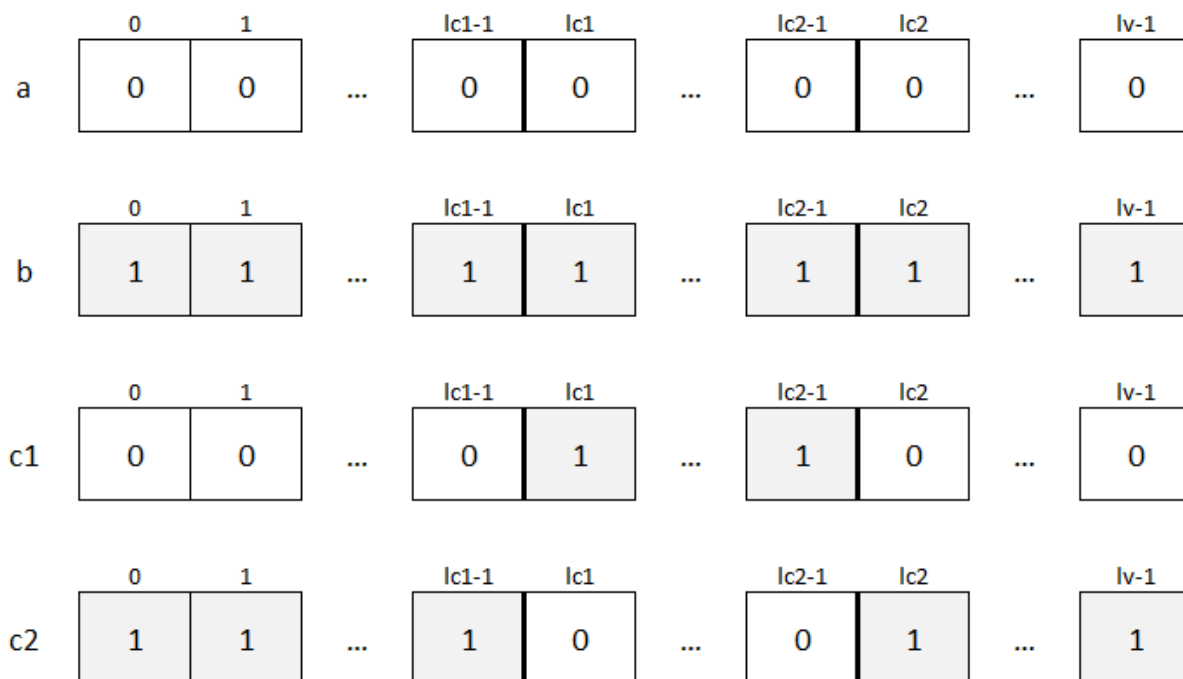


Рис. 4.4 Двоточкове схрещення

Проте при такому схрещенні $\overline{a}(0)$ та $\overline{a}(lv/2)$ і подібні елементи мають більший шанс не бути включеними разом до результуючого вектору, що знов таки може погіршити результат при їх залежності.

Для того, щоб усунути проблему зчеплення, використовується однорідне схрещення. Воно полягає у тому, що кожен елемент вектору обирається з однаковою ймовірністю (Рис. 4.5).

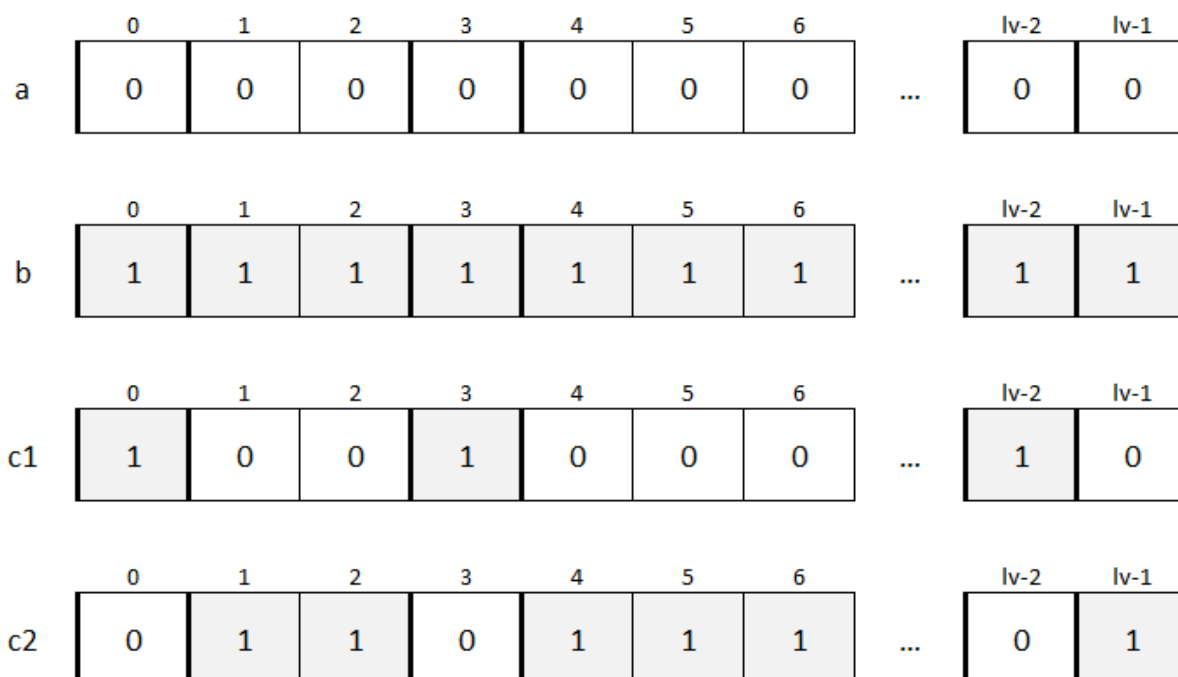


Рис. 4.5 Однорідне схрещення

Як правило, завантаження судна відбувається заздалегідь визначеною множиною контейнерів, що мають бути розміщені у певних позиціях на судні. Тому при кожному варіанті розміщення для кожного x параметри t та c визначені наперед.

Схрещувати вектори у вигляді, який використовується для оцінки їх допасованості, викликатиме постійну зміну кількості контейнерів (так як будь яка зміна 1 на 0 вбирає один контейнер, і навпаки, зміна 0 на 1 добавляє новий). Для усунення цієї проблеми було вирішено працювати з контейнерами у вигляді двовимірної матриці їх координат на судні (i, j та k). Через те, що допасованість розв'язку (або особини) визначається розташуванням окремих контейнерів, то при схрещенні також має сенс обмінюватись повними наборами координат контейнерів, тобто рядками таких матриць/параметрами i, j та k одночасно.

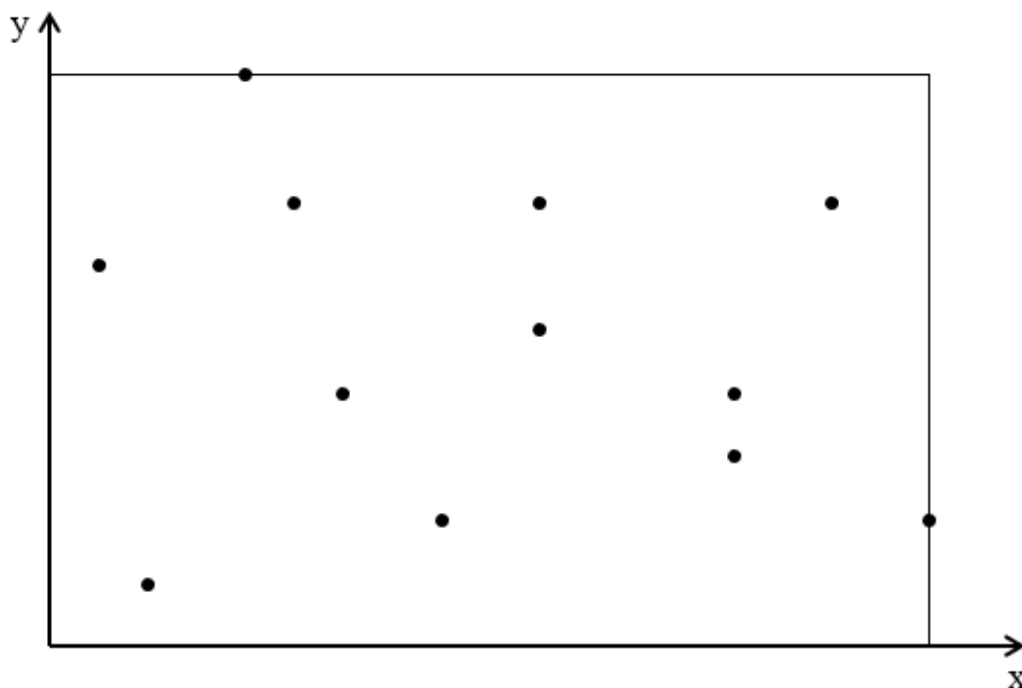


Рис. 4.7 Межі можливих нащадків

Для булевих векторів зазвичай використовується побітова мутація, тобто кожний з елементів вектору може бути змінений на протилежний (0 на 1 і навпаки) з деякою ймовірністю. Для розв'язання поставленої задачі виконувати стандартну мутацію на векторі, що використовується для оцінки допасованості, непродуктивно, тому що внаслідок такої мутації розглядується область параметрів, які не відносяться до конкретної задачі. Наприклад, якщо у списку контейнерів для завантаження немає жодного з небезпечними вантажами, то потреби розглядати область розв'язків, де $t \neq 0$, немає. Тому для даної задачі мутація виконується наступним чином:

- Розглядається матриця координат існуючих контейнерів, що вказана вище;
- Вводяться максимальні значення параметрів i_{max} , j_{max} та k_{max} ;

- При переборі елементів матриці координат деякі координати змінюються на випадкові значення з відповідного проміжку $[0, i_{max}]$, $[0, j_{max}]$ та $[0, k_{max}]$.

Таким чином відбувається мутація лише тих параметрів, що може бути змінено (координат розташування контейнерів), і не відбувається мутації параметрів, що змінити не можна (розмір, клас небезпечних вантажів, тощо).

4.2.4 Елітизм та формування алгоритму

Елітизм – це модифікація генетичного алгоритму, що змушує його зберігати декілька найкращих особин між поколіннями. Такі особини називаються елітами. Так як процес вибору батьків має ймовірнісний характер, то при використанні звичайного алгоритму такі особини можуть бути втрачені. Елітизм також застосовується у інших алгоритмах, таких як мурашиний, метод рою часток тощо.

Більшість генетичних алгоритмів, описаних у літературі, оперують поколіннями: кожне нове покоління містить лише нащадків батьків попереднього покоління. При застосуванні елітизму послідовні покоління містять деяку частину особин попередніх. Однією з альтернатив традиційного алгоритму є генетичний алгоритм стаціонарного стану. У ньому використовується лише одна популяція, а нащадки, отримані після схрещення батьків, замінюють декількох особин з найменшою допасованістю. Переваги таких алгоритмів були показані у [127]–[130].

Таким чином для розв’язання поставленої задачі використовується генетичний алгоритм стаціонарного стану з використанням вищевказаних функцій допасованості, селекції, схрещення та мутації (Рис. 4.8).

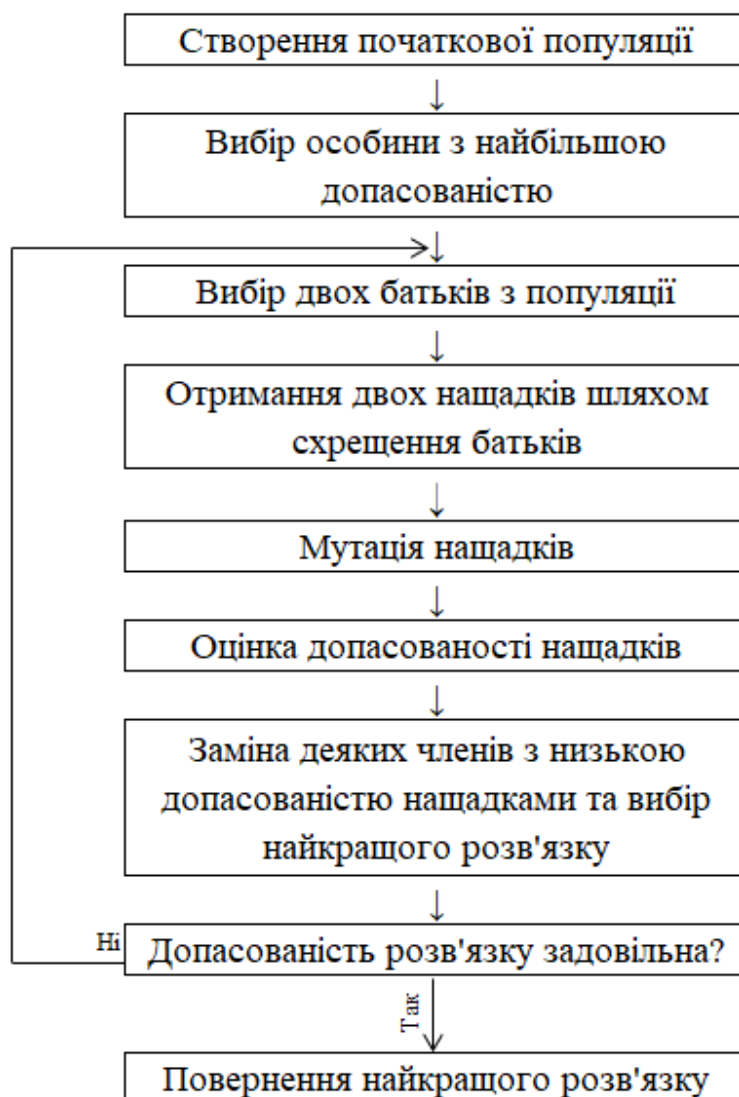


Рис. 4.8 Генетичний алгоритм стаціонарного стану

4.3 Висновки за розділом 4

Задача розташування деякої кількості контейнерів у вантажному просторі судна є NP повною, проте загальна кількість варіантів розташування робить використання точних методів непрактичними.

Серед неточних методів було обрано генетичний алгоритм через його гнучкість, точність та історію успішного використання для розв'язання подібних задач.

Використання генетичного алгоритму вимагає розгляд та розробку відповідних функцій допасованості, схрещення, мутації та відбору для розв'язання задачі у поставленому вигляді, а також вибір придатних структур для представлення даних при розв'язанні.

Формування особин відбувається у вигляді двовимірних матриць, які містять координати окремих контейнерів.

Функція допасованості оцінює загальну кількість невірно розташованих контейнерів щодо кожної нерівності, використовуючи математичну модель, приведену у підрозділі 3.2, обмеження якої зведено до вигляду матриці, описаної у підрозділі 3.3.

Функція схрещення оперує безпосередньо наборами координат, що дозволяє обирати найбільш ефективні розміщення.

Функція мутації змінює позиції окремих контейнерів у межах припустимих значень координат.

Селекція відбувається за допомогою турнірного відбору, що дозволяє більш точно обирати найкращих особин.

Для розв'язання задачі використовується генетичний алгоритм стаціонарного стану, який дозволяє поступово модифікувати популяцію, впроваджуючи до неї нових більш допасованих особин та усуваючи менш допасованих.

РОЗДІЛ 5. ВДОСКОНАЛЕННЯ РОЗТАШУВАННЯ КОНТЕЙНЕРІВ ЩОДО ПАРАМЕТРІВ МОРЕХІДНОСТІ

5.1 Остійність

Міжнародні вимоги щодо остійності судна вказані у «Кодексі остійності суден у непошкодженому стані» [131], який є обов'язковим згідно до конвенції SOLAS. Цей кодекс містить критерії остійності для вантажних та пасажирських суден, о яких відносяться наступні:

- 1) Площа поверхні під кривою відновлюючого моменту на діаграмі статичної остійності має бути не менше 0.055 м-р до кута крену 30° та не менше 0.09 м-р до кута крену 40° або до кута заливання, якщо такий кут менше 40° . Окрім цього площа під кривою між кутами 30° і 40° , або 30° та кутом заливання, має бути не менше 0.03 м-р.
- 2) Плече відновлюючого моменту GZ має бути не менш 0.2 м при куті крену 30° або більше.
- 3) Максимальне плече повинно відповідати куту крену не менш 25° , бажано не менш 30° .
- 4) Початкова метацентрична висота h_0 повинна бути не менш 0.15 м.

Остійність судна визначається статичним моментом Mz , тому для забезпечення виконання критеріїв остійності необхідно, щоб цей момент знаходився у деяких межах Mz_{min} та Mz_{max} . Ці значення знаходяться через діаграму статичної остійності та залежать від водотоннажності судна.

При завантаженні судна Mz визначається наступним чином:

$$Mz = Mz_0 + Mz_S + Mz_B + Mz_C$$

Де Mz_0 – момент порожнього судна, Mz_S – момент запасів, Mz_B – момент баласту, а Mz_C – момент вантажу. При завантаженні судна r контейнерами:

$$Mz_c = \sum_{n=0}^{r-1} (P_n * z_n)$$

Де r – загальна кількість контейнерів, P_n – вага n -го контейнеру, а z_n – апліката його центра ваги.

5.2 Диферент та міцність

Диферент та загальна міцність судна залежать від поздовжнього розподілу навантаження на корпус.

Диферент визначається статичним моментом Mx .

$$Mx = Mx_0 + Mx_S + Mx_B + Mx_C$$

Де Mx_0 – момент порожнього судна, Mx_S – момент запасів, Mx_B – момент баласту, а Mx_C – момент вантажу. При завантаженні судна r контейнерами:

$$Mx_C = \sum_{n=0}^{r-1} (P_n * x_n)$$

Де x_n – абсцисса центра ваги контейнеру.

Для забезпечення деяких граничних значень диференту $d \in [d_{min}, d_{max}]$, необхідно визначити границі моментів Mx_{Cmin} та Mx_{Cmax} . Вони знаходяться наступним чином:

$$Mx = D(x_g - x_c)$$

Де x_g та x_c є абсцисами центру ваги та величини судна, що відповідає даній водотоннажності D , відповідно.

$$d = \frac{Mx}{\bar{M}}$$

Де $\bar{M}$ – момент, що диферентує судно на 1 см.

Звідси маємо

$$x_g = x_c + \frac{d\bar{M}}{D}$$

Одночасно:

$$x_g = \frac{Mx_0 + Mx_S + Mx_B + Mx_C}{D}$$

Так як у даній задачі виконуються операції лише над вантажем, то необхідно виразити Mx_C .

$$Mx_C = x_g * D - (Mx_0 + Mx_S + Mx_B)$$

$$Mx_C = x_c * D + d * \bar{M} - (Mx_0 + Mx_S + Mx_B)$$

Звідси:

$$Mx_{Cmin} = x_c * D + d_{min} * \bar{M} - (Mx_0 + Mx_S + Mx_B)$$

$$Mx_{Cmax} = x_c * D + d_{max} * \bar{M} - (Mx_0 + Mx_S + Mx_B)$$

Розрахунок загальної міцності корпусу відбувається щодо міделю судна. Таким чином для перевірки необхідно просумувати модулі моментів щодо міделю:

$$Mxb_c = \sum_{n=0}^{r-1} (P_n * |x_n|)$$

$$Mxb = Mxb_0 + Mxb_S + Mxb_B + Mxb_C$$

Mxb має знаходитись у деяких межах: $Mxb \in [Mxb_{min}, Mxb_{max}]$, які визначаються з даних судна для даного дедвейту.

Перевірка місцевої міцності виконується для кожного стеку контейнерів, вага котрого має бути меншою деякого параметра Ps_{max} . Позначимо матрицю, що містить ваги кожного стеку як Ps . Тоді:

$$Ps_{i,j} = \sum_k (P_{i,j,k}) \quad \forall i,j$$

$$Ps_{i,j} \leq Ps_{max} \quad \forall i,j$$

Де x_i – абсциса позицій з індексом i , $P_{i,j,k}$ – вага контейнеру, розташованого у позиції (i,j,k) .

5.3 Крен

Аналогічно до попередніх підрозділів, крен судна залежить від кренуючого моменту, який можна обчислити наступним чином:

$$My = My_0 + My_S + My_B + My_C$$

Де My_0 – момент порожнього судна, My_S – момент запасів, My_B – момент баласту, а My_C – момент вантажу.

Момент, що крінить судно на один градус, може бути розрахований за формулою:

$$My_{1^\circ} = \frac{h_0 * D}{57.3}$$

Звідси момент, що крінить судно на θ градусів:

$$My_{\theta^\circ} = \theta \frac{h_0 * D}{57.3}$$

Як правило, крену необхідно уникати, проте можна задати деякий невеликий проміжок $\theta \in (\theta_{min}, \theta_{max})$, що буде визначати припустимі значення крену. Наприклад, $(-0.5^\circ, 0.5^\circ)$. Тоді обмеження щодо кренуючого моменту $My \in (My_{min}, My_{max})$ будуть виглядати наступним чином:

$$My_{min} = \theta_{min} \frac{h_0 * D}{57.3}$$

$$My_{max} = \theta_{max} \frac{h_0 * D}{57.3}$$

5.4 Модифікація розв'язку з урахуванням морехідних якостей

Після отримання розв'язку, що задовольняє обмеження математичної моделі, необхідно переконатися, що вимоги щодо морехідних якостей також виконуються.

Хоча формування такого розв'язку відбувається відносно деякої множини конкретних контейнерів, це виконується для полегшення розв'язання задачі, сама модель таких обмежень не ставить. Модель вимагає тільки те, що контейнери, які

мають задані характеристики, мають бути розміщені у отриманих позиціях, ці характеристики не визначають окремі контейнери.

Розв'язок моделі має вигляд одновимірної матриці. Для зручності її можна перетворити в двовимірну таким чином, що отримана матриця містить лише список позицій, у яких розташовані контейнери (з одновимірної матриці беруться лише одиниці). Тоді така двовимірна матриця $S1$ буде мати наступний вигляд (Рис. 5.1).

$S1$

t_0	c_0	i_0	j_0	k_0
t_1	c_1	i_1	j_1	k_1
...	...	...	...	...
t_n	c_n	i_n	j_n	k_n

Рис. 5.1 Розв'язок моделі у вигляді двовимірної матриці $S1$

Після отримання розв'язку щодо заданих контейнерів необхідно визначити моменти Mx , Mxb , My та Mz , а також ваги кожного зі стеків контейнерів та перевірити їх задовільність щодо поставлених обмежень.

У рамках цієї роботи розглядаються лише моменти, що створюються вантажами, тому можна записати, що розв'язок є припустимим якщо виконуються наступні вимоги:

$$Mx_c \in [Mx_{c_{min}}, Mx_{c_{max}}]$$

$$Mxb_c \in [Mxb_{c_{min}}, Mxb_{c_{max}}]$$

$$My_C \in [My_{C_{min}}, My_{C_{max}}]$$

$$Mz_C \in [Mz_{C_{min}}, Mz_{C_{max}}]$$

Проте припустимість розв'язку не є гарантованою, і у разі його незадовільності необхідно визначити, чи можна його привести до припустимого у межах існуючого розв'язку моделі $S1$.

У випадку, коли один з моментів не входить до області припустимих значень, вказаних вище, необхідно визначити межі моментів для такого розв'язку. Для виконання такої перевірки, множина завантажуваних контейнерів N розбивається на підмножини, поміж якими ці контейнери можна вільно переміщувати. Кількість таких множин m визначається за формулою:

$$m = (t_{max} + 1) * (c_{max} + 1) * P,$$

де P – кількість портів призначення.

Таким чином,

$$N_q \subseteq N \forall q \in [0, m - 1],$$

Де N_q – відповідна підмножина контейнерів потужністю r_q , які мають однакові параметри t і c .

Плечі моментів контейнерів для розглядуемого судна можна представити у вигляді векторів $\bar{X}$, $\bar{Y}$ та $\bar{Z}$. Таким чином, вважаючи, що центр ваги кожного контейнера знаходиться у його центрі, кожному i моделі відповідає плечо $\bar{X}_i$, кожному j – плечо $\bar{Y}_j$, а кожному k відповідає плечо $\bar{Z}_k$. Якщо контейнер – FEU, тобто займає дві позиції по i , то він розглядається як два TEU контейнери, ваги яких рівні половині ваги FEU. Кожна підмножина контейнерів містить набір координат з розв'язку $S1$, у які можна поставити контейнери даного підтипу. Зведемо ці координати до матриці L_q . Кожен рядок цієї матриці відповідає деякому місцю під контейнер, а кожна колонка – відповідній координаті (i , j та k). Ваги контейнерів підмножини N_q записуються до матриці P_q . Тоді ці матриці будуть мати наступний вигляд (Рис. 5.2).

Вторинний розв'язок задачі відповідає на питання, яке полягає у тому, як розмістити контейнери з підмножини N_q за координатами з матриці L_q , або відносно матриці P_q : яким чином відсортувати матрицю P_q так, щоб розв'язок був припустимим для кожного набору координат L_{q_n} у якому розміщений контейнер вагою P_{q_n} . Такий розв'язок можна представити у вигляді такої двовимірної матриці $S2$, що для кожного елементу матриці $S2_{q_n}$ контейнер вагою $P_{q_{S2_{q_n}}}$ знаходиться у позиції L_{q_n}

L_q			P_q
i_0	j_0	k_0	P_0
i_1	j_1	k_1	P_1
...	...	...	...
i_{r_q-1}	j_{r_q-1}	k_{r_q-1}	P_{r_q-1}

Рис. 5.2 Матриці L_q та P_q

Подальші обчислення виконуються вважаючи, що кожна матриця P_q відсортована таким чином, що кожний елемент P_{q_n} розміщений за номером $S2_{q_n}$.

Загальні моменти, викликані вантажами, можна поділити поміж цими підмножинами:

$$Mx_{Cq} = \sum_{n=0}^{r_q-1} (P_{q_n} * \bar{X}_{L_{q_{n,0}}}) \quad \forall q \in [0, m-1]$$

$$\begin{aligned}
Mx_C &= \sum_{q=0}^{m-1} (Mx_{C_q}) \\
Mxb_{C_q} &= \sum_{n=0}^{r_q-1} (P_{q_n} * |\bar{X}_{L_{q_{n,0}}}|) \forall q \in [0, m-1] \\
Mxb_C &= \sum_{q=0}^{m-1} (Mxb_{C_q}) \\
My_{C_q} &= \sum_{n=0}^{r_q-1} (P_{q_n} * \bar{Y}_{L_{q_{n,1}}}) \forall q \in [0, m-1] \\
My_C &= \sum_{q=0}^{m-1} (My_{C_q}) \\
Mz_{C_q} &= \sum_{n=0}^{r_q-1} (P_{q_n} * \bar{Z}_{L_{q_{n,2}}}) \forall q \in [0, m-1] \\
Mz_C &= \sum_{q=0}^{m-1} (Mz_{C_q})
\end{aligned}$$

Для кожної підмножини можна знайти як мінімальні, так і максимальні моменти для даного розв'язку $S1$.

Нехай $L_q id$ – матриця, що представляє собою матрицю L_q , відсортовану за i за спаданням. Тобто:

$$L_q id_{n-1,0} \geq L_q id_{n,0} \geq L_q id_{n+1,0} \forall n \in [1, r_q - 2]$$

Аналогічно $L_q ibd$ – матриця, що представляє собою матрицю L_q , відсортовану за $|\bar{X}_i|$ за спаданням, $L_q jd$ – матриця, що представляє собою матрицю L_q , відсортовану за j за спаданням, а $L_q kd$ – матриця, що представляє собою матрицю L_q , відсортовану за k за спаданням:

$$|\bar{X}_{L_q ibd_{n-1,0}}| \geq |\bar{X}_{L_q ibd_{n,0}}| \geq |\bar{X}_{L_q ibd_{n+1,0}}| \forall n \in [1, r_q - 2]$$

$$L_q j d_{n-1,1} \geq L_q j d_{n,1} \geq L_q j d_{n+1,1} \quad \forall n \in [1, r_q - 2]$$

$$L_q k d_{n-1,2} \geq L_q k d_{n,2} \geq L_q k d_{n+1,2} \quad \forall n \in [1, r_q - 2]$$

$P_q d$ – матриця P_q , відсортована за спаданням, а $P_q a$ – матриця P_q , відсортована за зростанням, тобто:

$$P_q d_{n-1} \geq P_q d_n \geq P_q d_{n+1} \quad \forall n \in [1, r_q - 2]$$

$$P_q a_{n-1} \leq P_q a_n \leq P_q a_{n+1} \quad \forall n \in [1, r_q - 2]$$

Таким чином можна обчислити максимальні і мінімальні моменти, які можуть бути викликані q -ю партією контейнерів: $L_q i d_{n,0}$

$$Mx'_{max_q} = \sum_{n=0}^{r_q-1} (P_q d_n * \bar{X}_{L_q i d_{n,0}}) \quad \forall q \in [0, m - 1]$$

$$Mx'_{min_q} = \sum_{n=0}^{r_q-1} (P_q a_n * \bar{X}_{L_q i d_{n,0}}) \quad \forall q \in [0, m - 1]$$

$$Mxb'_{max_q} = \sum_{n=0}^{r_q-1} (P_q d_n * |\bar{X}_{L_q i b d_{n,0}}|) \quad \forall q \in [0, m - 1]$$

$$Mxb'_{min_q} = \sum_{n=0}^{r_q-1} (P_q a_n * |\bar{X}_{L_q i b d_{n,0}}|) \quad \forall q \in [0, m - 1]$$

$$My'_{max_q} = \sum_{n=0}^{r_q-1} (P_q d_n * \bar{Y}_{L_q j d_{n,1}}) \quad \forall q \in [0, m - 1]$$

$$My'_{min_q} = \sum_{n=0}^{r_q-1} (P_q a_n * \bar{Y}_{L_q j d_{n,1}}) \quad \forall q \in [0, m - 1]$$

$$Mz'_{max_q} = \sum_{n=0}^{r_q-1} (P_q d_n * \bar{Z}_{L_q k d_{n,2}}) \quad \forall q \in [0, m - 1]$$

$$Mz'_{min_q} = \sum_{n=0}^{r_q-1} (P_q a_n * \bar{Z}_{L_q k d_{n,2}}) \quad \forall q \in [0, m-1]$$

Звідси мінімальні та максимальні моменти, що викликані множиною контейнерів для даного розв'язку $S1$:

$$Mx'_{min} = \sum_{q=0}^{m-1} (Mx'_{min_q})$$

$$Mx'_{max} = \sum_{q=0}^{m-1} (Mx'_{max_q})$$

$$Mxb'_{min} = \sum_{q=0}^{m-1} (Mxb'_{min_q})$$

$$Mxb'_{max} = \sum_{q=0}^{m-1} (Mxb'_{max_q})$$

$$My'_{min} = \sum_{q=0}^{m-1} (My'_{min_q})$$

$$My'_{max} = \sum_{q=0}^{m-1} (My'_{max_q})$$

$$Mz'_{min} = \sum_{q=0}^{m-1} (Mz'_{min_q})$$

$$Mz'_{max} = \sum_{q=0}^{m-1} (Mz'_{max_q})$$

Можна стверджувати, що розв'язок $S1$ не є припустимим у разі, коли виконується принаймні одна з наступних умов:

$$[Mx'_{min}, Mx'_{max}] \cap [Mx_{c_{min}}, Mx_{c_{max}}] = \emptyset$$

$$[Mxb'_{min}, Mxb'_{max}] \cap [Mxb_{c_{min}}, Mxb_{c_{max}}] = \emptyset$$

$$[My'_{min}, My'_{max}] \cap [My_{Cmin}, My_{Cmax}] = \emptyset$$

$$[Mz'_{min}, Mz'_{max}] \cap [Mz_{Cmin}, Mz_{Cmax}] = \emptyset$$

Якщо не одна з зазначених умов не виконується, то виконується знаходження припустимого розв'язку $S2$.

Окрім моментів необхідно виконати перевірку місцевої міцності. Нехай k'_{max} – максимальне значення k для розв'язку, тобто

$$k'_{max} = \max(L_{q_{n,2}}) \quad \forall q \in [0, m-1], n \in [0, r_q-1]$$

Тоді максимально можливе навантаження на палубу для такого розв'язку $S1$ буде наступним:

$$Ps'_{max} = \sum_{n=0}^{k'_{max}} (Pd_n)$$

Де Pd_n – матриця, що містить усі ваги завантажуваних контейнерів, відсортована за спаданням. Якщо $Ps'_{max} \leq Ps_{max}$, то перевірку на місцеву міцність проводити не потрібно.

Для m підмножин N_q загальна кількість розміщень дорівнює

$$\sum_{q=0}^{m-1} (r_q!)$$

Звідси очевидно, що хоча така задача розміщення є простішою за розглянуту раніше, перебір усіх можливих розв'язків $S2$ залишається незастосовним.

Для розв'язання цієї частини також використовується генетичний алгоритм, проте з іншими функціями допасованості, мутації та схрещення.

Функція допасованості для розв'язку $S2$ повинна враховувати відповідні моменти Mx_C , Mxb_C , My_C та Mz_C , а також вагу кожного стеку контейнерів, якщо $Ps'_{max} \leq Ps_{max}$ не виконується.

За оптимальні прийняті такі моменти, що знаходяться посередині припустимих меж, тобто:

$$\begin{aligned}
Mx_{c\text{opt}} &= \frac{Mx_{c\min} + Mx_{c\max}}{2} \\
Mxb_{c\text{opt}} &= \frac{Mxb_{c\min} + Mxb_{c\max}}{2} \\
My_{c\text{opt}} &= \frac{My_{c\min} + My_{c\max}}{2} \\
Mz_{c\text{opt}} &= \frac{Mz_{c\min} + Mz_{c\max}}{2}
\end{aligned}$$

Тоді різниці між оптимальними та дійсними моментами:

$$\begin{aligned}
\Delta Mx_c &= |Mx_{c\text{opt}} - Mx_c| \\
\Delta Mxb_c &= |Mxb_{c\text{opt}} - Mxb_c| \\
\Delta My_c &= |My_{c\text{opt}} - My_c| \\
\Delta Mz_c &= |Mz_{c\text{opt}} - Mz_c|
\end{aligned}$$

Елементи матриці навантажень стеків контейнерів на палубу для даного розв'язку обчислюються наступним чином (спочатку вони дорівнюють нулю):

$$PS_{L_{q_{n,0}}, L_{q_{n,1}}} = PS_{L_{q_{n,0}}, L_{q_{n,1}}} + P_{q_n} \quad \forall q \in [0, m-1], n \in [0, r_q-1]$$

Нехай є матриця PSk , елементи якої обчислюються так:

$$PSk_{i,j} = \frac{PS_{i,j} - PS_{\max} + |PS_{i,j} - PS_{\max}|}{2} \quad \forall i, j$$

Тобто вона містить лише додатні різниці між навантаженнями, викликаними стеками та припустимою місцевою міцністю. Розв'язок є припустимим лише тоді, коли всі елементи цієї матриці є нулями, тобто сума її елементів дорівнює нулю:

$$SPsk = \begin{cases} 0, \text{ якщо } \sum_i \sum_j PSk_{i,j} > 0 \\ 1, \text{ інакше} \end{cases}$$

Беручи до уваги вказане вище, функція допасованості прийнята наступною:

$$fg = \frac{SPsk}{(1 + \Delta Mx_c + \Delta Mxb_c + \Delta My_c + \Delta Mz_c)}$$

Таким чином, чим ближчі моменти, викликані даним розміщенням, до оптимальних, тим вище допасованість такого розв'язку, а у разі невиконання вимог за місцевою міцністю допасованість розв'язку різко спадає. $fg \in (0,1]$.

Функція мутації оперує матрицею $S2$, специфіка якої не дозволяє їй мати елементи, що повторюються у межах $S2_q$. Тому, при виконанні мутації, замість модифікації одного елемента, виконується обмін позицій деяких елементів $S2_q$, що дозволяє залишати множину елементів незмінною.

Схрещення має ту саму проблему: множина елементів $S2_q$ повинна залишатись незмінною. Тому воно виконується наступним чином:

- 1) Беруться двоє батьків $S2^1$ і $S2^2$.
- 2) Батьки клонуються до нащадків $S2^{1'}$ і $S2^{2'}$ відповідно.
- 3) Для кожного елемента $S2^{1'}_{q_n}$ з деякою ймовірністю обирається значення $S2^2_{q_n}$, це значення знаходиться у матриці $S2^{1'}_q$, і виконується обмін поточного елемента зі знайденим.

Таким чином операція схрещення може брати найвдаліші позиції з батьків, зберігаючи відповідні множини елементів незмінними.

5.5 Початкове розташування контейнерів

Використання генетичного алгоритму для знаходження припустимого розташування контейнерів вимагає створення початкової популяції таких розташувань. Хоча у більшості генетичних алгоритмів така популяція створюється випадково, це викликає набагато довший процес знаходження розв'язків. Чим менше контейнерів, що розташовані невірно у початковій популяції, тим швидше буде знайдено необхідний розв'язок. Тому необхідно створити таке початкове розташування контейнерів на судні, яке буде намагатися мінімізувати кількість невірних розміщень.

Пропонується наступний алгоритм створення початкового розв'язку:

- 1) Виконати сортування усіх контейнерів, що необхідно завантажити за:
 - a. портами призначення (за спаданням),
 - b. класами небезпечних вантажів (за спаданням),
 - c. вагою (за спаданням).
- 2) Почати розташування контейнерів, починаючи з початку списку, у такі слоти, щоб:
 - a. мінімізувати різниці між оптимальними та кінцевими моментами Mx_C , Mxb_C , My_C та Mz_C ,
 - b. не розташовувати TEU на FEU,
 - c. виконувалась сегрегація наявних небезпечних вантажів,
 - d. не перевищувалась місцева міцність палуби,
 - e. за можливості уникати розташування FEU на TEU.

Розроблено функцію, яка визначає придатність окремого контейнеру для розташування у визначене місце.

Для цього спочатку визначається матриця O , яка містить усі незайняті позиції контейнерів (i, j, k) , тобто такі, у яких можна розмістити контейнери на палубі або інших контейнерах. Зіставний список C містить координати вже розміщених контейнерів, згруповані за класом небезпечних вантажів. Тоді C_q містить список позицій контейнерів, що мають небезпечні вантажі класу q у вигляді (i, j, k) . di, dj і dk – мінімальні відстані у TEU по i, j та k відповідно між контейнерами з небезпечними вантажами класів $c1$ та $c2$.

Використовуються наступні параметри:

$$pi1 = \sum_{q=0}^{\#C-1} \sum_{q'=0}^{\#C_q-1} \frac{di - |C_{q_{q'},0} - i| + |di - |C_{q_{q'},0} - i||}{2}$$

$\#C$ – кількість елементів C , $\#C_q$ – кількість елементів C_q . Якщо для даного контейнеру немає несумісних небезпечних вантажів для даної позиції щодо i , то $pi = 0$.

Додатково для FEU:

$$i2 = i + 1 - i\%2 * 2,$$

де $\%$ - знак операції остачі від цілочисельного ділення.

$$pi2 = \sum_{q=0}^{\#C-1} \sum_{q'=0}^{\#C_q-1} \frac{di - |C_{q_{q'},0} + 1 - i2| + |di - |C_{q_{q'},0} + 1 - i2||}{2}$$

Звідси:

$$pi = \begin{cases} \text{Max}(p1, p2), & \text{якщо } c = 1, \\ p1, & \text{якщо } c = 0. \end{cases}$$

Для інших координат:

$$pj = \sum_{q=0}^{\#C-1} \sum_{q'=0}^{\#C_q-1} \frac{dj - |C_{q_{q'},1} + 1 - j| + |dj - |C_{q_{q'},1} + 1 - j||}{2}$$

$$pk = \sum_{q=0}^{\#C-1} \sum_{q'=0}^{\#C_q-1} \frac{dk - |C_{q_{q'},2} + 1 - k| + |dk - |C_{q_{q'},2} + 1 - k||}{2}$$

Якщо для даного контейнеру відсутні несумісні небезпечні вантажі для обраної позиції щодо j або k , то $pj = 0$ і $pk = 0$ відповідно.

$$pt = \begin{cases} 0, & \text{якщо } pi > 0, pj > 0 \text{ та } pk > 0 \\ 1, & \text{інакше} \end{cases}$$

Розміщення кожного контейнеру із відсортованого списку відбувається шляхом послідовної підстановки у кожну позицію з матриці O , після чого визначається його допасованість за наступною функцією:

$$fg = \frac{Psc * \sqrt{|100 * t - (t(k-1) - 1)^2|} * pt}{(1 + \Delta Mx_c + \Delta Mxb_c + \Delta My_c) * \Delta Mz_c^3 * (1 + k)}$$

$$P_{sc} = \begin{cases} 1, \text{ якщо } Psk_{i,j} > 0 \\ 0, \text{ інакше} \end{cases}$$

де $t(k-1)$ – параметр t контейнеру, що знаходиться у позиції $(i, j, k-1)$, або 0, якщо його не існує (контейнер ставиться на палубу).

Запропонована функція має штраф на розстановку FEU контейнерів поверх TEU контейнерів, а при постановці TEU поверх FEU повертає нуль. Також вона повертає нуль при невиконанні вимог щодо розміщення небезпечних вантажів або при перевищенні припустимої місцевої міцності. Щодо інших параметрів: чим ближчі моменти, викликані розміщенням даного контейнеру, до оптимальних, тим вище значення функції $fg \in [0,1]$. Варто зазначити, що оптимальні моменти, які використовуються для розрахунку значень допасованості конкретного розміщення, у цьому підрозділі не мають дорівнювати оптимальним.

Отримане таким чином початкове розміщення дає можливість мінімізувати кількість циклів алгоритмів з оптимізації вантажного плану.

5.6 Висновки за розділом 5

Параметри остійності непошкоджених суден регулюються ІМО і є обов'язковими до виконання для усіх торгових суден, що виконують міжнародні рейси.

Для забезпечення виконання параметрів остійності, міцності та диференту було розроблено алгоритм, який дозволяє переміщувати контейнери у межах множин позицій, які задовольняють як структурним вимогам щодо розміщення вантажів, так і порядку портів вивантаження. Цей алгоритм можна налаштовувати за допомогою зміни множників перед складовими параметрами, що дозволяє змінювати їх вагу.

Алгоритм підбору розв'язку для виконання обмежень розробленої математичної моделі оперує окремими контейнерами та їх позиціями, є стохастичним, тому підбір

найбільш придатного для моделі початкового розташування може значно скоротити швидкість його роботи.

Розроблений алгоритм початкового розташування працює шляхом сортування списку контейнерів за різними параметрами, після чого відбувається почергова оцінка можливих позицій для розташування кожного окремого контейнеру. Запропонований алгоритм також можна налаштовувати у залежності від потреб конкретної задачі за допомогою зміни множників перед відповідними параметрами.

РОЗДІЛ 6. ОЦІНКА ЕФЕКТИВНОСТІ РОЗРОБЛЕНИХ МЕТОДІВ ЗА ДОПОМОГОЮ ІМІТАЦІЙНОГО МОДЕЛЮВАННЯ ЗАВАНТАЖЕННЯ

6.1 Початкове розташування та подальша оптимізація завантаження

Для перевірки розробленої математичної моделі та методів було проведене імітаційне моделювання завантаження судна «Камі» партією, що складається із 750 контейнерів, а саме 690 TEU та 60 FEU.

Розглядуваний у даному розділі випадок містить один (перший) порт завантаження «А» та три (послідовні) порти вивантаження «В», «С» та «D» (див. Таблиця 6.1), при цьому приведена партія контейнерів може бути завантажена на судно повністю, не перевищуючи його місткість, яка становить 1076 TEU або 501 FEU, або максимальну літню водотоннажність, що дорівнює 36235 т. Такі випадки часто зустрічаються у роботі невеликих контейнерних суден, коли необхідно завантажити відому партію товару, замість вибору найбільшої множини контейнерів з усіх наявних на терміналі.

Таблиця 6.1

Інформація по контейнерах

Порт	TEU, шт.	Вага, т	FEU, шт.	Вага, т
А	118	504.6	22	211.4
В	224	1434.1	12	159.6
С	198	1927.6	6	110.2
D	150	2287	20	583.8
Всього	690	6153.3	60	1065

У приведеному випадку було враховано необхідні судові запаси для переходу, які складаються з важкого та дизельного палив, масел та прісної води. Початкове розташування цих запасів по відповідних танках приведено у Таблиця 6.2, де також вказано їх наповненість, у відсотках, для врахування у розрахунках початкової остійності.

Таблиця 6.2

Початкове розташування судових запасів

Назва танку	Маса, т	x, м	z, м	Mx, тм	Mz, тм	Mh, тм	Наповненість, %
HFO tank #3 C	196.93	-39.9	9.51	-7857.3	1872.8	8.00	94.22
HFO tank #2 P/S	77.82	-41.35	8.8	-3217.9	684.8	0.00	98.51
HFO tank #2 S/S	77.82	-41.35	9.5	-3217.9	739.3	0.00	98.51
DO tank #2 C	29.15	-45.9	10.91	-1338.2	318.1	2.00	46.28
LO tank #3 C	15.76	-51.5	1.47	-811.6	23.2	0.00	98.50
LO tank #2 S/S	3.33	-51.35	6.8	-171.0	22.6	0.00	83.25
FW tank #1 C	4.80	24.4	4.24	117.1	20.4	579.00	2.25
FW tank #2 P/S	5.66	-54.3	1.33	-307.3	7.5	0.00	80.86
FW tank #2 S/S	18.72	-54.44	14.1	-1019.1	264.0	0.00	98.53
FW tank #3 C	69.94	-66.9	1.1	-4679.0	76.9	0.00	98.51
FW tank #4 C	33.49	-73.4	3.22	-2458.2	107.8	0.00	98.50
ВСЬОГО	533.42	-46.79	7.76	-24960.3	4137.3	589.00	-

Для приведенного розрахунку завантаження робиться припущення про неможливість проведення баластних операцій у подальших портах, звідси прийняття баласту відбувається до початку вантажних операцій у першому порту, т.я. судно має підтримувати задовільні параметри остійності та міцності у кожен момент часу. Розташування взятого баласту для покращення початкових параметрів морехідності приведено у Таблиця 6.3.

Таблиця 6.3

Початкове розташування баласту

Назва танку	Маса, т	x, м	z, м	Mx, тм	Mz, тм	Mh, тм	Наповненість, %
DBT #1	912.00	62.2	5.5	56726.4	5016.0	0.00	100.00
FPT	586.00	57.3	9.54	33577.8	5590.4	0.00	100.00
DBT #2 S/S	568.00	35.4	1.98	20107.2	1124.6	0.00	100.00
DBT #2 P/S	627.00	37.1	1.89	23261.7	1185.0	0.00	100.00
TST #2 S/S	186.00	29.9	9.05	5561.4	1683.3	0.00	100.00
TST #2 P/S	186.00	29.9	9.05	5561.4	1683.3	0.00	100.00
TST #3 S/S	671.00	7	10.14	4697.0	6803.9	0.00	100.00
TST #3 P/S	671.00	7	10.14	4697.0	6803.9	0.00	100.00
DBT #4 S/S	786.00	-20.9	2.67	-16427.4	2098.6	0.00	100.00
DBT #4 P/S	981.00	-23.1	2.31	-22661.1	2266.1	0.00	100.00
ВСЬОГО	6174.00	18.64	5.55	115101.4	34255.3	0.00	-

Таким чином, беручи до уваги масу та плечі судна порожнем, можна отримати початкові значення мас та моментів, які відповідають за параметри остійності, перед завантаженням (див. Таблиця 6.4).

Таблиця 6.4

Початкові моменти судна

Назва	Маса, т	x, м	z, м	Mx, тм	Mz, тм
Судно	9950	-14.4	12.31	-143280.0	122484.5
Мертвий запас	350.00	-15.60	12.80	-5460.0	4480.0
Судові запаси	533.42	-46.79	7.76	-24960.3	4137.3
Баласт	6174.00	18.64	5.55	115101.4	34255.3
Всього	17007.42	-3.45	9.72	-58598.9	165357.2

Ці маси та моменти використовуються надалі як точка відліку для подальших розрахунків. Морехідні якості для такого незавантаженого судна можна побачити на Рис. 6.1.

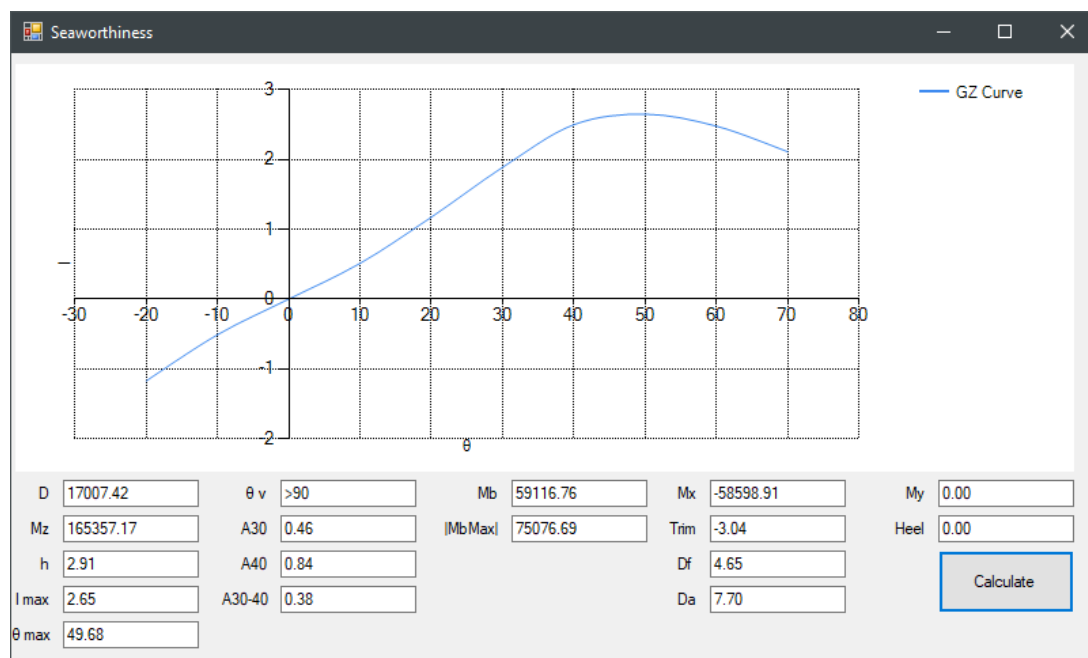


Рис. 6.1 Морехідні якості судна після розташування запасів та баласту

Для цілей даної роботи розглядається ситуація завантаження у першому порту, та тільки вивантаження у послідовних. Так як судно повинно задовольняти усім вимогам щодо остійності та міцності у кожному порту, то такі вимоги мають виконуватись для кожної партії контейнерів.

Спочатку відбувається завантаження контейнерів, що прямують до останнього порту за допомогою алгоритму, запропонованого у підрозділі 5.6. Вона складається з 150 TEU загальною вагою 2287 т та 20 FEU загальною вагою 583.8 т. Результати цього завантаження можна побачити на Рис. 6.2 та Рис. 6.3.

На рисунках білим кольором позначено контейнери, що не містять небезпечних вантажів. Фіолетовим кольором позначено контейнери, що містять небезпечні вантажі класу 5.2, а синім – контейнери, що містять небезпечні вантажі класу 4.1. Буква у

кожній комірці відповідає порту. FEU контейнери позначаються стандартним чином, тобто як TEU у непарних баях та хрестами у парних.

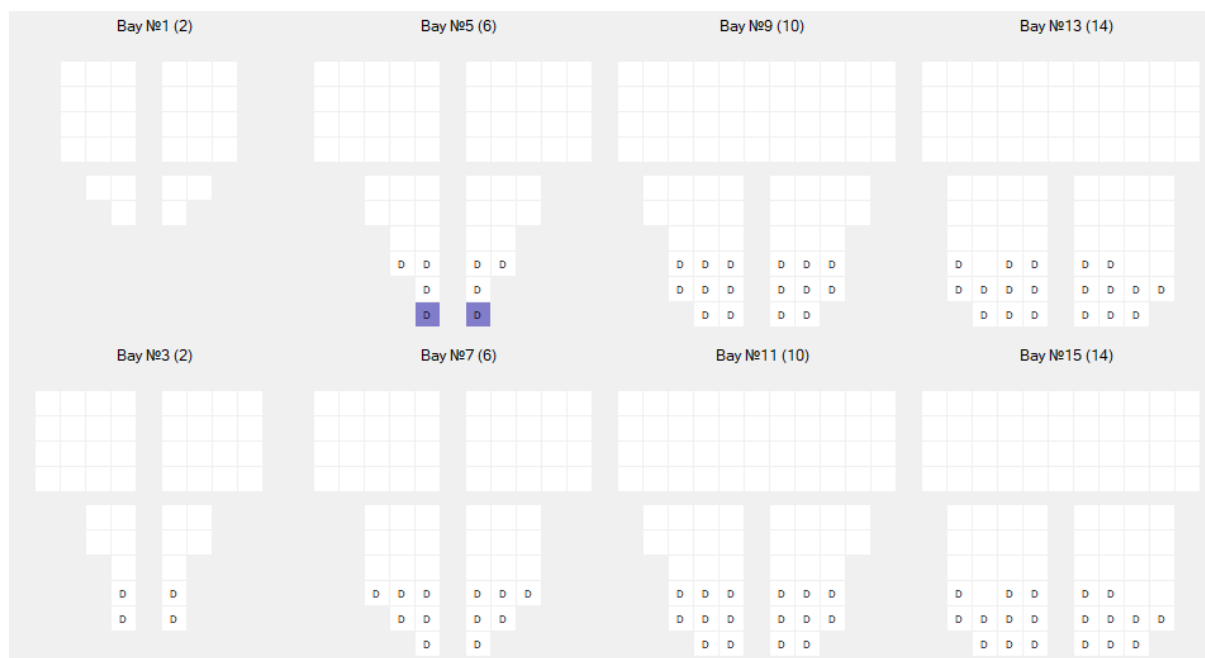


Рис. 6.2 Розташування першої партії контейнерів, частина 1

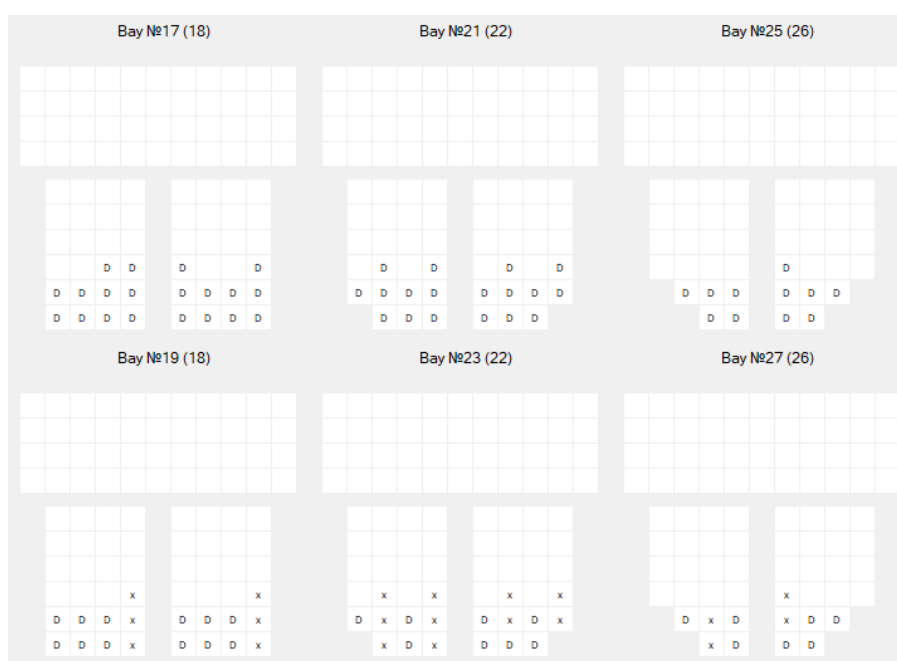


Рис. 6.3 Розташування першої партії контейнерів, частина 2

При цьому розміщенні морехідні якості розглядуваного судна мають наступний вигляд (Рис. 6.4):

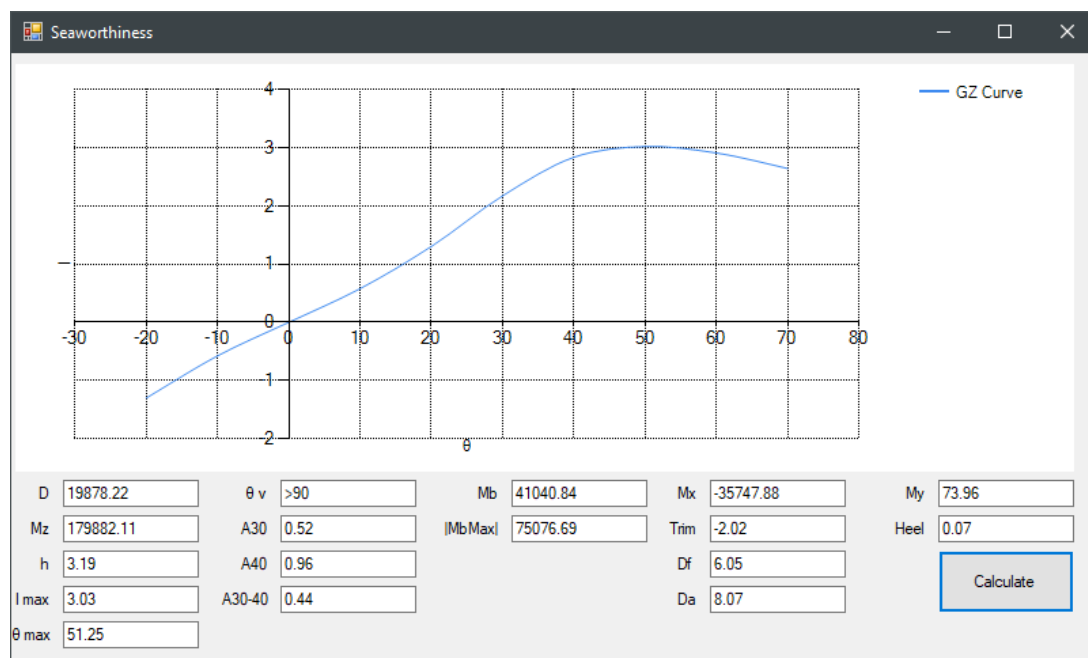


Рис. 6.4 Морехідні якості судна з першою партією вантажів без оптимізації

Як можна побачити з приведеного рисунку, параметри морехідності знаходяться у межах норми, проте значення диференту може бути занадто великим у залежності від характеру плавання. Для усунення цього недоліку контейнери, які мають однакові характеристики (такі як розмір, клас небезпечних вантажів тощо), але різну вагу варто поміняти місцями таким чином, щоб зменшити диферент, але при цьому забезпечити виконання вимог щодо морехідності судна. Ця задача розв'язується за допомогою застосування алгоритму, описаного у підрозділі 5.4, який розподіляє такі контейнери по групах та підбирає їх оптимальне розташування у межах таких груп. Обмін позицій контейнерів лише у межах груп забезпечує виконання вимог щодо взаємного розташування контейнерів відповідно їх типу та класу небезпечних вантажів. Результат роботи цього алгоритму приведений на Рис. 6.5.

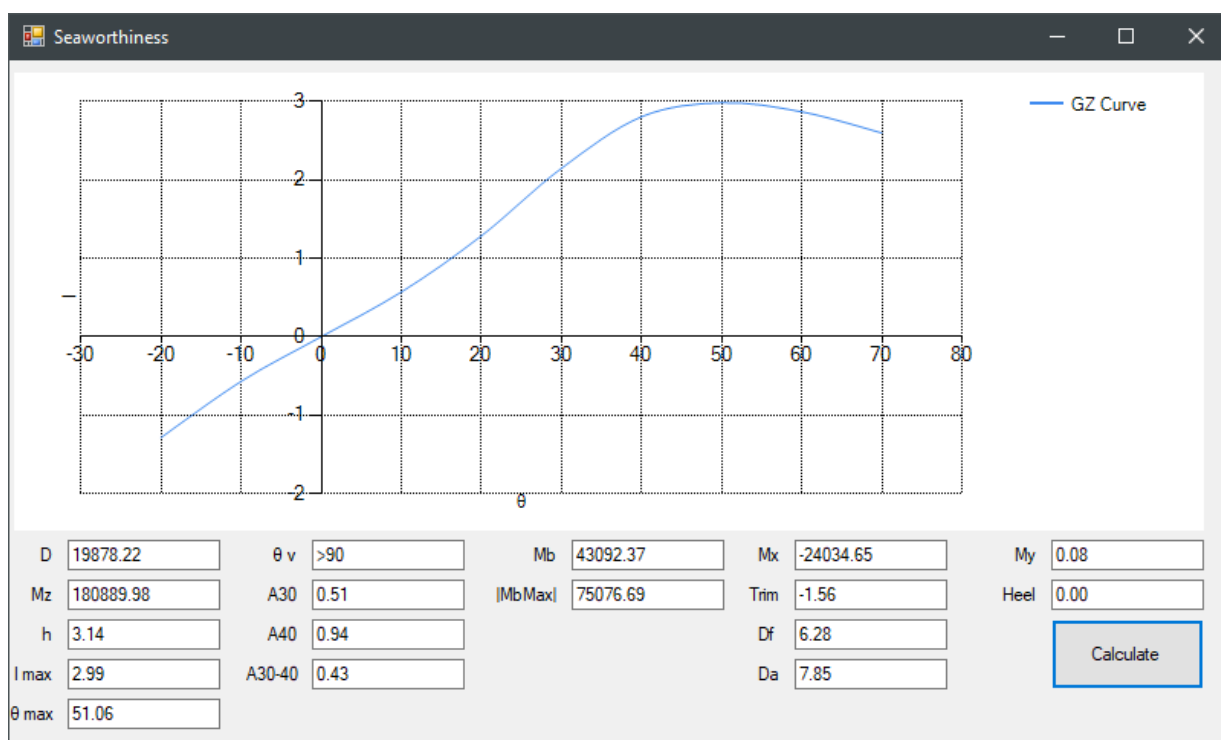


Рис. 6.5 Морехідні якості судна з першою партією вантажів після оптимізації

Як можна побачити на рисунку вище, диферент судна було покращено з -2.02 м до -1.56 м при дуже незначному зміні параметрів остійності, значення яких залишаються задовільними. Зменшення початкової метацентричної висоти має позитивний характер, так як занадто великий відновлюючий момент викликає зменшення періоду хитамиці та підвищення навантажень як на судові елементи, так і на кріплення контейнерів.

Результат завантаження другої партії контейнерів, що направлена у передостанній порт «В» та складається з 198 TEU загальною вагою 1927.6 т та 6 FEU загальною вагою 110.2 т, по бях судна за допомогою використання алгоритму, запропонованого у підрозділі 5.6, приведено нижче на Рис. 6.6 та Рис. 6.7.

Параметри морехідності судна після завантажень партіями контейнерів #1 та #2 та подальших оптимізацій за допомогою методу, приведенного у підрозділі 5.4, приведено у Таблиця 6.5.

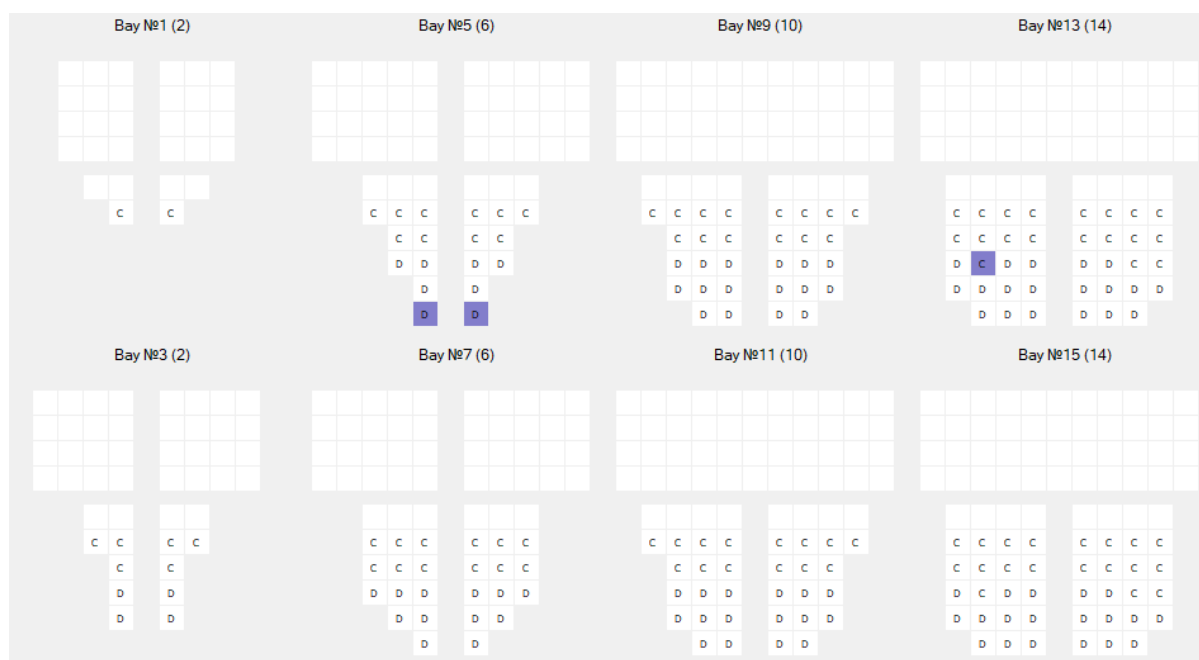


Рис. 6.6 Розташування другої партії контейнерів, частина 1

Приведене розташування було виконано без порушень вимог щодо розташування TEU поверх FEU, при цьому більшість контейнерів завантажено у найбільш нижні з доступних ярусів.

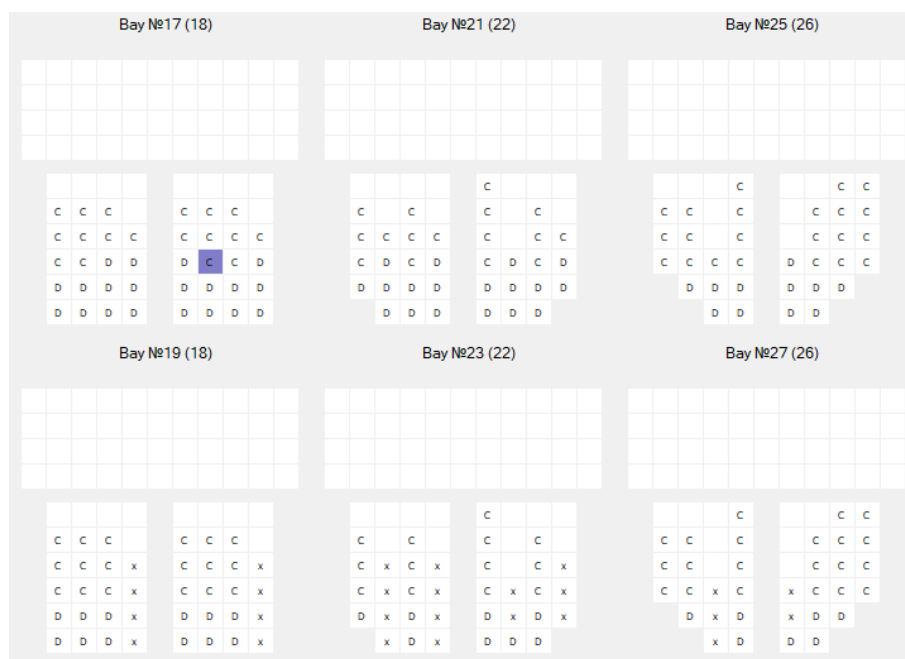


Рис. 6.7 Розташування другої партії контейнерів, частина 2

Таблиця 6.5

Параметри морехідності судна після завантажень #1 та #2

Параметр	#1	#1 опт.	#2	#2 опт.	Припустимі
D	19878.22	19878.22	21916.02	21916.02	≤ 36235
h	3.19	3.14	2.88	2.83	≥ 0.15
$l_{\max}$	3.03	2.99	2.76	2.72	≥ 0.2
$\theta_{\max}$	51.25	51.06	49.63	49.35	≥ 30
θ_v	>90	>90	>90	>90	≥ 60
A_{30}	0.52	0.51	0.48	0.47	≥ 0.055
A_{40}	0.96	0.94	0.88	0.87	≥ 0.09
A_{30-40}	0.44	0.43	0.41	0.40	≥ 0.03
M_b	41040.84	43087.16	30476.42	33568.95	$[-75076, 75076]$
Trim	-2.02	-1.56	-1.19	-0.53	$[-2; 0]$
D_f	6.05	6.28	7.08	7.41	-
D_a	8.07	7.85	8.27	7.94	-
Heel	0.07	0.00	0.00	0.00	-

Як можна побачити з таблиці, кожна наступна оптимізація покращила значення диференту, при цьому зміни інших параметрів були незначними та не перечать припустимим значенням.

Після цього аналогічним чином виконується розташування наступних третьої та четвертої партій контейнерів, постійно беручи до уваги поточні значення параметрів морехідності.

Таким чином, завантаження кожної послідовної партії контейнерів за допомогою запропонованих у даній роботі алгоритмів забезпечує виконання вимог щодо остійності, місцевої та загальної міцності, диференту та сегрегації небезпечних вантажів, що підтверджує їх практичну застосовність.

Результат завантаження третьої партії контейнерів, що складається з 224 TEU загальною вагою 1434.1 т та 12 FEU загальною вагою 159.6 т та направлена у другий за черговістю порт за допомогою вищевказаного алгоритму, приведено на Рис. 6.8 та Рис. 6.9.

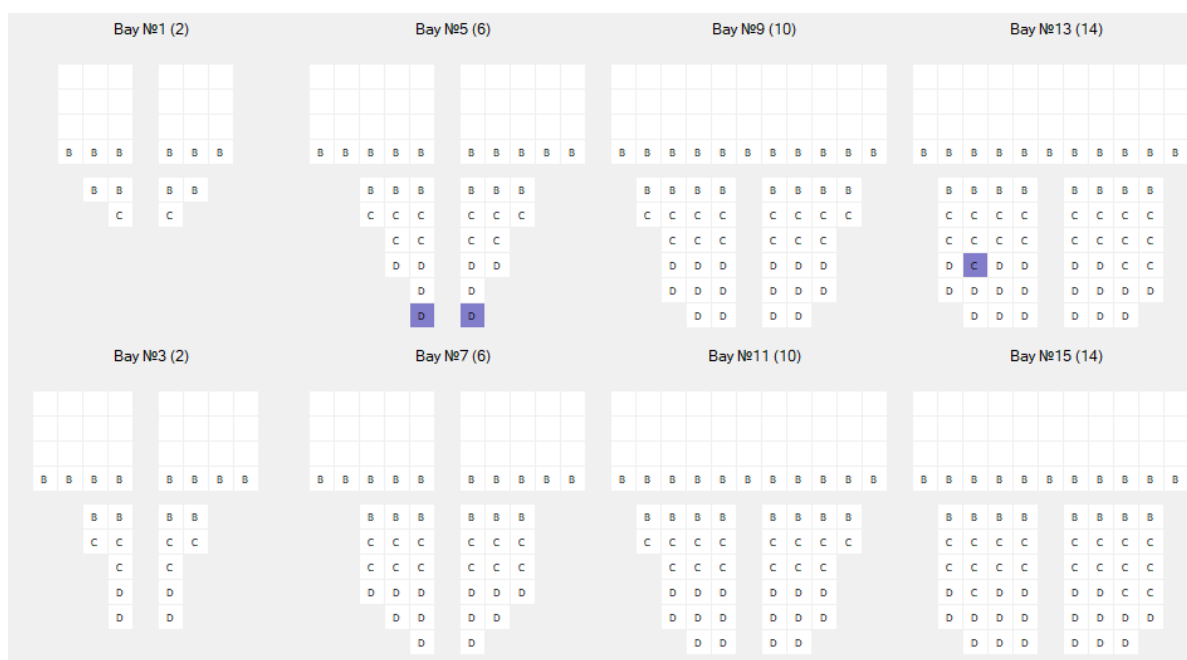


Рис. 6.8 Розташування третьої партії контейнерів, частина 1

Таке розташування продовжує тренд попередніх, тобто обмеження щодо розташування TEU поверх FEU виконуються, та більшість контейнерів завантажено у найнижчі яруси.

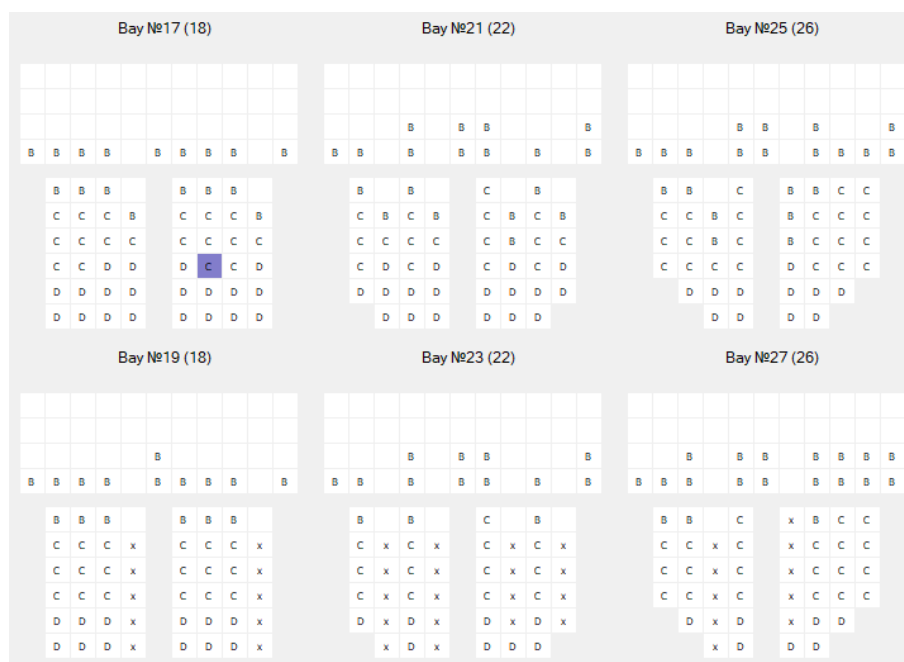


Рис. 6.9 Розташування третьої партії контейнерів, частина 2

Нарешті, результат завантаження четвертої партії контейнерів, що направлена у перший порт та складається з 118 TEU загальною вагою 504.6 т та 22 FEU загальною вагою 211.4 т, за допомогою запропонованого алгоритму початкового розміщення, приведено нижче (Рис. 6.10 та Рис. 6.11).

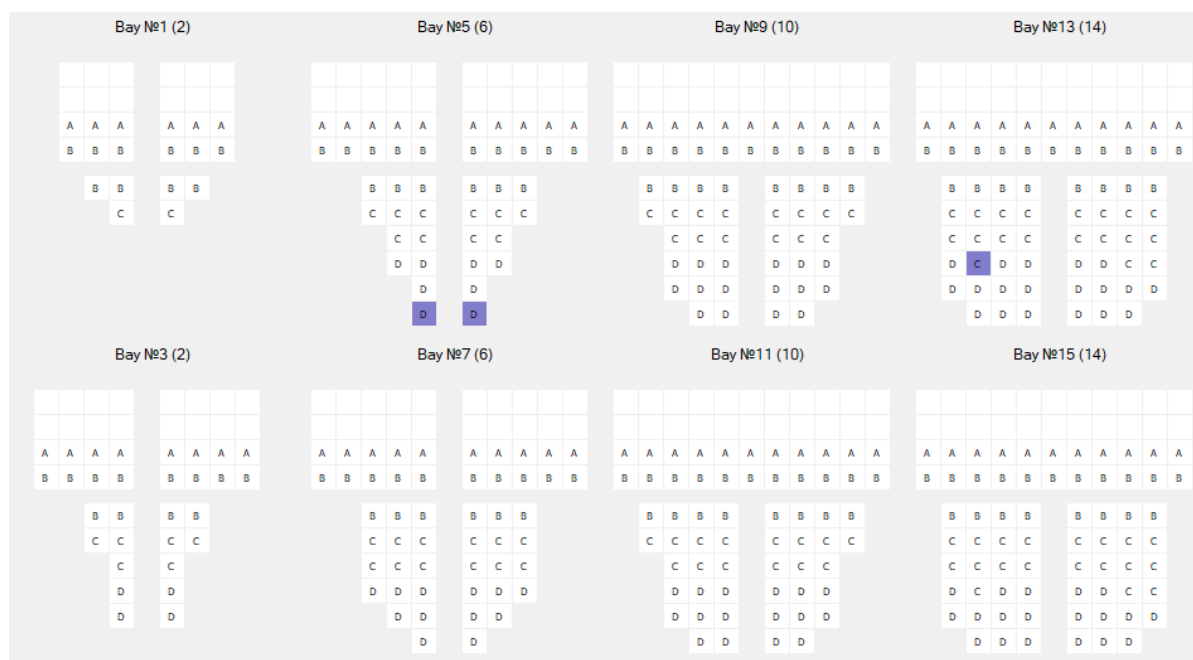


Рис. 6.10 Розташування четвертої партії контейнерів, частина 1

Варто зазначити, що приведена партія містить чотири FEU контейнера, які містять небезпечні вантажі класу 4.1. Беручи до уваги, що попередні партії містили контейнери з небезпечними вантажами класу 5.2, необхідно притримуватись правил сегрегації, приведених у підрозділі 3.1.

Для даного випадку контейнери з небезпечними вантажами різних класів не можуть знаходитись у одній вертикальній лінії, а відстань між ними повздовжно або поперек судна має становити, як мінімум, одне контейнерне місце. Ці вимоги було задоволено при попередньому розміщенні, що дозволяє уникнути застосування методу, приведенного у підрозділі 4.2.

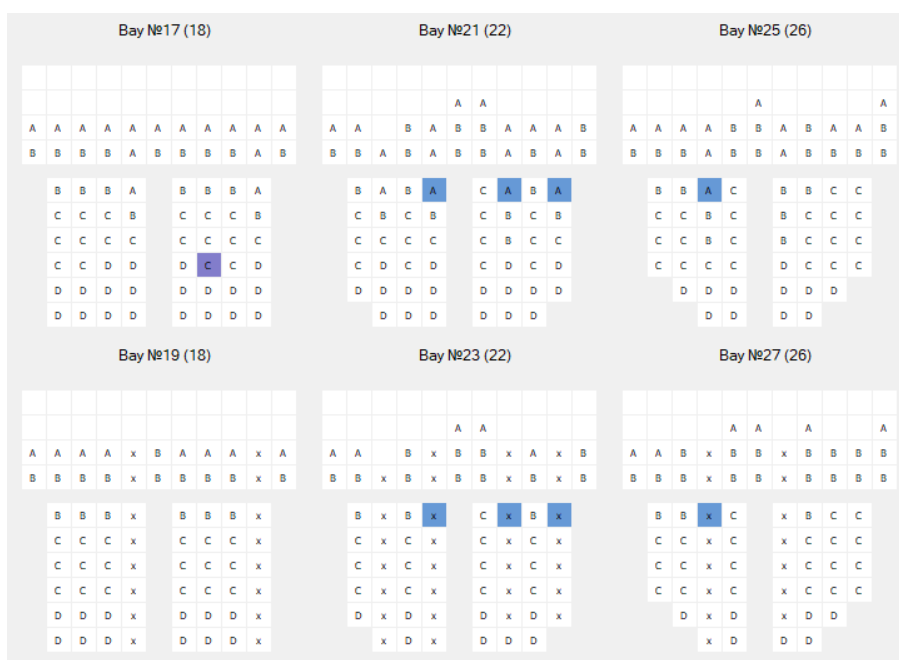


Рис. 6.11 Розташування четвертої партії контейнерів, частина 2

Параметри морехідності судна після завантажень партіями контейнерів #3 та #4 за допомогою методу початкового розташування та після оптимізації приведено у Таблиця 6.6

Таблиця 6.6

Параметри морехідності судна після завантажень #3 та #4

Параметр	#3	#3 опт.	#4	#4 опт.	Припустимі
D	23509.72	23509.72	24225.72	24225.72	≤ 36235
h	2.24	2.23	1.90	1.89	≥ 0.15
$l_{\max}$	2.23	2.22	1.97	1.96	≥ 0.2
$\theta_{\max}$	45.40	45.33	44.77	44.75	≥ 30
θ_v	>90	>90	>90	>90	≥ 60
A_{30}	0.38	0.38	0.34	0.33	≥ 0.055
A_{40}	0.72	0.72	0.63	0.63	≥ 0.09
A_{30-40}	0.34	0.34	0.30	0.30	≥ 0.03
M_b	26462.58	24980.23	19809.00	20119.35	$[-75076, 75076]$
Trim	-0.03	-0.01	0.10	0.00	$[-2; 0]$
D_f	8.12	8.13	8.39	8.34	-
D_a	8.15	8.14	8.29	8.34	-
Heel	0.01	0.00	0.00	0.00	-

Отримані таким чином варіанти завантаження після оптимізації дозволяють завантажити судно на рівний киль, виконуючи вимоги щодо параметрів морехідності. Таке завантаження може бути актуальним при обмеженнях за посадкою у порту або для максимізації кількості вантажу. Проте при необхідності отримати інший диферент, наприклад, для покращення ефективності руху, цього можна досягти за допомогою задання іншого оптимального моменту M_x для алгоритму оптимізації.

6.2 Окремий випадок перевірки моделі

Варіант завантаження, описаний у підрозділі 6.1 задовольняє усім вимогам математичної моделі (1) – (7). Тому для її перевірки було підібрано окремий випадок завантаження, який не задовольняє усім вимогам після застосування алгоритму початкового розташування, описаного у підрозділі 5.5.

Беручи за початкове завантаження, вказане на Рис. 6.10 та Рис. 6.11, на судно було додатково завантажено додаткову п'яту партію контейнерів, що направлені до першого порту вивантаження з наступними характеристиками (див. Таблиця 6.7) :

Таблиця 6.7

Інформація по контейнерах для додаткового завантаження

Порт	TEU, шт.	Вага, т	FEU, шт.	Вага, т
А	241	602.5	18	68.4

Завантаження цієї партії за допомогою вищезазначеного алгоритму викликало наступний варіант завантаження (Рис. 6.12 та Рис. 6.13).

Перевірка цього розташування шляхом перемноження матриці умов на вектор розташування виявила контейнери, що не задовольняють моделі (Рис. 6.14).

На рисунку червоним кольором позначено ті контейнери, що не задовольняють деяким рівнянням моделі (1) – (7). Більш детальний аналіз розташування показав, що було не задоволено рівняння (3) та (4). Це відбулося через те, що FEU контейнер, який

займає позиції (0, 2, 9) та (1, 2, 9) було розташовано у позицію, що була вже зайнята TEU контейнером – (1, 2, 9).

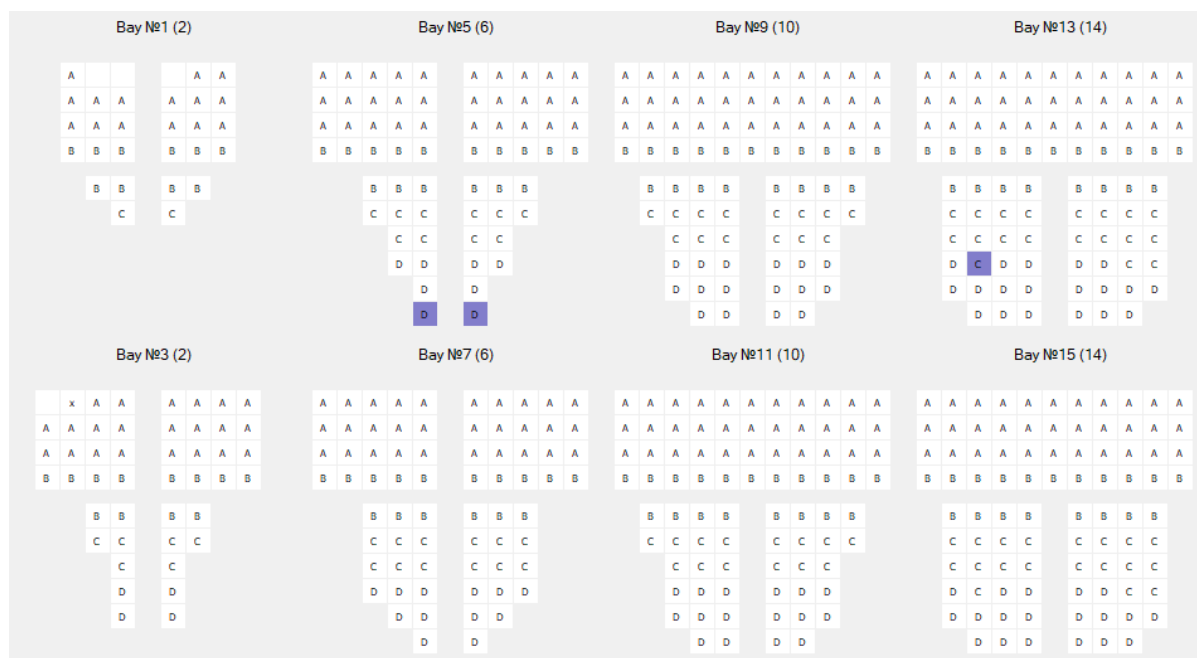


Рис. 6.12 Розташування п'ятої партії контейнерів, частина 1

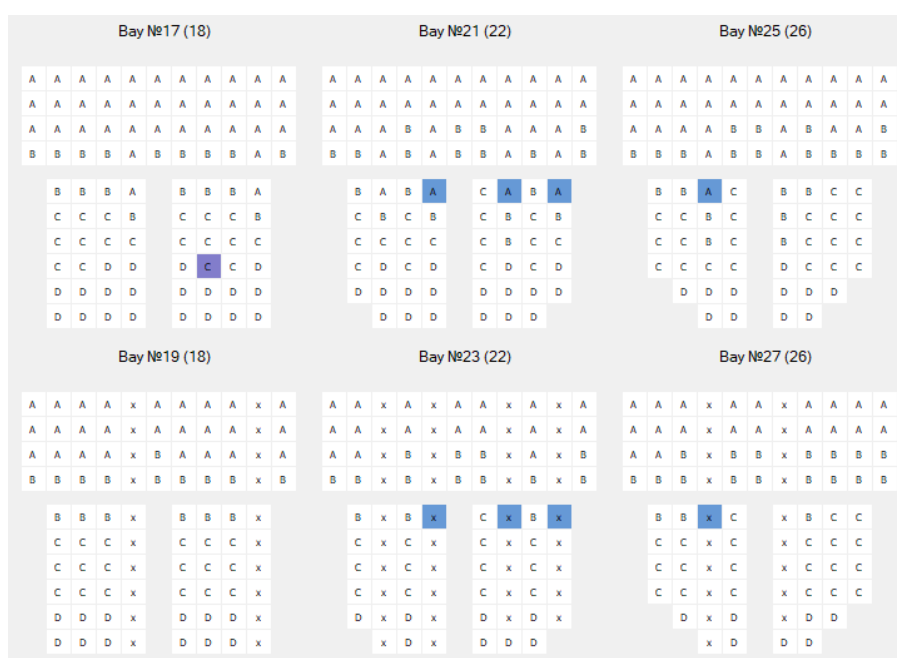


Рис. 6.13 Розташування п'ятої партії контейнерів, частина 2

Таким чином усі контейнери, що вказані у рівняннях (3) та (4) були помічені як незадовільні.

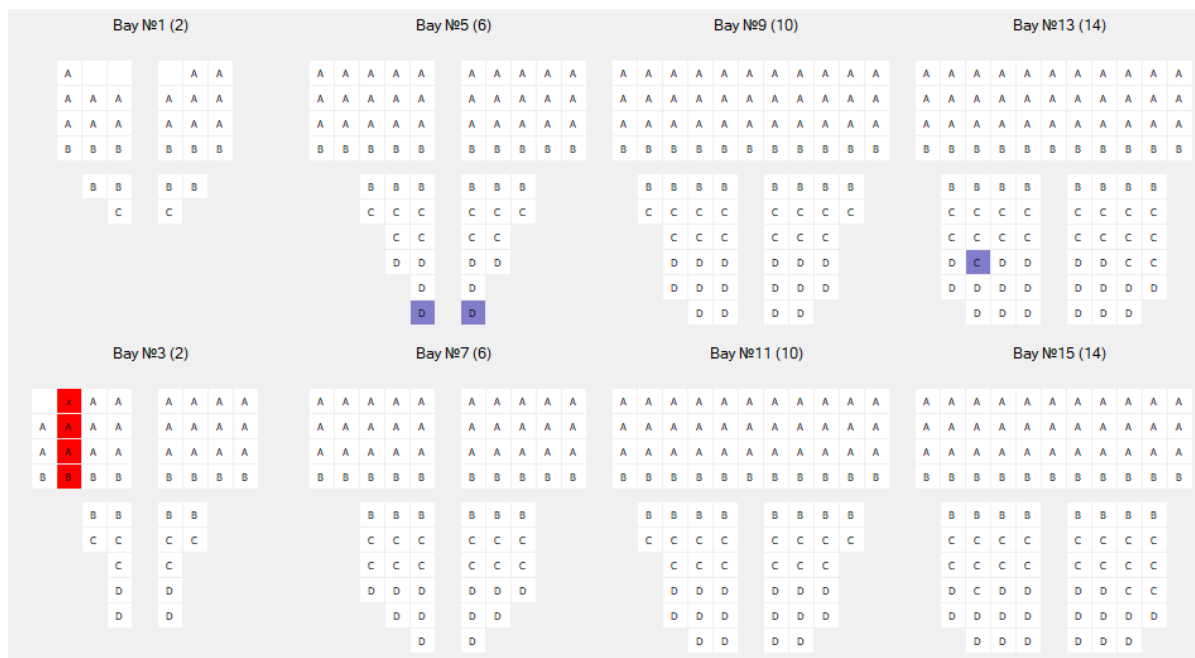


Рис. 6.14 Контейнери, що не задовольняють моделі

Для отримання такого розташування, яке б задовольняло моделі (1) – (7), було застосовано алгоритм, описаний у підрозділі 4.2.

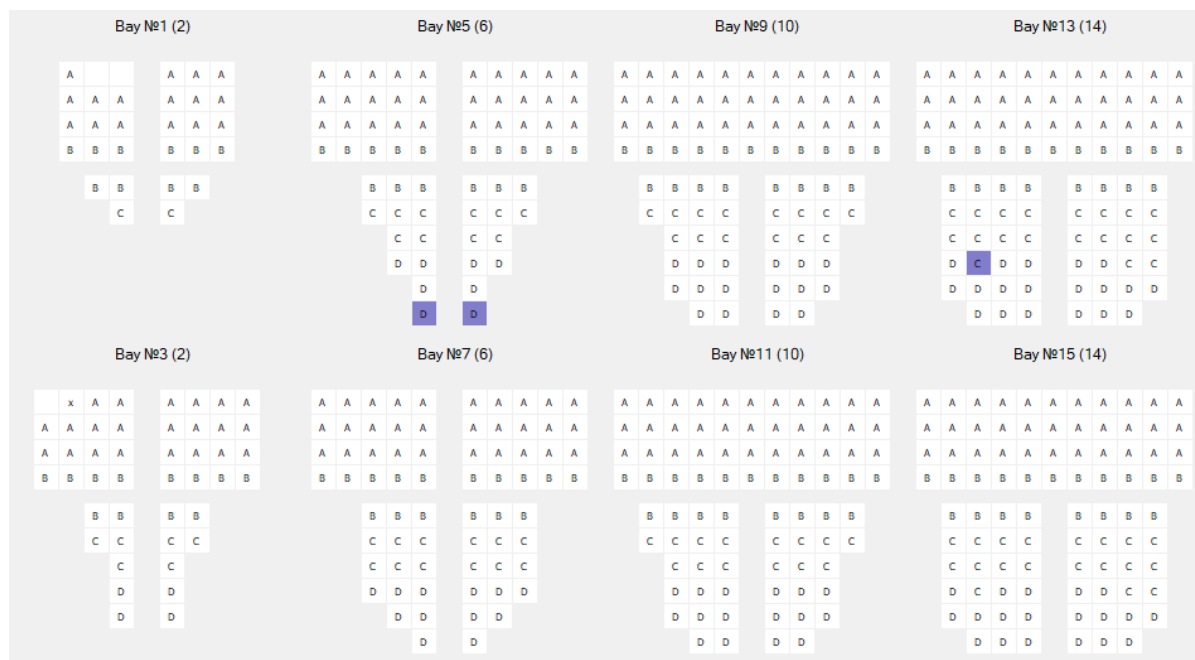


Рис. 6.15 Виправлене розташування п'ятої партії контейнерів

У результаті його роботи було отримано виправлене розташування контейнерів, приведене на Рис. 6.15. З рисунку видно, що для виконання вимог математичної моделі TEU контейнер, який займав позицію (1, 2, 9), було переміщено у позицію (0, 6, 9). Морехідні якості після розташування п'ятої партії контейнерів можна побачити на Рис. 6.16.

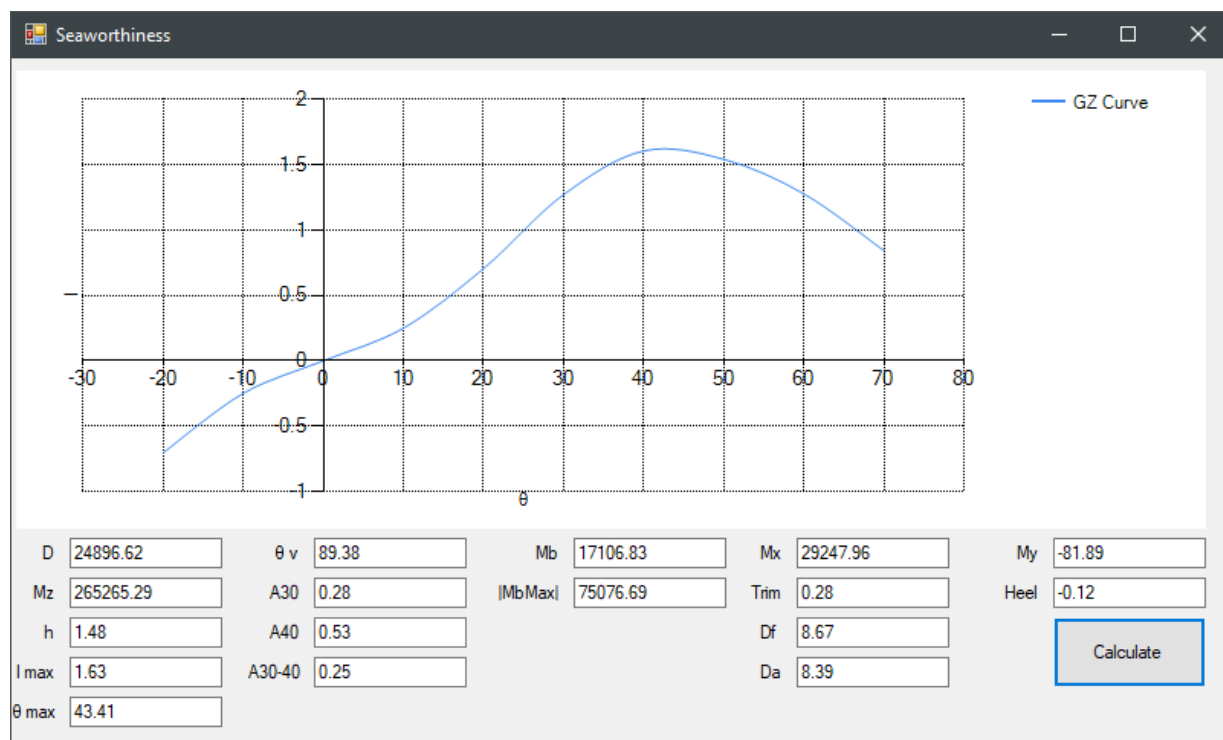


Рис. 6.16 Морехідні якості судна з п'ятою партією вантажів без оптимізації

Як можна побачити з рисунку, параметри остійності та загальної міцності у нормі, проте диферент на ніс є небажаним. Тому виконується повторна оптимізація щодо контейнерів, направлених у перший порт вивантаження, за допомогою алгоритму, приведеного у підрозділі 5.4.

Внаслідок проведеної оптимізації вдалось отримати значення диференту, що є припустимим (4 см на корму), при дотриманні припустимих значеннях остійності та загальної міцності (Рис. 6.17).

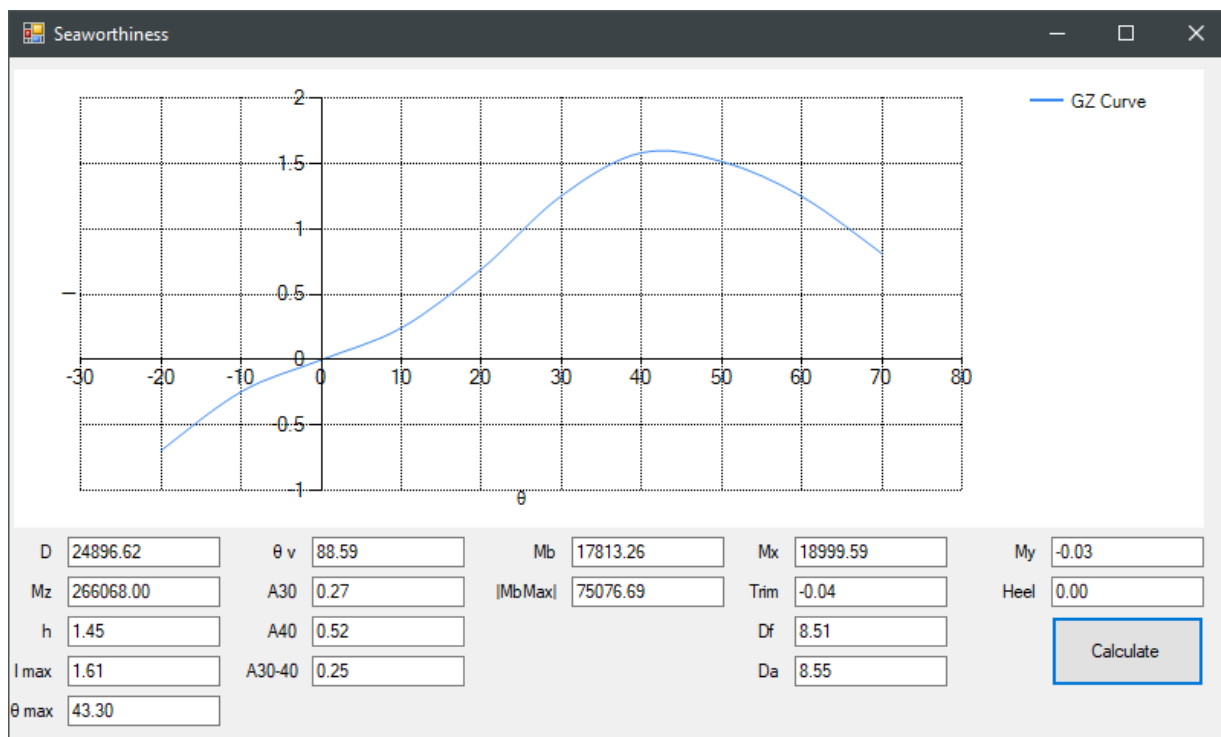


Рис. 6.17 Морехідні якості судна з п'ятою партією вантажів після оптимізації

6.3 Висновки за розділом 6

При виконанні розподілення стандартних партій контейнерів, які направлені у різні порти, по вантажним приміщенням судна, використання алгоритму початкового розташування, приведеного у підрозділі 5.5, дає припустимі результати щодо параметрів посадки, міцності та остійності.

Проте виконання оптимізації за допомогою алгоритму, приведеного у підрозділі 5.4, дає можливість покращити значення диференту та крену судна при незначній зміні параметрів остійності та міцності, які залишаються припустимими.

При виконанні складного завантаження, такого як повне завантаження судна контейнерами, можуть виникнути випадки, коли алгоритм початкового розташування не буде достатнім для забезпечення виконання усіх вимог щодо розміщення контейнерів, які охоплюються математичною моделлю (1) – (7). У таких випадках

використання алгоритму, приведеного у підрозділі 4.2, дозволяє виконати усі вимоги розробленої математичної моделі без урахування вимог щодо морехідних якостей судна.

Після отримання розв'язку, який задовольняє математичній моделі (1) – (7), доцільно провести подальшу оптимізацію за допомогою алгоритму, приведеного у підрозділі 5.4.

Виконана таким чином оптимізація може дозволити формування вантажного плану контейнерного судна, який буде задовольняти як структурним, так і операційним обмеженням.

ВИСНОВКИ

Дисертаційне дослідження спрямоване на оптимізацію завантаження контейнерних суден при забезпеченні обмежень щодо їх безпеки на етапі підготовки попереднього вантажного плану.

Серед джерел, присвячених питанню безпеки судноплавства, формування, обробка і представлення даних для вантажних програм, а також автоматизація завантаження суден представляються найбільш актуальними у зв'язку з сучасним розвитком обчислювальної техніки.

Роботи, що відносяться до розв'язання проблеми завантаження контейнеровозів, або основної проблеми бей-плану, можна поділити на дві категорії: ті, що розглядають проблему ціликом та ті, що підрозділяють її на декілька підпроблем. Проте через складність задачі усі підходи мають ряд припущень або спрощень, які роблять неможливим їх використання для розв'язання практичних задач. Через це розробка моделей та методів, які усувають такі спрощення, є актуальним та перспективним науковим напрямом.

Головний науковий результат даного дослідження полягає у розробці гнучкої математичної моделі, що дає можливість враховувати структурні та операційні обмеження як контейнерів, так і самого судна.

До основних наукових та практичних результатів приведеної роботи відносяться:

1. Покращена булева математична модель, що охоплює структурні та експлуатаційні обмеження, яка була розроблена для розв'язання поставленої задачі, та метод приведення обмежень цієї моделі до числового вигляду, що дозволяє використовувати методи теорії оптимізації для її розв'язання. Приведена модель враховує сегрегацію небезпечних вантажів і була

перевірена за допомогою точного методу, заснованого на загальному методі гілок та границь.

2. Метод розв'язання розглядуваної задачі виконання обмежень, який враховує загальну кількість невірно розташованих контейнерів та специфіку їх розташування на борту судна.
3. Метод оптимізації розміщення контейнерів щодо параметрів мореплавства, який використовує вагу, порт призначення, тип і клас небезпечного вантажу кожного окремого контейнера, і відрізняється можливістю використовувати гнучку цільову функцію, яка може модифікуватися щодо окремих параметрів мореплавства у залежності від особливостей конкретного рейсу.
4. Метод початкового розташування контейнерів за бемами, який враховує як обмеження математичної моделі, так і морехідні якості судна. Цей метод також може бути налаштований відносно окремих морехідних якостей.

Отримані у дисертації результати можуть бути використані при завантаженні контейнерних суден, розробці програмного забезпечення (вантажних програм) для оптимізації завантаження контейнерних суден, а також для підвищення кваліфікації спеціалістів та у навчальному процесі.

До якісних показників приведеного дослідження можна віднести можливість підвищення безпеки контейнерних суден при перевезенні вантажів. До кількісних показників можна віднести мінімізацію невірно розміщених контейнерів при автоматичному розподілі та оптимізацію окремих параметрів морехідності.

Результати імітаційного моделювання дослідження, які було отримано у даній роботі, обґрунтовують та доводять достовірність результатів приведеної дисертаційної роботи.

Розроблені модель та методи для розв'язання задачі оптимізації підготовки попереднього вантажного плану є прикладом застосування сучасних технологій при експлуатації контейнерних суден.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

- [1] Кабінет міністрів України, *Про схвалення Національної транспортної стратегії України на період до 2030 року*. Україна, 2018.
- [2] “Review of Maritime Transport 2020,” 2020, 2020th ed., [Online]. Available: https://unctad.org/system/files/official-document/rmt2020_en.pdf.
- [3] K. Kamieniev, A. Kamienieva, and M. Tsymbal, “Construction of a mathematical model and a method for arranging hazardous cargoes on a containership,” *Eastern-European J. Enterp. Technol.*, vol. 6, no. 3–102, pp. 20–27, 2019, doi: 10.15587/1729-4061.2019.183385.
- [4] А. В. Каменєва and К. І. Каменєв, “Розробка вантажної програми для контейнерного судна з урахуванням вимог Міжнародного морського кодексу з небезпечних вантажів,” *Наукові праці: науково-методичний журнал «Комп’ютерні технології»*, vol. 308, no. 296, pp. 94–99, 2017.
- [5] А. В. Каменєва and К. І. Каменєв, “Використання адитивного алгоритму для розміщення небезпечних вантажів на контейнерному судні,” *Судовождение: сборник научных трудов*, no. 28, pp. 70–77, 2018.
- [6] В. Г. Сизов, *Теория корабля*, Феникс. Одесса, 2003.
- [7] Л. Р. Аксютин, *Грузовой план судна*. Москва, 1976.
- [8] Ф. М. Кацман, Д. В. Дорогостайский, А. В. Конов, and Б. П. Коваленко, *Теория и устройство судов*, Судостроен. Ленинград, 1991.
- [9] Ю. Л. Воробьев, *Гидродинамика судна в стесненном фарватере*, Судостроен. Санкт-Петербург, 1992.
- [10] С. И. Войткунский, *Справочник по теории корабля. Т.2.*, Судостроен. Ленинград, 1985.
- [11] В. Н. Мельник, *Эксплуатационные расчеты мореходных характеристик судна*, Транспорт. Москва, 1990.

- [12] Н. И. Восковщук and В. В. Князев, “Определение остойчивости судна при больших наклонениях в начальной стадии проектирования,” *Тр. Дальневост. гос. техн. ун-та*, no. №113, pp. 28–32, 1994.
- [13] Ю. И. Нечаев and Э. Ю. Васильева, “Способ контроля остойчивости судна.” 1997.
- [14] M. Simonovich, V. G. Sizov, and Y. L. Vorobjov, “The correlation of ship hull form and her static stability diagram,” in *21 Jugosloven. kongr. teor. i primenjene meh.*, 1995, pp. 167–173.
- [15] Ю. П. Кучер and В. К. Рябченко, “Об альтернативных требованиях к остойчивости,” *Судовождение*, vol. 2, pp. 133–137, 2000.
- [16] Т. Ну, “Вероятностный расчет остойчивости поврежденного судна и его влияние на разделение корпуса судна на отсеки,” *Shanghai jiaotong daxue xuebao. Shanghai Jiaotong Univ.*, vol. 31, no. 11, pp. 24–29, 1997.
- [17] K. Ren and G. Chuanbo, “Влияние свободной циркуляции воды в аварийном отсеке на остойчивость судна,” *Sh. Eng.*, no. 4, pp. 4–7, 1992.
- [18] О. Э. Суров, *Влияние проектных характеристик судна на его мореходные и прочностные качества. Автореф. канд. техн. наук*, Дальневост. Владивосток, 2000.
- [19] Г. Б. Крыжевич, “О запасах местной прочности корпусов судов новых типов,” *Судостроение*, no. 8–9, pp. 3–6, 1992.
- [20] А. С. Брикер, С. Ю. Неклюдов, and С. А. Неклюдова, “Оценка местной прочности корпуса судна,” *Судостроение и судоремонт Сб. научн. тр. С-Петербург*, pp. 61–66, 1999.
- [21] Ю. В. Головешкин, *Третья проблема строительной механики корабля (нормирование прочности)*, Судостроен. Санкт-Петербург, 1999.
- [22] “Новые принципы нормирования прочности судов,” *Судостроение*, no. 4, pp. 9–10, 2003.

- [23] Г. В. Бойцов, “Метод расчета напряжений и деформаций при скручивании широко раскрытых судов,” *Тр. ЦНИИ им. ак. А.Н. Крылова*, no. 3, pp. 24–27, 1996.
- [24] Л. И. Коростылев, “О долговечности конструктивных узлов судового корпуса при циклических нагрузках,” *Строит. мех. корабля, Николаевс. кораблест. ин-т*, pp. 24–31, 1991.
- [25] А. В. Александров, А. Ю. Бабурин, А. А. Булгаков, and А. Е. Шишенин, “Программный комплекс для расчета общего напряженного состояния корпусов скоростных судов,” *Тр. ЦНИИ им. акад. А.Н. Крылова*, no. 3, pp. 44–48, 1996.
- [26] А. В. Иванов, “Влияние поперечной нагрузки и начального прогиба продольных ребер на предельный изгибающий момент корпуса судна,” *Сб. науч. тр. Ин-т инж. водн. трансп.*, no. 261, pp. 50–67, 1991.
- [27] С. Н. Гирич, “Прочность ребер жесткости при ударе грейфера,” *Сб. науч. тр. ин-т инж. вод. трансп.*, no. 261, pp. 14–27, 1991.
- [28] V. A. Kulesh, “Computer investigation of construction reliability,” in *Proc. 6th Int. Offshore and Polar Eng. Conf., Los Angeles, Calif.*, 1996, pp. 395–401.
- [29] L. Miller, “Advanced calculation techniques for ship structural design,” *Germ. Marit. Ind. J.*, no. 8, pp. 37–40, 1992.
- [30] X. Xiao and J. W. F. Guo, “Конструкционные расчеты и система программирования расчетов крупнотоннажных танкеров,” *Shanghai jiaotong daxue xuebao. Shanghai Jiaotong Univ.*, vol. 31, no. 11, pp. 18–23, 1997.
- [31] Z. Wan and F. He, “Estimation of ultimate strength of ship’s hull girders,” *Sh. Mech.*, vol. 7, no. 3, pp. 58–67, 2003.
- [32] Г. В. Егоров, “Развитие требований к средствам контроля загрузки морских и смешанного плавания судов,” *Автоматизация судовых технических средств*, no. 5, pp. 36–53, 2000.
- [33] Г. В. Егоров, “Изгиб сильно поврежденного корпуса судна,” *Судовождение*, no.

- 2, pp. 50–62, 2000.
- [34] В. Н. Бугаев, “Системный способ расчета непотопляемости судна,” in *Тез. докл. науч.-техн. конф. «Пробл. обеспеч. живучести кораблей и судов» НТО судостроителей им. акад. А. Н. Крылова*, 1992, pp. 35–36.
- [35] В. В. Васюнькин and М. Ю. Шумаев, “Динамическая модель состояния корабля при борьбе за непотопляемость,” in *Тез. докл. науч.-техн. «Проблемы обеспечения живучести кораблей и судов», НТО судостроителей им. акад. А. Н. Крылова*, 1992, pp. 38–40.
- [36] B.-C. Chang, “On the damage survivability of Ro-Ro ships investigated by motion simulation in a seaway,” *Schiffstechnik*, vol. 46, no. 4, pp. 192–207, 1999.
- [37] Е. А. Власенко, “Определение ускорения, возникающего при бортовой качке судна,” in *XXX Международная конференция «Развитие науки в XXI веке»*, 2017, pp. 41–46.
- [38] Е. А. Власенко, “Угловые ускорения, возникающие при качке судна,” in *X Міжнародна наук.-практ. конф. “Сучасні інформаційні та інноваційні технології на транспорті” (MINTT-2018)*, 2018, pp. 96–98.
- [39] Е. А. Власенко, “Зависимость сил инерции бортовой качки от момента инерции судна относительно продольного оси,” *Austria Sci.*, no. 23, pp. 54–60, 2019.
- [40] А. И. Вакс and Б. Н. Куров, “Определение параметров статической посадки корабля на волнении,” *Судостроение*, no. 3, pp. 14–17, 1997.
- [41] С. Н. Гирин and А. М. Фролов, “Анализ влияния спектра волнения на величину волнового изгибающего момента в корпусе судна,” *Тр. Волж. гос. акад. водн. трансп.*, vol. 6, no. 283, pp. 39–43, 1999.
- [42] В. П. Коротков, Б.П. Модников and Э. Ю. Терехович, “Динамическая система контроля и обеспечения безопасности судов в штормовом море,” *Судостроение*, no. 7, pp. 4–10, 1996.
- [43] Y. Huang, H. Xu, and Y. Hu, “Критическое, грозящее опрокидыванием,

- положение судна в условиях параметрического резонанса при попутном волнении,” *Chuanbo lixie*, vol. 3, no. 5, pp. 27–33, 1999.
- [44] Е. А. Власенко, “Безопасная перевозка груза с учетом качки судна,” *Автоматизация судовых технических средств*, no. 24, pp. 19–26, 2018.
- [45] Е. А. Власенко, “Влияние загрузки судна на величину сил инерции, возникающих при качке,” in *Транспортні технології (морський та річковий флот): інфраструктура, судноплавство, перевезення автоматизація*, 2018, pp. 110–111.
- [46] Е. А. Власенко and И. А. Бурмака, “Определение максимального значения горизонтальной составляющей сил инерций, действующей на груз при качке судна,” *Sci. Educ. a New Dimens. Nat. Tech. Sci.*, vol. VI (18), no. 158, pp. 54–80, 2018.
- [47] И. А. Власенко, Е. А. Бурмака, “Загрузка судна с учетом сил инерции, возникающих при переходе,” in *Транспортні технології (морський та річковий флот): інфраструктура, судноплавство, перевезення автоматизація*, 2017, pp. 142–144.
- [48] Ю. Д. Жуков, “Пути снижения аварийности малых судов,” *Проектирование судов и судов. устройств*, pp. 77–85, 1991.
- [49] А. Я. Миврезов, “Анализ повреждений судов при контактах,” *Проек-тир. судов и судов. устройств. Николаевс. кораблестроит. ин-т*, pp. 40–46, 1991.
- [50] В. А. Лукин, “Статистический анализ данных о повреждении судовых корпусных конструкций,” *Производство судов и судов. устройств, Николаевс. кораблестроит. ин-т*, pp. 23–29, 1991.
- [51] Г. В. Егоров, “Идентификация опасностей и меры по снижению риска при смене балласта существующими судами в море,” *Судовождение*, no. 3, pp. 64–75, 2001.
- [52] В. И. Бекерский and Б. П. Богданов, “Оперативный расчет крепления грузов на морских судах,” *Судовождение*, no. 4, pp. 26–31, 2002.

- [53] В. К. Рябченко, “Грузовая шкала для определения водоизмещения и осадок при значительном дифференте,” *Судовождение*, no. 2, pp. 175–177, 2000.
- [54] Е. Г. Егоров, “Учет гибкости корпуса при расчетах посадки и элементов изгиба судна,” *Автоматизация судовых технических средств*, no. 9, pp. 37–45, 2004.
- [55] Ю. П. Кучер and В. К. Рябченко, “К уточнению расчета водоизмещения и количества груза по осадкам,” *Судовождение*, no. 7, pp. 41–46, 2004.
- [56] А. А. Родионов, “Исследование математических моделей оптимизации судовых конструкций,” *Судостроение*, no. 8–9, pp. 6–10, 1992.
- [57] Ю. Л. Пономаренко, “Оптимизация загрузки транспортных средств, осуществляющих коммерческие перевозки,” in *4-а Міжнародна науково-практична конференція «Ринок послуг комплексних транспортних систем та прикладні проблеми логістики» проблеми логістики*, 2002, pp. 265–269.
- [58] В. В. Козляков and Г. В. Егоров, “Автоматизированный расчет грузового плана и непотопляемости судна на бортовой ПЭВМ,” *Проектир. судов и судов. устройств. Николаев. кораблестроит. ин-т*, pp. 55–63, 1992.
- [59] Н. Н. Цымбал and Ю. Ю. Васьков, “Расчет границ множества допустимых стратегий проведения грузовых операций навалочных судов,” *Судовождение*, no. 8, pp. 22–31, 2004.
- [60] Н. Н. Цымбал and Ю. Ю. Васьков, “Формирование оптимизационной задачи проведения грузовых операций навалочных судов,” *Судовождение*, no. 7, pp. 3–9, 2004.
- [61] Н. Н. Цымбал and Ю. Ю. Васьков, “Выбор оптимального варианта проведения грузовых операций навалочных судов,” *Автоматизация судовых технических средств*, no. 9, pp. 103–107, 2004.
- [62] Ю. Ю. Васьков, “Некоторые вопросы оптимизации грузовых операций навалочных судов,” *Судовождение*, no. 6, p. 40 45, 2003.
- [63] В. С. Заичко, “Специфика учета ограничений по грузоместимости трюмов при

- загрузке судов генеральными грузами,” *Судовождение*, no. 10, pp. 53–57, 2005.
- [64] В. С. Заичко, “Особенности формирования загрузки судов генеральными грузами,” *Судовождение*, no. 9, pp. 25–28, 2005.
- [65] В. С. Заичко, “Алгоритм формирования множества допустимых загрузок судна генеральными грузами,” *Судовождение*, no. 12, pp. 59–62, 2006.
- [66] В. С. Заичко, “Моделирование укладки генеральных грузов в трюма судна,” *Судовождение*, no. 11, pp. 51–55, 2006.
- [67] В. С. Заичко, “Отношение предпочтения партий генерального груза по доходности их перевозок,” *Судовождение*, no. 14, pp. 35–38, 2007.
- [68] А. О. Чепок and Н. Н. Цымбал, “Выбор стратегии загрузки судна генеральными грузами с учетом их укладки,” *Судовождение*, no. 16, pp. 137–141, 2009.
- [69] А. О. Чепок, “Выбор типов генеральных грузов при моделировании их укладки в грузовые помещения судна,” *Судовождение*, no. 17, pp. 233–238, 2009.
- [70] Л. Л. Николаева and А. О. Чепок, “Моделирование укладки генеральных грузов в грузовые помещения судна,” *Судовождение*, no. 18, pp. 143–150, 2010.
- [71] В. Д. Савчук and Е. Н. Клименко, “Обеспечение безопасной перевозки цемента насыпью на балкерах,” *Судовождение*, no. 7, pp. 72–75, 2004.
- [72] Л. Л. Николаева, А. В. Гайченя, and М. Ю. Соколов, “Разработка метода оперативной оценки критериев мореходности судна,” *Судовождение*, no. 16, pp. 132–136, 2009.
- [73] М. Ю. Соколов, “Метод формирования грузовой программы судна с использованием ранее созданной базы данных,” *Судовождение*, no. 18, pp. 169–172, 2010.
- [74] А. В. Гайченя, “Ввод в базу данных характеристик судна,” *Судовождение*, no. 18, pp. 64–67, 2010.
- [75] А. В. Гайченя, “Ввод информации о танках в базу данных судна,” *Судовождение*, no. 19, pp. 61–65, 2011.

- [76] А. В. Гайченя, “Программное описание стандартных грузовых помещений,” *Судовождение*, no. 17, pp. 37–40, 2009.
- [77] Л. Л. Николаева and А. В. Гайченя, “Способ формализации формы судовых грузовых помещений нестандартного типа,” *Автоматизация судовых технических средств*, no. 16, pp. 71–82, 2010.
- [78] М. Ю. Соколов, “Общий вид зависимостей параметров посадки, остойчивости и прочности судна от исходных данных,” *Судовождение*, no. 17, pp. 199–204, 2009.
- [79] А. О. Чепок, “Отображение параметров посадки, остойчивости и общей продольной прочности в грузовой компьютерной программе судна,” *Судовождение*, no. 19, pp. 163–169, 2011.
- [80] А. О. Чепок, “Разработка процедуры отображения укладки генерального груза в трюмах судна,” *Судовождение*, no. 20, pp. 146–149, 2011.
- [81] Е. А. Власенко, Е. В. Калиниченко, and М. Н. Цымбал, “Имитационное моделирование загрузки контейнеровоза,” *Austria - Sci.*, no. 26, pp. 43–49, 2019.
- [82] Е. А. Власенко and Н. Н. Цымбал, “Некоторые особенности составления грузового плана контейнеровозов,” *Судовождение*, no. 28, pp. 35–41, 2018.
- [83] Е. А. Власенко, “Допустимая загрузка контейнеровоза,” *Sci. Educ. a New Dimens. Nat. Tech. Sci.*, vol. VI (22), no. 186, pp. 87–94, 2018.
- [84] D. Ambrosino, A. Sciomachen, and E. Tanfani, “Stowing a containership: The master bay plan problem,” *Transp. Res. Part A Policy Pract.*, vol. 38, no. 2, pp. 81–99, 2004, doi: 10.1016/j.tra.2003.09.002.
- [85] A. Sciomachen and E. Tanfani, “A 3D-BPP approach for optimising stowage plans and terminal productivity,” *Eur. J. Oper. Res.*, vol. 183, no. 3, pp. 1433–1446, 2007, doi: 10.1016/j.ejor.2005.11.067.
- [86] D. Ambrosino, D. Anghinolfi, M. Paolucci, and A. Sciomachen, “A new three-step heuristic for the Master Bay Plan Problem,” *Marit. Econ. Logist.*, vol. 11, no. 1, pp. 98–120, 2009, doi: 10.1057/mel.2008.19.

- [87] D. Ambrosino, D. Anghinolfi, M. Paolucci, and A. Sciomachen, “An experimental comparison of different heuristics for the master bay plan problem,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 6049 LNCS, pp. 314–325, 2010, doi: 10.1007/978-3-642-13193-6_27.
- [88] D. Ambrosino and A. Sciomachen, “Using a Bin Packing Approach for Stowing Hazardous Containers into Containerships,” in *Springer Optimization and Its Applications*, 2015, pp. 1–18.
- [89] D. Ambrosino, M. Paolucci, and A. Sciomachen, “Experimental evaluation of mixed integer programming models for the multi-port master bay plan problem,” *Flex. Serv. Manuf. J.*, vol. 27, no. 2–3, pp. 263–284, 2015, doi: 10.1007/s10696-013-9185-4.
- [90] O. Dubrovsky, G. Levitin, and M. Penn, “A genetic algorithm with a compact solution encoding for the container ship stowage problem,” *J. Heuristics*, vol. 8, no. 6, pp. 585–599, 2002, doi: 10.1023/A:1020373709350.
- [91] I. D. Wilson, P. A. Roach, and J. A. Ware, “Container stowage pre-planning: Using search to generate solutions, a case study,” *Knowledge-Based Syst.*, vol. 14, no. 3–4, pp. 137–145, 2001, doi: 10.1016/S0950-7051(01)00090-9.
- [92] M. Zeng, M. Y. H. Low, W. J. Hsu, S. Y. Huang, F. Liu, and C. A. Win, “Automated stowage planning for large containerships with improved safety and stability,” in *Proceedings of the 2010 Winter Simulation Conference*, Dec. 2010, pp. 1976–1989, doi: 10.1109/WSC.2010.5678873.
- [93] H. Lei and M. Ok, “Dangerous goods container allocation in ship stowage planning,” *ICORES 2020 - Proc. 9th Int. Conf. Oper. Res. Enterp. Syst.*, pp. 241–246, 2020, doi: 10.5220/0009160602410246.
- [94] H. Zhu, M. Ji, and W. Guo, “Integer Linear Programming Models for the Containership Stowage Problem,” *Math. Probl. Eng.*, vol. 2020, 2020, doi: 10.1155/2020/4382745.
- [95] D. M. Rahsed, M. S. Gheith, and A. B. Eltawil, “A Rule-based Greedy Algorithm to Solve Stowage Planning Problem,” *IEEE Int. Conf. Ind. Eng. Eng. Manag.*, vol. 2019-

- Decem, no. January 2021, pp. 437–441, 2019, doi: 10.1109/IEEM.2018.8607517.
- [96] Q. Zhang, Z. Liu, and J. Yan, “Research on Decision-making Optimization of Container MBPP Based on Evolutionary Strategy Algorithm,” *Proc. 31st Chinese Control Decis. Conf. CCDC 2019*, pp. 3776–3782, 2019, doi: 10.1109/CCDC.2019.8832471.
- [97] C. Parreño-Torres, R. Alvarez-Valdes, and F. Parreño, “Solution strategies for a multiport container ship stowage problem,” *Math. Probl. Eng.*, vol. 2019, 2019, doi: 10.1155/2019/9029267.
- [98] M. A. Yurtseven, E. Boulougouris, and O. Turan, “Container ship stowage plan using steepest ascent hill climbing, genetic, and simulated annealing algorithms,” *Mar. Des. XIII*, vol. 1, pp. 617–623, 2018.
- [99] Matsaini and B. Santosa, “Solving the Container Stowage Problem (CSP) using Particle Swarm Optimization (PSO),” *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 337, no. 1, 2018, doi: 10.1088/1757-899X/337/1/012002.
- [100] M. W. Cohen, V. N. Coelho, A. Dahan, and I. Kaspi, “Container vessel stowage planning system using genetic algorithm,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 10199 LNCS, pp. 557–572, 2017, doi: 10.1007/978-3-319-55849-3_36.
- [101] M. Hu and W. Cai, “Multi-objective optimization based on improved genetic algorithm for containership stowage on full route,” *2017 4th Int. Conf. Ind. Eng. Appl. ICIEA 2017*, pp. 224–228, 2017, doi: 10.1109/IEA.2017.7939211.
- [102] Y. Shen, N. Zhao, M. Xia, and X. Du, “A deep Q-learning network for ship stowage planning problem,” *Polish Marit. Res.*, vol. 24, no. S3, pp. 102–109, 2017, doi: 10.1515/pomr-2017-0111.
- [103] F. Parreño, D. Pacino, and R. Alvarez-Valdes, “A GRASP algorithm for the container stowage slot planning problem,” *Transp. Res. Part E Logist. Transp. Rev.*, vol. 94, pp. 141–157, 2016, doi: 10.1016/j.tre.2016.07.011.

- [104] Z. Q. Lee, R. Fan, and W. J. Hsu, "Optimizing constraint test ordering for efficient automated stowage planning," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 9335, pp. 343–357, 2015, doi: 10.1007/978-3-319-24264-4_24.
- [105] A. Delgado, R. M. Jensen, K. Janstrup, T. H. Rose, and K. H. Andersen, "A Constraint Programming model for fast optimal stowage of container vessel bays," *Eur. J. Oper. Res.*, vol. 220, no. 1, pp. 251–261, 2012, doi: 10.1016/j.ejor.2012.01.028.
- [106] F. Liu, M. Y. H. Low, W. J. Hsu, S. Y. Huang, M. Zeng, and C. A. Win, "Randomized algorithm with tabu search for multi-objective optimization of large containership stowage plans," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 6971 LNCS, pp. 256–272, 2011, doi: 10.1007/978-3-642-24264-9_20.
- [107] Верховна Рада України, Закон України «Про перевезення небезпечних вантажів». Україна, 2020.
- [108] IMO, *International Maritime Dangerous Goods Code - Imdg Code*. IMO, 2020.
- [109] Каменев К.И. and Каменева А.В., "Automation of constraints creation for master bay plan problem," *Slovak Int. Sci. J.*, vol. 2, no. 45, pp. 8–11, 2020, [Online]. Available: <http://sis-journal.com/wp-content/uploads/2020/10/Slovak-international-scientific-journal-№45-2020-VOL.2.pdf>.
- [110] M. Fisun, H. Horban, and K. Ihor, "Processing of Relational Algebra Expressions by the Shunting Yard Algorithm," in *2019 IEEE 14th International Conference on Computer Sciences and Information Technologies (CSIT)*, Sep. 2019, pp. 240–243, doi: 10.1109/STC-CSIT.2019.8929740.
- [111] O. Danvy and K. Millikin, "Refunctionalization at work," *Sci. Comput. Program.*, vol. 74, no. 8, pp. 534–549, 2009, doi: 10.1016/j.scico.2007.10.007.
- [112] P. Rusnak, "Parser of Input Data in Reliability Analysis based on Logical Differential Calculus," *Cent. Eur. Res. J.*, vol. 2, no. 2, pp. 11–16, 2016, [Online]. Available:

<http://ceres-journal.eu/iss160202>.

- [113] Y. Wang, P. Siriaraaya, H. Sakata, Y. Kawai, and K. Tajima, “A map search system based on a spatial query language,” *Adv. Database Technol. - EDBT*, vol. 2019-March, pp. 578–581, 2019, doi: 10.5441/002/edbt.2019.60.
- [114] K. Kamieniev, “Input data preparation for the Master Bay Plan problem solution,” *Sci. Educ. a New Dimens.*, vol. VIII(238), no. 29, pp. 51–54, 2020, doi: 10.31174/send-nt2020-238viii29-12.
- [115] X. Taxa, *Введение в исследование операций: В 2-х книгах. Кн. I*. М.: Мир, 1985.
- [116] M. Avriel, M. Penn, and N. Shpirer, “Container ship stowage problem: Complexity and connection to the coloring of circle graphs,” *Discret. Appl. Math.*, vol. 103, no. 1–3, pp. 271–279, 2000, doi: 10.1016/S0166-218X(99)00245-0.
- [117] S. H. Zhan, Z. J. Zhang, L. J. Wang, and Y. W. Zhong, “List-based simulated annealing algorithm with hybrid greedy repair and optimization operator for 0-1 knapsack problem,” *IEEE Access*, vol. 6, pp. 54447–54458, 2018, doi: 10.1109/ACCESS.2018.2872533.
- [118] A. E. Ezugwu, V. Pillay, D. Hirasen, K. Sivanarain, and M. Govender, “A Comparative Study of Meta-Heuristic Optimization Algorithms for 0 - 1 Knapsack Problem: Some Initial Results,” *IEEE Access*, vol. 7, no. April, pp. 43979–44001, 2019, doi: 10.1109/ACCESS.2019.2908489.
- [119] A. Liu, J. Wang, G. Han, S. Wang, and J. Wen, “Improved simulated annealing algorithm solving for 0/1 knapsack problem,” *Proc. - ISDA 2006 Sixth Int. Conf. Intell. Syst. Des. Appl.*, vol. 2, pp. 1159–1164, 2006, doi: 10.1109/ISDA.2006.253776.
- [120] Y. H. Chou, Y. J. Yang, and C. H. Chiu, “Classical and quantum-inspired Tabu search for solving 0/1 knapsack problem,” *Conf. Proc. - IEEE Int. Conf. Syst. Man Cybern.*, no. 1, pp. 1364–1369, 2011, doi: 10.1109/ICSMC.2011.6083848.
- [121] J. Zhang, “Comparative study of several intelligent algorithms for knapsack problem,” *Procedia Environ. Sci.*, vol. 11, no. PART A, pp. 163–168, 2011, doi:

10.1016/j.proenv.2011.12.025.

- [122] J. Langeveld and A. P. Engelbrecht, “Set-based particle swarm optimization applied to the multidimensional knapsack problem,” *Swarm Intell.*, vol. 6, no. 4, pp. 297–342, 2012, doi: 10.1007/s11721-012-0073-4.
- [123] L. F. Mingo López, N. Gómez Blas, and A. Arteta Albert, “Multidimensional knapsack problem optimization using a binary particle swarm model with genetic operations,” *Soft Comput.*, vol. 22, no. 8, pp. 2567–2582, 2018, doi: 10.1007/s00500-017-2511-0.
- [124] H. Fingler, E. N. Cáceres, H. Mongelli, and S. W. Song, “A CUDA based solution to the Multidimensional Knapsack Problem using the ant colony optimization,” *Procedia Comput. Sci.*, vol. 29, no. 30, pp. 84–94, 2014, doi: 10.1016/j.procs.2014.05.008.
- [125] A. Alzaqebah and A. A. Abu-Shareha, “Ant Colony System Algorithm with Dynamic Pheromone Updating for 0/1 Knapsack Problem,” *Int. J. Intell. Syst. Appl.*, vol. 11, no. 2, pp. 9–17, 2019, doi: 10.5815/ijisa.2019.02.02.
- [126] M. Melanie, *An Introduction to Genetic Algorithms*, no. 5. 1999.
- [127] G. Syswerda, “Uniform crossover in genetic algorithms,” in *Proceedings of the 3rd International Conference on Genetic Algorithms*, 1989, pp. 2–9.
- [128] G. Syswerda, *A Study of Reproduction in Generational and Steady-State Genetic Algorithms*, vol. 1. Morgan Kaufmann Publishers, Inc., 1991.
- [129] K. A. De Jong and J. Sarma, “Generation Gaps Revisited,” in *Foundations of Genetic Algorithms*, vol. 2, L. D. WHITLEY, Ed. Elsevier, 1993, pp. 19–28.
- [130] D. Whitley, “The GENITOR algorithm and selection pressure: Why rank-based allocation of reproductive trials is best,” *Proc. Third Int. Conf. Genet. Algorithms*, vol. 1, pp. 116–121, 1989.
- [131] IMO, *International Code on Intact Stability*, 2008. International Maritime Organization, 2009.

ДОДАТКИ

Додаток А

Акт впровадження результатів дисертаційного дослідження

«ЗАТВЕРДЖУЮ»
 Перший проректор НУ «ОМА»
 д.ю.н., професор
 _____ Шемякін О. М.
 «17» травня 2021 р.



АКТ

про використання результатів дисертаційної роботи здобувача наукового ступеня Камєнєва Кирила Ігоровича на тему «Розробка методу розв'язання задачі складання вантажного плану контейнеровозу з урахуванням структурних та операційних обмежень» у навчальному процесі НУ «ОМА».

Ми, що нижче підписалися, директор навчально-наукового інституту навігації, д.т.н., професор Цимбал М. М. та зав. кафедрою «Морські перевезення» д.е.н., професор Ніколаєва Л. Л. склали цей акт в тому, що результати здобувача Камєнєва Кирила Ігоровича на тему «Розробка методу розв'язання задачі складання вантажного плану контейнеровозу з урахуванням структурних та операційних обмежень» впроваджені у освітній процес на кафедрі «Морські перевезення» у дисципліні «Експлуатація спеціалізованих суден».

Директор навчально-наукового інституту
 навігації, д.т.н., професор

Зав. Кафедрою «Морські перевезення»,
 д.е.н., професор




Цимбал М. М.

Ніколаєва Л. Л.

Додаток Б

Коди методів реалізації алгоритмів мовою програмування C#

```

public static int[,] algorithmSteadyStateStage1(int[,] startingSolution, int[] maxMutationArray, int
populationSize,
    int iterationsNumber, Func<int[,], int> unfitnessFunction)
{
    int solutionLength0D = startingSolution.GetLength(0);
    int solutionLength1D = startingSolution.GetLength(1);
    int[] unFitnesses = new int[populationSize];
    int[,] population = new int[populationSize][,];
    population[0] = new int[solutionLength0D, solutionLength1D];
    population[0] = startingSolution.Clone() as int[,];
    unFitnesses[0] = unfitnessFunction(population[0]);
    int bestUnfitness = unFitnesses[0];
    //creating an initial population
    for (int i = 1; i < populationSize; i++)
    {
        population[i] = new int[solutionLength0D, solutionLength1D];
        population[i] = startingSolution.Clone() as int[,];
        mutateStage1V3(population[i], new int[] { 0, 0, 0 }, maxMutationArray, 0.3);
        unFitnesses[i] = unfitnessFunction(population[i]);
        if (unFitnesses[i] < bestUnfitness) bestUnfitness = unFitnesses[i];
    }
    double probability1 = 0.3; // (double)5 / solutionLength0D / solutionLength1D;
    for (int i = 0; i < iterationsNumber; i++)
        if (bestUnfitness != 0)
        {
            FormMain.backgroundWorkerArray[0].ReportProgress(i * 100 / iterationsNumber);
            int[,] parent1;
            int[,] parent2;
            int parent1No = selectionTournamentStage1(population.Length / 2, unFitnesses, true);
            int parent2No = selectionTournamentStage1(population.Length / 2, unFitnesses, true);
            parent1 = population[parent1No];
            while (parent1No == parent2No)
                parent1No = selectionTournamentStage1(population.Length / 2, unFitnesses, true);
            parent2 = population[parent2No];

            int[,] children = crossoverUniformStage1(parent1, parent2, probability1);
            int[,] child1 = children[0];
            int[,] child2 = children[1];

            mutateStage1V3(child1, new int[] { 0, 0, 0 }, maxMutationArray, probability1);
            mutateStage1V3(child2, new int[] { 0, 0, 0 }, maxMutationArray, probability1);
            int unfitnessChild1 = unfitnessFunction(child1);
            int unfitnessChild2 = unfitnessFunction(child2);

```

```

        if (unfitnessChild1 < bestUnfitness) bestUnfitness = unfitnessChild1;
        if (unfitnessChild2 < bestUnfitness) bestUnfitness = unfitnessChild2;

        int loser1No = selectionTournamentStage1(population.Length / 2, unFitnesses, false);
        int loser2No = selectionTournamentStage1(population.Length / 2, unFitnesses, false);
        while (loser1No == loser2No)
            loser2No = selectionTournamentStage1(population.Length / 2, unFitnesses, false);

        if (unFitnesses[loser1No] > unfitnessChild1)
        {
            population[loser1No] = child1;
            unFitnesses[loser1No] = unfitnessChild1;
        }
        if (unFitnesses[loser2No] > unfitnessChild2)
        {
            population[loser2No] = child2;
            unFitnesses[loser2No] = unfitnessChild2;
        }
    }
    int bestSolutionNo = Array.IndexOf(unFitnesses, unFitnesses.Min());
    return population[bestSolutionNo];
}

public static List<List<int>>> algorithmSteadyStateStage2(List<List<int[]>> L, List<List<float>>
P, List<List<int>>> S2,
    List<Box> containersOthersList, int populationSize, int iterationsNumber, int[]
optimalMoments,
    Func<List<List<int[]>>, List<List<float>>, List<List<int>>>,int[], List<Box>, double>
unfitnessFunction)
{
    List<List<int>>>[] population = new List<List<int>>>[populationSize];
    population[0] = S2.Select(x => x.ToList()).ToList();

    double[] unFitnesses = new double[populationSize];
    unFitnesses[0] = unfitnessFunction(L, P, population[0], optimalMoments,
containersOthersList);
    //creating an initial population
    for (int i = 1; i < populationSize; i++)
    {
        population[i] = S2.Select(x => x.ToList()).ToList();
        unFitnesses[i] = unfitnessFunction(L, P, population[i], optimalMoments,
containersOthersList);
        mutateStage2(population[i], 0.1);
        unFitnesses[i] = unfitnessFunction(L, P, population[i], optimalMoments,
containersOthersList);
    }
    double probability1 = 0.01;

```

```

for (int i = 0; i < iterationsNumber; i++)
{
    FormMain.backgroundWorkerArray[0].ReportProgress(i * 100 / iterationsNumber);
    int parent1No = selectionTournamentStage2(population.Length / 2, unFitnesses, true);
    int parent2No = selectionTournamentStage2(population.Length / 2, unFitnesses, true);
    List<List<int>> parent1 = population[parent1No];
    while (parent1No == parent2No)
        parent2No = selectionTournamentStage2(population.Length / 2, unFitnesses, true);
    List<List<int>> parent2 = population[parent2No];

    List<List<int>>[] children = crossoverUniformStage2(parent1, parent2, probability1);
    List<List<int>> child1 = children[0];
    List<List<int>> child2 = children[1];
    mutateStage2(child1, probability1);
    mutateStage2(child2, probability1);
    double unfitnessChild1 = unfitnessFunction(L, P, child1, optimalMoments,
containersOthersList);
    double unfitnessChild2 = unfitnessFunction(L, P, child2, optimalMoments,
containersOthersList);

    int loser1No = selectionTournamentStage2(population.Length / 2, unFitnesses, false);
    int loser2No = selectionTournamentStage2(population.Length / 2, unFitnesses, false);
    while (loser1No == loser2No)
        loser2No = selectionTournamentStage2(population.Length / 2, unFitnesses, false);

    if (unFitnesses[loser1No] > unfitnessChild1)
    {
        population[loser1No] = child1;
        unFitnesses[loser1No] = unfitnessChild1;
    }
    if (unFitnesses[loser2No] > unfitnessChild2)
    {
        population[loser2No] = child2;
        unFitnesses[loser2No] = unfitnessChild2;
    }
}
int bestSolutionNo = Array.IndexOf(unFitnesses, unFitnesses.Min());
return population[bestSolutionNo];
}

public static void mutateStage1V3(int[,] individual, int[] rangeMin, int[] rangeMax, double
probability)
{
    int[,] returnVector = individual;
    List<int[]> unfitContainerList1 = unfitContainersList(individual);
    List<int[]> emptyLayer = Vsl.emptyLayer(FormMain.vessel.obstructionsCoordinatesList,
individual);

```



```

for (int i = 0; i < returnVector.GetLength(0); i++)
{
    int chanceMultiplier = 0;
    for (int ii = 0; ii < unfitContainerList1.Count; ii++)
    {
        if (unfitContainerList1[ii][0] == returnVector[i, 0] && unfitContainerList1[ii][1] ==
returnVector[i, 1] &&
            unfitContainerList1[ii][2] == returnVector[i, 2])
        {
            chanceMultiplier = unfitContainerList1[ii][3];
            break;
        }
    }
    if (chanceMultiplier > 0)
    {
        int rndNumber = rnd1.Next(0, 100) / (chanceMultiplier + 1);
        if (probability * 100 >= rndNumber)
        {
            rndNumber = rnd1.Next(0, emptyLayer.Count);
            returnVector[i, 0] = emptyLayer[rndNumber][0];
            returnVector[i, 1] = emptyLayer[rndNumber][1];
            returnVector[i, 2] = emptyLayer[rndNumber][2];

            if (FormMain.containerList[i].size == 40)
            {
                returnVector[i + 1 - i % 2 * 2, 0] = emptyLayer[rndNumber][0] + 1 -
emptyLayer[rndNumber][0] % 2 * 2;
                returnVector[i + 1 - i % 2 * 2, 1] = emptyLayer[rndNumber][1];
                returnVector[i + 1 - i % 2 * 2, 2] = emptyLayer[rndNumber][2];
            }
        }
    }
}

```

```

public static void mutateStage2(List<List<int>> individual, double probability)

```

```

{
    for (int i = 0; i < individual.Count; i++)
    {
        for (int ii=0; ii< individual[i].Count;ii++)
        {
            int rndValue = rnd1.Next(0,100);
            if (probability*100> rndValue)
            {
                int oldValue = individual[i][ii];
                int rndValue2 = rnd1.Next(0,individual[i].Count);
                individual[i][ii] = individual[i][rndValue2];
                individual[i][rndValue2] = oldValue;
            }
        }
    }
}

```

```

        if (i >= individual.Count/2) // if container=FEU
        {
            int firstIndex = ii + 1 - ii % 2 * 2;
            oldValue = individual[i][firstIndex];
            int secondIndex = rndValue2 + 1 - rndValue2 % 2 * 2;
            individual[i][firstIndex] = individual[i][secondIndex];
            individual[i][secondIndex] = oldValue;
        }
    }
}

public static int[,] crossoverUniformStage1(int[,] parent1, int[,] parent2, double probability)
{
    int[,] returnVector1 = new int[parent1.GetLength(0), parent1.GetLength(1)];
    returnVector1 = parent1.Clone() as int[,];
    int[,] returnVector2 = new int[parent2.GetLength(0), parent2.GetLength(1)];
    returnVector2 = parent2.Clone() as int[,];
    for (int i = 0; i < parent1.GetLength(0); i++)
    {
        int rndNumber = rnd1.Next(0, 100);
        if (probability * 100 >= rndNumber)
        {
            for (int ii = 0; ii < parent1.GetLength(1); ii++)
            {
                returnVector1[i, ii] = parent2[i, ii];
                returnVector2[i, ii] = parent1[i, ii];
            }
            if (FormMain.containerList[i].size == 40)
            {
                returnVector1[i + 1 - i % 2 * 2, 0] = parent2[i, 0] + 1 - parent2[i, 0] % 2 * 2;
                returnVector2[i + 1 - i % 2 * 2, 0] = parent1[i, 0] + 1 - parent1[i, 0] % 2 * 2;
                for (int ii = 1; ii < parent1.GetLength(1); ii++)
                {
                    returnVector1[i + 1 - i % 2 * 2, ii] = parent2[i, ii];
                    returnVector2[i + 1 - i % 2 * 2, ii] = parent1[i, ii];
                }
            }
        }
    }
    return new int[,] { returnVector1, returnVector2 };
}

public static List<List<int>>[] crossoverUniformStage2(List<List<int>> parent1,
List<List<int>> parent2, double probability)
{
    List<List<int>> child1 = parent1.Select(x => x.ToList()).ToList();

```

```

List<List<int>> child2 = parent2.Select(x => x.ToList()).ToList();

for (int i = 0; i < child1.Count; i++)
{
    for (int ii = 0; ii < child1[i].Count; ii++)
    {
        int rndValue = rnd1.Next(100);
        if (probability * 100 > rndValue)
        {
            int oldValue = child1[i][ii];
            int indexOfNew = child1[i].IndexOf(parent2[i][ii]);
            child1[i][ii] = child1[i][indexOfNew];
            child1[i][indexOfNew] = oldValue;

            int oldValue2 = child2[i][ii];
            int indexOfNew2 = child2[i].IndexOf(parent1[i][ii]);
            child2[i][ii] = child2[i][indexOfNew2];
            child2[i][indexOfNew2] = oldValue2;

            if (i >= child1.Count / 2) // if container=FEU
            {
                int firstIndex = ii + 1 - ii % 2 * 2;
                oldValue = child1[i][firstIndex];
                int secondIndex = indexOfNew + 1 - indexOfNew % 2 * 2;
                child1[i][firstIndex] = child1[i][secondIndex];
                child1[i][secondIndex] = oldValue;

                oldValue2 = child2[i][firstIndex];
                int secondIndex2 = indexOfNew2 + 1 - indexOfNew2 % 2 * 2;
                child2[i][firstIndex] = child2[i][secondIndex2];
                child2[i][secondIndex2] = oldValue2;
            }
        }
    }
}
return new List<List<int>>[] { child1, child2 };
}

public static int selectionTournamentStage1(int tournamentStages, int[] unFitnessesArray, bool
winner)
{
    int fittestNo = rnd1.Next(0, unFitnessesArray.Length);

    for (int i = 0; i < tournamentStages; i++)
    {
        int nextNo = rnd1.Next(0, unFitnessesArray.Length);

        if (winner)

```

```

        {
            if (unFitnessesArray[fittestNo] > unFitnessesArray[nextNo])
            { fittestNo = nextNo; }
        }
        else //loser
        {
            if (unFitnessesArray[fittestNo] < unFitnessesArray[nextNo])
            { fittestNo = nextNo; }
        }
    }
    return fittestNo;
}

public static int selectionTournamentStage2(int tournamentStages,double[] unFitnessesArray,
bool winner)
{
    int fittestNo = rnd1.Next(0, unFitnessesArray.Length);

    for (int i = 0; i < tournamentStages; i++)
    {
        int nextNo = rnd1.Next(0, unFitnessesArray.Length);

        if (winner)
        {
            if (unFitnessesArray[fittestNo] > unFitnessesArray[nextNo])
            { fittestNo = nextNo; }
        }
        else //loser
        {
            if (unFitnessesArray[fittestNo] < unFitnessesArray[nextNo])
            { fittestNo = nextNo; }
        }
    }
    return fittestNo;
}

public static int unFitnessStage1(int[,] individual2D)
{
    List<int[]> unfitContainersList1 = unfitContainersList(individual2D);
    int result = 0;

    foreach (int[] arr1 in unfitContainersList1)
    {
        result += arr1[3];
    }
    return result;
}

```

```

    public static double unFitnessStage2(List<List<int[]>> L, List<List<float>> P,
    List<List<int>>S2,
        int[] optimalMoments, List<Box> containersOthersList)
    {
        double[] DMxdMydMzdMxbP = FormMain.vessel.DMxMyMzMxbP(L, P, S2,
containersOthersList);
        double dMx = Math.Abs(DMxdMydMzdMxbP[1] - optimalMoments[0]);
        double dMy = Math.Abs(DMxdMydMzdMxbP[2] - optimalMoments[1]);
        double dMz = Math.Abs(DMxdMydMzdMxbP[3] - optimalMoments[2]);
        double dMxb = Math.Abs(DMxdMydMzdMxbP[4] - optimalMoments[3]);
        double pOver = Math.Abs(DMxdMydMzdMxbP[5]);
        return (1 + 2*dMx + dMxb + dMy + 2*dMz) * (1 + Math.Pow(pOver,10));
    }

    public static List<int[]> unfitContainersList(int[, ] individual2D)
    {
        List<Parameter> variableParameterList = FormMain.vessel.variablesParameterList[0];
        int array1dLength = FormMain.vessel.solutionArrayLength;
        int[] array1d = new int[array1dLength];

        for (int i = 0; i < FormMain.containerList.Count; i++)
        {
            if (FormMain.containerList[i].k >= 0)
            {
                variableParameterList[0].CurrentValue = (FormMain.containerList[i].size - 20) / 20;
                variableParameterList[1].CurrentValue =
Imdg.classesArray.IndexOf(FormMain.containerList[i].imdgType);
                variableParameterList[2].CurrentValue = individual2D[i, 0];
                variableParameterList[3].CurrentValue = individual2D[i, 1];
                variableParameterList[4].CurrentValue = individual2D[i, 2];
                //add check if i&j&k>=0 ?
                int oneDIndex = Parameter.ConvertNDto1D(variableParameterList);
                array1d[oneDIndex] = 1;
            }
        }
        //adding postitions taken by obstructions
        for (int i=0;i< array1dLength;i++)
        {
            array1d[i] = Math.Max(array1d[i], FormMain.vessel.emptySolutionArray[i]);
        }
        int[, ] inequalitiesMatrix = FormMain.vessel.inequalitiesArray;
        int[] inequalitiesConstantsMatrix = FormMain.vessel.inequalitiesConstantsArray;

        int[] matrixProductArray;
        matrixProductArray = Matrix.mult2Dx1D(inequalitiesMatrix, array1d);
        List<int[]> unfitContainersList = new List<int[]>();
        for (int i = 0; i < inequalitiesConstantsMatrix.Length; i++)
        {

```

```

if (inequalitiesConstantsMatrix[i] < matrixProductArray[i])
{
    for (int ii=0; ii< array1d.Length; ii++)
    {
        if (inequalitiesMatrix[i,ii] != 0 && array1d[ii] != 0)
        {
            Parameter.Convert1DtoND(variableParameterList, ii);
            int contI = variableParameterList[2].CurrentValue;
            int contJ = variableParameterList[3].CurrentValue;
            int contK = variableParameterList[4].CurrentValue;
            int[] contCoordinates = new int[] {contI, contJ, contK, 1};
            bool isNotInList = true;
            for (int iii=0; iii< unfitContainersList.Count;iii++)
            {
                if (unfitContainersList[iii][0] == contI && unfitContainersList[iii][1] == contJ
&&
                    unfitContainersList[iii][2] == contK)
                {
                    unfitContainersList[iii][3]++;
                    isNotInList = false;
                    break;
                }
            }
            if (isNotInList)
            {
                unfitContainersList.Add(contCoordinates);
            }
        }
    }
}
return unfitContainersList;
}

```