

СИМЕОН ДЕКОВ МОНОВ

**ЕФЕКТИВНО ПЛАНИРАНЕ
И РАЗПРЕДЕЛЕНИЕ НА ЗАДАЧИ**

А В Т О Р Е Ф Е Р А Т

на дисертационен труд
за присъждане на образователната и научна степен „доктор“
по област на висше образование: 4. Природни науки,
математика и информатика
професионално направление: 4.6. Информатика и компютърни науки,
докторска програма Информатика

Научни ръководители: проф. д-р Асен Рахнев
доц. д-р Николай Павлов

Пловдив, 2022 г.

Съдържание

Обща характеристика на дисертационния труд	5
<i>Актуалност на проблема.....</i>	<i>5</i>
<i>Цел и задачи на дисертационния труд</i>	<i>5</i>
<i>Структура и обем на дисертационния труд.....</i>	<i>6</i>
Кратко съдържание на дисертационния труд	7
Глава I. Изследване на проблема по планиране и разпределение на задачи в контейнерен терминал.....	7
<i>Операции в контейнерните терминали.....</i>	<i>8</i>
<i>Преглед на изследванията, свързани с планирането на операциите в контейнерните пристанища</i>	<i>11</i>
<i>Използвани технологии.....</i>	<i>11</i>
<i>Резултати и изводи.....</i>	<i>12</i>
Глава II. Ефективно управление на маршрути на транспортно оборудване с помощта на алчен алгоритъм	13
<i>Общ преглед на алчните алгоритми</i>	<i>13</i>
<i>Оптимизация на транспортни задачи</i>	<i>14</i>
<i>Премествания и обща цена спрямо график на операциите</i>	<i>14</i>
<i>Алгоритъм.....</i>	<i>15</i>
<i>Унгарски метод.....</i>	<i>16</i>
<i>Резултати и изводи.....</i>	<i>16</i>
Глава III. Генетичен алгоритъм за ефективно управление на маршрути на транспортно оборудване.....	17
<i>Общ преглед на генетичните алгоритми</i>	<i>17</i>
<i>Генетичен алгоритъм.....</i>	<i>17</i>
<i>Резултати и изводи.....</i>	<i>19</i>
Глава IV. Табу търсене за ефективно управление на маршрути на транспортно оборудване.....	20
<i>Локална оптимизация.....</i>	<i>20</i>
<i>Глобална оптимизация</i>	<i>21</i>
<i>Локална или глобална оптимизация</i>	<i>21</i>
<i>Табу търсене.....</i>	<i>21</i>
<i>Табу търсене за планиране и разпределение на задачи, свързани с движението на контейнерите</i>	<i>22</i>
<i>Авторски алгоритъм за табу търсене за подобряване на резултатите от алгоритмите, разгледани в Глава II и Глава III.....</i>	<i>22</i>
<i>Резултати и изводи.....</i>	<i>23</i>
Глава V. Симулатор на операциите в пристанище.....	24
<i>Резултати и изводи.....</i>	<i>25</i>
Заклучение	25
Перспективи	26
Основни приноси на дисертационния труд.....	26
Публикации по дисертационния труд.....	27
Апробация	27
Библиография	27

Списък на съкращенията

ALV – Automated Lifting Vehicle
ASC – Automated Stacking Crane
B&B – Branch and Bound
BAP – Berth Allocation Problem
BFGS – Broyden–Fletcher–Goldfarb–Shanno
CHE – Container Handling Equipment
FCFS – First Come First Serve
GA – Genetic Algorithm
GC – Garbage Collector
JSON – JavaScript Object Notation
JVM – Java Virtual Machine
MIP - Mixed Integer Program
NP – Nondeterministic Polynomial (time)
OR – Operational research
PSGC – Parallel Scavenge Garbage Collector
PSOGAO – Particle Swarm Optimization with Genetic Algorithm Operators
QCAP – Quay Crane Assignment Problem
RMGC – Rail Mounted Gantry Crane
RTGC – Rubber Tired Gantry Crane
SC – Straddle Carrier
SD – System Dynamics
TEU – Twenty-feet Equivalent Unit
TS – Tabu Search
YCS – Yard Crane Scheduling
YTS – Yard Truck Scheduling
АА – Алчен Алгоритъм
ГА – Генетичен Алгоритъм
ОПК – Оборудване за Преместване на Контейнери

Обща характеристика на дисертационния труд

Актуалност на проблема

Изготвянето на план за изпълнение на задачи по време спрямо наличните технически средства за тяхната реализация е ключов инструмент за повишаване ефективността на редица дейности. Решения на този проблем се търсят с помощта на програмни алгоритми от десетилетия [Dantzig and Ramser, 1959]. Сложността на проблема произлиза не само от огромният брой възможни решения, които трябва да се изследват, но и от спецификите на всяка отделна икономическа дейност. Последните правят невъзможно намирането на ефективно универсално решение. Всяка отделна икономическа дейност се води от свои правила, които налагат индивидуален подход и създаване на специфични за тях модели и алгоритми.

Има клас задачи, които работят с напълно постоянни или слабо променящи се във времето данни; те могат да бъдат решени чрез еднократно изготвяне на план за изпълнение и използване на наличните ресурси.

Съществуват икономически процеси, които са силно променливи във времето и се характеризират с висока степен на динамика – възможно е да настъпят неочаквани промени по време на провеждане на дейността и изпълнението на задачите, които да направят предварително изготвения план най-малкото не само неефективен, но дори и невъзможен за изпълнение. Динамичните дейности изискват алгоритми, способни да произведат за много кратко време ново решение, което да вземе предвид настъпилата промяна в обстоятелствата. Времето за изготвяне на този нов план трябва да е от порядъка на минути, тъй като по-голямо забавяне води до ефективно прекъсване на цялата дейност на предприятието.

Ефективността на икономическите процеси не е едностранен проблем. Тя може да се измерва чрез различни показатели, всеки от които да бъде обект на оптимизация (време за цялостно изпълнение до завършване на всички задачи, енергийна ефективност, необходими ресурси, точност на изпълнение спрямо графика за предоставяне на резултатите на контрагентите, и др.). Следователно, тези оптимизационни задачи се характеризират със собствени функции за пресмятане на тяхната цена.

В настоящия дисертационен труд се изследват нови модели на алгоритми за ефективно планиране и разпределение на задачи във високо динамична среда, в която те следват определени икономически закони. Избраният подход може да се приложи в различни сфери на икономическа дейност след подходящо адаптиране на избрани операции на алгоритмите.

За проверка на приложимостта на новите модели алгоритми в дисертационния труд е избрана конкретна област, а именно – превоз на контейнери в пристанищен товарен терминал.

Цел и задачи на дисертационния труд

Основната цел на дисертационния труд е да се разработят и реализират нови, по-ефективни решения за планиране и разпределение на задачи, свързани с движението на контейнерите в контейнерни пристанищни терминали, както и симулатор за тяхното реалистично тестване.

Основните задачи са:

1. Проучване на съвременни подходи и решения на задачи, свързани с движението на контейнерите в контейнерните терминали. Подзадачи:
 - 1.1. Обзор на различните видове операции и използвано оборудване в един контейнерен терминал;
 - 1.2. Преглед на изследванията по планиране на наземния транспорт, на задачите на крановете в блоковете за съхранение;
 - 1.3. Сравнителен анализ на основните съвременни подходи и решения за планиране и разпределение на задачи, свързани с движението на контейнерите.
2. Проектиране на подходящи алгоритми за планиране и разпределение на задачи и тяхното оптимизиране. Подзадачи.
 - 2.1. Проектиране на евристичен алгоритъм, базиран на техниките на алчните алгоритми;
 - 2.2. Проектиране на евристичен алгоритъм, базиран на техниките на генетичните алгоритми;
 - 2.3. Проектиране на евристичен алгоритъм, базиран на техниките за Табу търсене.
3. Реализация на разработените подобрени алгоритми за планиране и разпределение на задачи на кранове и камиони за преместване на контейнери в пристанище.
4. Разработване на симулатор на операциите в пристанище с цел изпълнение на алгоритмите, когато преместванията на контейнерите са налични, както и симулиране изпълнението на задачите от оборудването във времето.
5. Проверка на разработените евристични алгоритми за планиране и разпределение на задачи по преместване на контейнери в практически условия. Сравнение и анализ на постигнатите резултати.

Структура и обем на дисертационния труд

Дисертационният труд се състои от увод, пет глави, заключение, библиография, списък на авторските публикации по темата и апробация на резултатите, и е в обем от 105 страници. Към работата са добавени приноси на дисертационния труд, декларация за оригиналност и перспективи за бъдещо развитие. Списъкът с реферираните източници се състои от 84 заглавия, от които 1 на кирилица и 81 на латиница, 2 интернет адреса.

В **Първа глава** е направен преглед на съществуващите операции в контейнерните пристанища. Проучени са съвременни подходи и решения за планиране и разпределение на задачи, свързани с движението на контейнерите, като е извършен сравнителен анализ на тези подходи. Направен е обзор на операциите, алгоритмите и технологиите с акцент върху планирането на транспорта (YTS) и работата на крановете в блоковете за съхранение на контейнери (YCS).

Във **Втора глава** е представен разработен авторски евристичен алгоритъм за маршрутизиране на движението на транспортно оборудване в пристанище на

контейнерни терминали на принципа на „алчните“ алгоритми. Алгоритмът работи на няколко итерации и използва Унгарския метод за избиране на локално оптимално решение за текущата итерация. Представеният алгоритъм взема в предвид ограничения като зависимости между задачите и използва реалистичен ценови модел за оценка на задачите.

В **Трета глава** е представен разработен авторски евристичен алгоритъм за маршрутизиране на движението на транспортно оборудване в пристанище на контейнерни терминали на принципа на генетичните алгоритми с цел подобряване на резултатите, получени във Втора глава. При инициализацията алгоритъмът използва както случайно генерирани решения, така и намерено „добро“ решение от „алчния“ авторски алгоритъм (Глава II) за насочване на търсенето, което помага за по-бързото намиране на по-добро решение на проблема. Представеният алгоритъм взема предвид ограничения като зависимости между задачите и използва реалистичен ценови модел за оценка на задачите.

В **Четвърта глава** е представен разработен авторски евристичен алгоритъм, който реализира Табу търсене за по-нататъшна оптимизация на резултатите от алгоритмите, представени във Втора и Трета глава. Табу търсенето е метоевристичен метод за локално търсене. Алчният и генетичният алгоритъм обикновено намират „добро“ решение на проблем, което може да бъде допълнително оптимизирано чрез методи за локално търсене. Разработеният алгоритъм използва Табу търсене чрез използване на генетични оператори, разработени в Трета глава.

В **Пета глава** е представен разработен симулатор на операциите в пристанище. Симулаторът изпълнява алгоритмите, когато има налични премествания, и симулира изпълнението на задачите от оборудването в пристанището; генерира статистика на общата цената за изпълнение на задачите и времето за пътуване без контейнер. Може да извежда подробна информация за отделните изпълнявани операции.

В **Заклучението** се съдържа резюме на получените резултати, претенциите на автора за научно-приложни приноси, апробация с участия в проекти и публикациите по дисертационния труд, декларация за оригиналност. Описани са възможностите за бъдещо развитие за по-нататъшно оптимизиране и решаване на проблемите при планирането и разпределението на задачите на транспортно оборудване в пристанища на контейнерни терминали.

Списъкът на авторските публикации по дисертационния труд се състои от 2 заглавия, и двете в международни научни списания с (SJR 2021:0.214).

Кратко съдържание на дисертационния труд

Глава I. Изследване на проблема по планиране и разпределение на задачи в контейнерен терминал

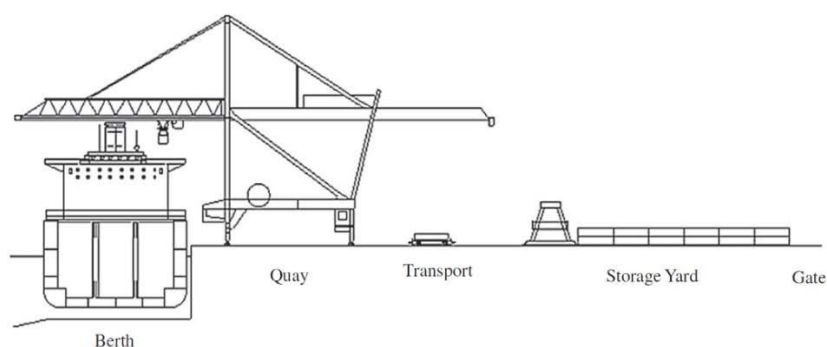
В последните години увеличаването на търговията с контейнери и недостига на капацитет в пристанищата създава все по-належаща нужда за подобряване и оптимизиране на операциите в контейнерните терминали. От 2007 до 2021 година трафикът на контейнери в световен мащаб се е увеличил от 485 милиона TEU до 811 милиона TEU [UNCTAD, 2008; UNCTAD, 2021]. Тенденцията е увеличаване на

размера на контейнерните кораби (големината на най-големия по капацитет се е увеличила с 10.9% през 2021 година). От една страна, това увеличаване води до по-добри икономически ползи за операторите на кораби, но от друга страна води до повече проблеми в логистиката в пристанищата и наземните оператори, като фактически увеличава крайната цена на доставките.

Увеличаването на средния размер на корабите често води до пиково търсене на камиони, дворно пространство и интермодални връзки, както и допълнителни инвестиции за драгиране и закупуване на по-големи кранове. Операторите на пристанища започват все повече да търсят и инвестират в по-оптимални решения за оптимизация на движението на контейнерите, което спомага за по-малки разходи, както и достигане на заложените ЕКО норми.

Операции в контейнерните терминали

Контейнерните терминали могат да се разделят на пет главни области, показани на Фигура 1 [Carlo et al., 2014a]: корабна стоянка (berth), кей (quay), транспорт (transport), блок за съхранение (storage yard) и порта (terminal gate). Корабната стоянка и кея са крайбрежната (water side) част на терминала. Блокът и портата са наземната част, а транспортът е връзката между тях. Много добър обзор на транспортните операции правят в [Carlo et al., 2014b].



Фигура 1. Главни области на контейнерните терминали

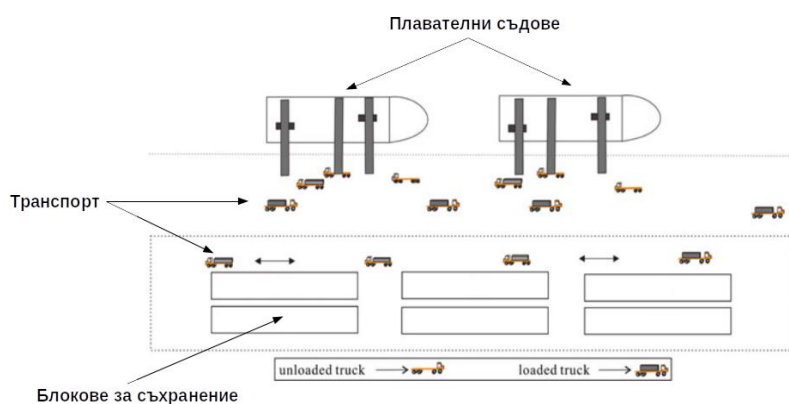
Операциите в един терминал могат да се разделят на 4 вида:

1. *Разтоварване на контейнери от плавателен съд*: **закотвяне** на стоянка на пристигащите плавателни съдове; **разтоварване** на контейнерите в определен ред в зависимост от плана; **превоз** към крайбрежна трансферна област; **преместване** от трансферната област вътре в блока до предварително разпределените места; **съхраняване** в блока до времето за натоварване; **преместване** в обратния ред към крайбрежна или наземна трансферна област за натоварване на друг кораб или камион/влак.
2. *Натоварване на контейнери на плавателен съд* – в обратен на разтоварването ред при спазване на определен план.
3. *Транспорт*.
4. *Операции между блоковете*.

Крайбрежни операции – операции в крайбрежната част на терминала, свързани с плавателния съд – разтоварване и натоварване на/от кея.

[Bierwirth and Meisel, 2010; Bierwirth and Meisel, 2015] правят много добър обзор на проблемите, свързани с разпределението на задачите на пристанищните оператори. [Chargui et al., 2021] и [He et al., 2021] проучват влиянието на разпределението на стоянките върху ефективността на операторите на крановете и количеството камиони за работа в трансферната зона. [Abdel-Fattah et al., 2013] предлагат методология за решение за Berth Allocation Problem (BAP) и Quay Crane Assignment Problem (QCAP) чрез интеграция на operational research (OR) техники и system dynamics (SD) техники. Като типична NP-пълна задача, различни трудове предлагат решения основно чрез използване на мета-евристични алгоритми [Meisel and Bierwirth, 2009; Lee et al., 2006; Olteanu et al., 2018; Msakni et al., 2018; Lee et al., 2010; Zhao et al., 2019; Expósito-Izquierdo et al., 2011; Tasoglu and Yildiz, 2019; Chen et al., 2020; Bacalhau et al., 2021].

Транспорт – връзката между кея и блока (Фигура 2 [Niu et al., 2017]).



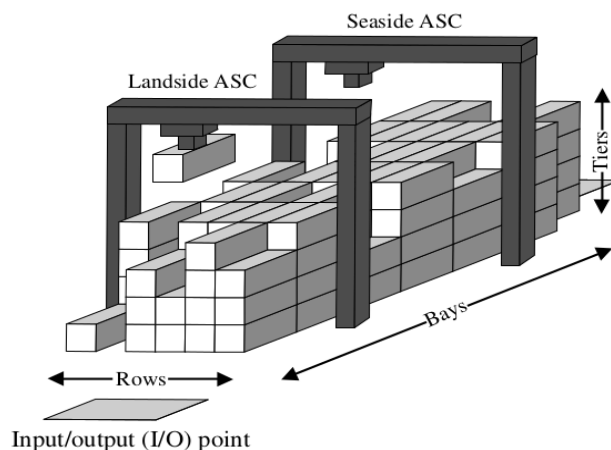
Фигура 2. Транспорт

Крановете на кея работят по план и наземният транспорт трябва да се синхронизира с тях. В зависимост от типа на оборудването, кранът може да постави/вземе контейнер върху/от транспорта директно или да постави контейнера на кея, откъдето да бъде взет. Съществуват няколко вида автоматично и неавтоматично (ръчно управляемо) оборудване за придвижване на контейнерите: *Self-lifting* – могат сами да повдигат контейнерите от земята с подреждане или без подреждане един върху друг; *Non-lifting* – изискват друго оборудване за натоварване/разтоварване (кранове на кея или кранове в блоковете за съхранение).

Блокове за съхранение (Yard storage) – в тях се случват едни от най-сложните операции в едно пристанище; много добър обзор се прави в [Carlo et al., 2014a].

Оборудване – едни от най-използваните са *порталните кранове* (Gantry Cranes). Делят се на два вида: *гумирани* (RTGCs) – обикновено са ръчно управляеми и имат възможност да пътуват свободно между отделните блокове за съхранение; *монтирани на релса* (RMGCs) – движението е лимитирано от релси. Изследване от [Wiese et al., 2009; Wiese et al., 2011] показва, че RTGC крановете са най-използваното оборудване с 63.2% от 114 изследвани контейнерни терминала. Автоматизирани RTMC кранове, известни като Автоматизирани Кранове за Подреждане (Automated Stacking Cranes, ASC), могат изцяло да оперират автоматично без човешка намеса при работа с автоматизиран транспорт; използват се в 6.1% от терминалите, основно намиращи се в Европа.

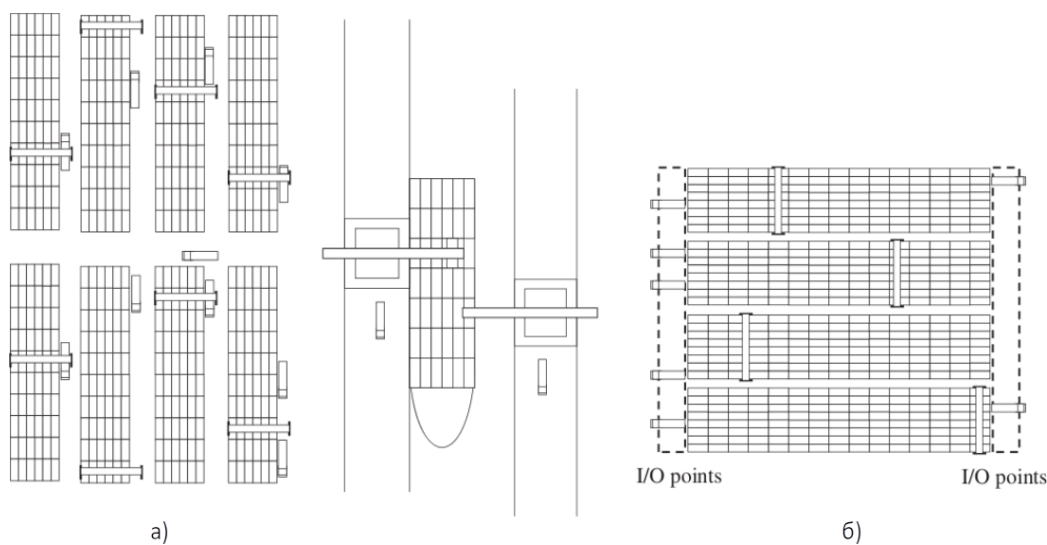
Разположение – типичното се състои от множество правоъгълни блокове, като всеки блок съдържа множество редове (rows) и стекове (stacks). Контейнерите може да се слагат един върху друг и да се натрупват на купчини (tiers) (Фигура 3 [Gharehgozli et al., 2015]). Броят на редовете, стековете и нивата зависи от използваното оборудване.



Фигура 3. Разположение на блок за съхранение

Съществуват две главни конфигурации на блоковете за съхранение [Liu et al., 2004; Petering, 2009; Lee and Kim, 2013] според мястото на входно/изходните зони (зоните за товарене и разтоварване на контейнерите от вътрешния и външния транспорт).

При първата конфигурация, основно използвана с неавтоматизирани кранове, блоковете имат резервирани един или повече реда за движение на камионите за товарене/разтоварване (Фигура 4а [Carlo et al., 2014a]) – камионите пътуват до стека за обслужване и обслужващият тази зона от блока кран се придвижва над камиона и поставя или повдига контейнера (много популярна в Азия; Азиатска конфигурация).



Фигура 4. Азиатска (а) и Европейска (б) конфигурации

Втората конфигурация е използвана основно с автоматизирани кранове (ASC) (Фигура 4б). Входно/изходните зони са разположени в двата края на блока: *крайбрежна трансферна зона* – разположена от към страната на кея (waterside

transfer area), използва се за трансфер на контейнери между блока и кея; *наземна трансферна зона* – разположена от другата страна на блока (landside transfer area) и се използва от външния транспорт да доставя/извозва контейнери към/извън пристанището, обслужва се от външни камиони за извозване на контейнерите. Този вид конфигурация за първи път е въведена в Европейски пристанища и често е наричана Европейка конфигурация.

Разпределение на места в блоковете за съхранение – една от важните операции за намаляване времето за доставяне и разместване на контейнерите. [Bazzazi et al., 2009] формулират задачата като обобщена задача за назначенията (Generalized assignment problem), което я поставя към задачите от NP-пълна категория, и предлагат решение чрез генетичен алгоритъм. Различни автори, работили по този проблем, използват мета-евристични алгоритми [Kim and Park, 2003; Zhang et al., 2003]. [Ayachi et al., 2013] разглеждат различни видове контейнери и използват генетичен алгоритъм.

Преглед на изследванията, свързани с планирането на операциите в контейнерните пристанища

Настоящият дисертационен труд разглежда планирането на наземния транспорт и на работата на крановете в блоковете за съхранение (YTS и YCS).

Планиране на наземния транспорт (YTS) – това е доказана NP-пълна задача [Bish et al., 2001] и като такава повечето от решенията използват мета-евристични решения. [Nguyen and Kim, 2009] разглеждат проблема за планиране работата на ALV-та и свеждат задачата до подобна за пътуващите пътници, като я решават чрез евристика. Други автори разглеждат проблема за планирането на автопарк от камиони в контейнерните терминали за минимизиране на разходите; формулират задачата като смесена целочислена програма (mixed integer program, MIP) и я решават: чрез генетичен алгоритъм (GA) [Ng et al., 2007]; чрез хибриден генетичен алгоритъм [Wang et al., 2015]; чрез генетичен алгоритъм (GA) и модифициран евристичен алгоритъм [Cao et al., 2010].

[Zhang et al., 2016] използват алгоритъм на изкуствените пчелни семейства (Artificial Bee Colony Optimization) за решаването на YTS-SAP проблема, като сравняват резултатите с тези от генетичен алгоритъм и по метода на оптимизация на рояка на частици (Particle Swarm Optimization).

Планиране на задачите на крановете в блоковете за съхранение (YCS) – [Zhu and Lim, 2006] формулират проблема като целочислена програма и доказват, че е NP-пълна задача; решават я чрез Branch and Bound (B&B) алгоритъм за намиране на оптимално решение, както и с евристичен алгоритъм. Други автори предлагат ефективен модел за планирането на дворни кранове, като взимат предвид някои оперативни ограничения [Li et al., 2009; Wu et al., 2015; He et al., 2019; Mak and Sun, 2009]. Приведено е доказателството на Zhu и Lim [Zhu and Lim, 2006], че проблемът за разпределение на задачите на кранове е NP-пълен.

Използвани технологии

Предложените алгоритми в този дисертационен труд са реализирани на Java – обектно ориентиран език за програмиране от високо ниво, проектиран така, че да има

възможно май-малко платформени зависимости, компилираният код на Java може да работи на множество платформи (поддържани от Java) без необходимост от прекомпилиране. Програмите, написани на Java, се компилират до език на междинно ниво (подобен на асемблер), наречен байткод (bytecode) – не се изпълнява директно от процесора, а от виртуална машина, наречена Java Virtual Machine (JVM).

Java виртуална машина (Java Virtual Machine) – изпълнява Java байт код, който е един и същ, независимо на каква платформа и хардуер се изпълнява програмата. JVM съдържа Just In Time (JIT) компилатор – превежда Java byte code до машинен код на процесора, на който се изпълнява програмата в реално време, както и кешира този код. Използването на междинен байт код позволява програмите, написани на Java, да се изпълняват на всички платформи, за които се поддържа JVM.

Библиотеки с класове – Java предоставя собствени стандартни библиотеки от класове (Java Class Library) за изпълнение на различни общи функционалности, като обработка на низове, обработка на списъци и други структури от данни, достъп до файлова система, достъп до мрежови услуги и много други (платформата на Java е независима от операционната система и програмите не могат да използват библиотеките на системата).

Система за почистване на паметта (Garbage Collector, GC) – JVM има автоматична система за почистване и поддържане из на паметта, която спестява тези задачи на програмистите. Всеки обект в Java заема определена памет, заделена в т.н. купчина памет (memory heap). Системата за почистване минава през купчината памет на определени периоди, проверява кои обекти вече не се използват и ги почиства (освобождава паметта), за да може тази памет да се използва от други обекти. В момента JVM поддържа няколко алгоритъма за почистване на паметта. Изборът на правилния алгоритъм е важна част от дизайна на софтуерните системи и зависи много от решаваните от системите проблеми – GC може много да влияе на скоростта на изпълнение и ефективното използване на паметта при изпълнение на различни алгоритми, особено такива, които създават много временни структури и обекти като Генетичните алгоритми и други алгоритми за търсене.

Различни автори правят много добър анализ както на вградените в JVM, така и на други, експериментални имплементации на различните алгоритми за почистване на паметта: [Pufek et al., 2019; Li et al., 2018; Lengauer et al., 2017]. Представено е описание на различните GC алгоритми.

Резултати и изводи

От направения преглед на съвременните изследвания в областта на планиране на задачите на машини в контейнерен пристанищен терминал, можем да направим следните изводи:

Проблемът е доказан като NP-пълна задача от [Zhu and Lim, 2006] – задачи, за които е доказано, че няма известни алгоритми за намиране на оптимално решение в полиномиално време, поради което като подход се използват евристични алгоритми за достигане на ефективни решения, близки до оптималното.

Нито едно от разгледаните изследвания не отчита зависимостта между отделните задачи за изпълнение от машините. Дадена задача може да зависи от редица фактори, като например:

- Задачи, които обслужват един и същ контейнер, трябва да бъдат правилно подредени във времето съгласно пътя на контейнера;
- Дадена задача може да започне само след като друга задача е приключила. Например, възможно е машина да трябва да изчака дадено пространство да се освободи от друга машина, за да започне своята задача.

Тези зависимости усложняват допълнително задачата, но имат критично значение за приложимостта на решението в практиката. Без тях решението има предимно теоретично или силно ограничено приложение. Представените във втора, трета и четвърта глава алгоритми са проектирани и разработени да отчитат изброените зависимости.

Мнозинството от разгледаните изследвания се стремят да оптимизират общото време на изпълнение на всички задачи. Докато това е валиден подход за подобряване на ефективността на пристанището, предложените в настоящия дисертационен труд решения изследват проблема от две други гледни точки:

1. Цена на времето – някои контейнери се характеризират с много строги изисквания относно времето им на обработка и транспорт (хладилен контейнер има определен срок, в който трябва да бъде преместен до източник на електроенергия); превишаването на този срок носи съществена материална отговорност на транспортния оператор;
2. Време на движение на машините без натоварен контейнер – следва да се ограничи с цел минимизиране разходите на енергия; енергийната ефективност сама по себе си може да бъде от значителна полза за транспортния оператор.

Глава II. Ефективно управление на маршрути на транспортно оборудване с помощта на алчен алгоритъм

Общ преглед на алчните алгоритми

Оптимизационните алгоритми обикновено преминават през поредица от стъпки, на всяка от които трябва да направят избор от множество възможности. Алчните (greedy) алгоритми (АА) винаги правят избора, който изглежда най-добър за момента, т.е. те правят локално оптимален избор с надеждата той да доведе до цялостно оптимално решение [Cormen et al., 2001]. Състоят се от следните компоненти: *Множество* от кандидати за решение (крайното решение е изградено от това множество); *Множество от кандидати*, които вече са включени в решението и не трябва да се използват наново; *Обективна функция* – присвоява числова стойност към решение или частично решение (кандидатите за решение се избират според тяхната минимална или максимална числова стойност); *Селектираща функция* – избира най-добрия за момента кандидат, който да бъде добавен към решението; *Оценяваща функция* – определя дали даден кандидат може да бъде използван като локално оптимизационно решение; *Предикатна функция* – определя дали намереното до момента решение е действително такова. АА не винаги намират оптималните

решения, тъй като изследват пълния набор от данни и всички възможни решения, но често намират решение, близко до оптималното [Наков и Добриков, 2015], при това за сравнително кратко време за изпълнение.

Известни приложения на АА – Алгоритъм на Дейкстра за намиране на най-кратки пътища в граф от даден връх до всички останали [Dijkstra, 1959]; Намиране на минимално покриващо дърво (алгоритми на Прим и Крускал); Кодирание/компресия на Хъфман [Huffman, 1952]; Разпределение и планиране на задачи върху налични ресурси.

Оптимизация на транспортни задачи

Използван е „алчният“ подход за маршрутизиране на движението на транспортно оборудване в пристанището на контейнерен терминал или оборудване за преместване на контейнери (ОПК). За всяка ОПК единица трябва да се намери подреден набор от премествания, всяко от които има начален час и продължителност. АА извършва няколко итерации (с най-добър ход за ОПК задание) до насрочване на всички премествания. Изборът на задание на всяка стъпка се осъществява с помощта на унгарския метод за присвояване на работа [Kuhn, 1955]. Предложеното от автора на дисертационния труд решение взема предвид някои ограничения поради спецификата на операциите на ОПК: *Достъпност* на зоната на местоназначение от определен ОПК; *Зависимости между преместванията* – от времето на вдигане или доставяне на друго преместване.

Премествания и обща цена спрямо график на операциите

Преместванията на ОПК определят предварително зададени задачи/работа, с които оборудването да се справи. Всяко преместване / движение има: налично време (когато е достъпно за изпълнение); място за вземане (логическа позиция на контейнера в двора); място за доставка (логическа позиция в двора за доставяне); желано време на доставяне, тип поток и предхождащи премествания (зависимости).

За всяко преместване/работа се определя цена. Общата цена на графика за работа е сумата от цените на всички премествания. Постигането на оптималния график е да се разпределят така задачите, че общата цена за изпълнение да е минимална. За изчисляване на цената се използва опростена полиномна функция: $f(x) = a \cdot x^b$.

За всяко преместване се разглеждат две функции на разходите:

- *Функция за времето за пътуване без превоз на контейнер* – разходите, свързани с пътуване на ОПК до място за получаване, без превоз на контейнер (a и b са параметри на функцията, а t е времето за пътуване на ОПК от текущото до мястото на вдигане):

$$C_{empty} = a_{empty} \cdot t^{b_{empty}}$$

- *Функция на разходите за навременност* – разходите за преместване на контейнер от място за получаване до място за доставяне (a и b са параметри на функцията, а Δt е разликата между желаното и реалното време за доставяне):

$$C_{timeliness} = a_{timeliness} \cdot \Delta t^{b_{timeliness}}$$

Цената за всяко преместване е $C_i = C_{empty_i} + C_{timeliness_i}$ и общата цена на графика е $C = \sum_{i=1}^m C_i$.

Алгоритъм

Идеята на представения от докторанта алгоритъм е да бъдат зададени работни места/премествания на транспортните единици стъпка по стъпка в множество итерации. Всяка стъпка изчислява най-добрите задачи за наличните в момента такива и ги присвоява на транспортните единици в съответното им налично време (текущо време или времето, когато те ще завършат текущо изпълняваната си задача). Всяка единица налично време се преизчислява, като се вземат предвид новите възложени работни места. Задачите се премахват от списъка с налични работни места и алгоритъмът се повтаря, докато няма повече достъпни работни места за възлагане. Високото ниво на „алчната“ логика за планиране на премествания и отчитане на всички ограничения за $N > 0$ брой транспортни единици и $M > 0$ брой премествания се извършва в следните стъпки:

1. Инициализира се празен списък с планирани премествания; ще съдържа окончателния график, когато приключи алгоритъмът;
2. Инициализира се списък с всички налични премествания, които е възможно да бъдат присвоени при текущата итерация. K – текущ размер на списъка;
3. За всяко налично в списъка преместване се изчисляват разходите от всяка единица транспортно оборудване при отчитане на вече присвоените премествания за всички единици; генерира се матрица $K \times N$: при *невъзможност* преместване да бъде обработено се приписва положителна безкрайна стойност на разходите – маркер при планиране на преместване (стъпка 5); при *зависимости* от други премествания на възможно за текущата итерация преместване съответното начално време за планиране е максималното време за изпълнение на всичките му зависими ходове;
4. Избират се премествания, които са най-добрите и с най-много N , да се присвоят на всяка единица, използвайки унгарския алгоритъм [Kuhn, 1955] – решава задачата за присвояване в $O(n^3)$ (n е по-голямото от K и N). Възможно е: 1) $K < N$ – ще има единици без присвояване на преместване при текущата итерация; 2) $K = N$ – всяко ОПК ще получи задание; 3) $K > N$ – всяко ОПК ще получи най-доброто (по цена) задание;
5. Добавят се заданията от стъпка 4 към списъка с планирани премествания (1); включват се само задачи с цена, различна от нашия маркер;
6. Премахват се назначените ходове от списъка с налични премествания (2);
7. Ако списъкът с наличните премествания не е празен, се преминава към стъпка 2;
8. Списъкът с планираните премествания (1) е решението.

Предложеният от нас алгоритъм използва унгарския метод, известен като Унгарски алгоритъм, прилаган за изчисляване на локално оптимално присвояване на всяка итерация.

Унгарски метод

Унгарският метод [Kuhn, 1955; Kuhn, 1956] е комбинаторен оптимизационен алгоритъм [Papadimitriou and Steiglitz, 1982], базиран на две теореми [König, 1931; Egerváry, 1931], който решава задачата за назначенията в полиномиално време. Методът е разработен и публикуван от Харолд Кун през 1955 година. През 1957 [Munkres, 1957] показва, че алгоритъмът е строго полиномиален и е с $O(n^4)$ сложност (Kuhn-Munkres алгоритъм). По-късно [Tomizawa, 1971] и [Edmonds and Karp, 1972] подобряват алгоритъма до $O(n^3)$ сложност. Следват редица изследвания и модификации за подобряване работата на Унгарския метод: [Date and Nagi, 2016; Munaro, 2020; Kumar, 2006].

Резултати и изводи

В тази глава е представен нов, авторски, алчен алгоритъм за оптимизиране на движението на транспортно оборудване в пристанища на контейнерни терминали, който използва Унгарския метод за локални, оптимални избори. Алгоритъмът взема предвид спецификата при оценяването на задачите, както и поддържа зависимости между отделните премествания. Разработеният алгоритъм е тестван с 14 различни сценария. Тестовите са проведени върху реални данни и са сравнени с данните на съществуващ, досега използван в практиката, алгоритъм First-Come-First-Serve (FCFS). Направените тестове показват, че предложеният алгоритъм намалява общата цена на графика над FCFS с 36,05% средна и 22,85% медиана (Таблица 1), и общото празно време за пътуване с 20,84% средна и 25,21% медиана (Таблица 2).

Таблица 1. Сравнение на резултатите според цената на времето за придвижване

Брой премествания	Брой ОПК единици	FCFS	Алчен алгоритъм	Подобрение %
100	1	937603.25	755584.50	19.41%
100	2	889288.75	687647.50	22.67%
100	4	773320.50	633981.25	18.02%
1000	20	7802013.00	1323434.75	83.04%
1000	40	2644572.00	246562.50	90.68%
100	1	993329.75	766315.50	22.85%
100	2	966117.75	588532.75	39.08%
100	4	892400.00	290505.00	67.45%
1330	21	1709383.75	1596160.00	6.62%
100	1	1005448.00	831735.50	17.28%
100	2	1005324.75	687989.00	31.57%
100	4	1005344.25	534292.00	46.85%
1001	20	10494282.50	8095749.25	22.86%
42	2	30797.00	25765.00	16.34%
Средно подобрение %:				36.05%
Медиана на подобрение %:				22.85%

Алгоритъмът е успешно интегриран в системата за планиране на оборудване за обработка на контейнери в порт за контейнерен терминал.

Таблица 2. Сравнение на резултатите според времето за пътуване без превоз на контейнер

Брой премествания	Брой ОПК единици	FCFS	Алчен алгоритъм	Подобрение %
100	1	04:11:41	01:20:16	68.11%
100	2	04:21:35	01:18:16	70.08%
100	4	03:41:42	01:17:03	65.25%

1000	20	17:50:04	15:00:17	15.87%
1000	40	16:47:20	11:11:20	33.36%
100	1	03:41:59	01:24:28	61.95%
100	2	03:49:55	01:50:37	51.89%
100	4	03:45:32	02:05:06	44.53%
1330	21	15:30:13	22:16:40	-43.69%
100	1	06:03:12	05:49:26	3.79%
100	2	05:54:59	05:48:44	1.76%
100	4	05:56:17	05:38:48	4.91%
1001	20	10:05:30	20:29:21	-103.03%
42	2	01:36:20	01:19:54	17.06%
Средно подобрене %:				20.84%
Медиана на подобрене %:				25.21%

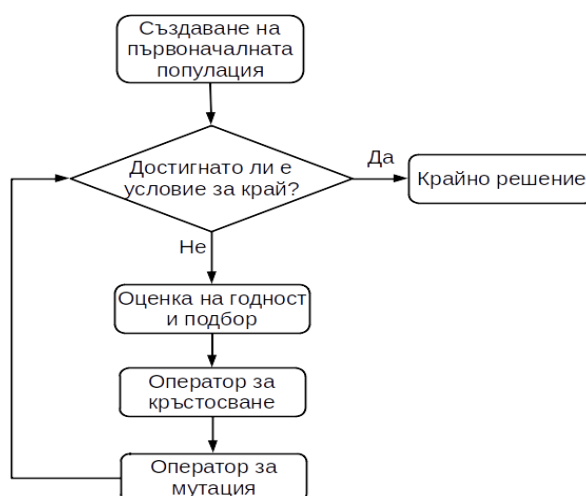
Глава III. Генетичен алгоритъм за ефективно управление на маршрути на транспортно оборудване

Общ преглед на генетичните алгоритми

Генетичните алгоритми (ГА) са алгоритми за търсене, базирани на принципите на естествения подбор и генетиката, въведени от J. Holland и вдъхновени от биологичната еволюция на живите същества. ГА представят множеството от възможни решения на проблема като популация от индивиди и се опитват да открият най-способния индивид, като създават последователно поколения. ГА развива популация от първоначални индивиди до популация от висококачествени, където всеки индивид представлява решение на поставения проблем. Процедурата започва от първоначална популация от произволно генерирани индивиди [Haldurai et al., 2016], като по време на всяко поколение се прилагат последователно три основни генетични оператора към всеки индивид с определени вероятности: селекция, кръстосване и мутация [Goldberg, 1989]. За намиране на добри резултати, ГА трябва да бъде настроен: избор на първоначална популация, функция за оценка, кръстосване и параметри на мутацията.

Генетичен алгоритъм

Предложеният от докторанта генетичен алгоритъм следва стандартната последователност (Фигура 5).



Фигура 5. Последователност на генетичен алгоритъм

Представяне на хромозомата и кодиране на информация – използва се двуредова хромозома за представяне на графика на премествания на контейнера: първи ред – последователност от премествания за текущия график, подредени по реда на изпълнение; втори ред – ОПК, присвоени за изпълнение на съответните премествания от първия ред.

Оценка на годността и подбор – операцията по селекция служи за подбор на най-подходящите/най-добрите индивиди в настоящата популация, които могат да произведат желан резултат. Стойностите за годност се използват за преценка кои индивиди са най-добри за избор. Съществуват различни методи за селекция на най-добрите хромозоми – избор на колело на рулетка, селекция на Болцман, избор на турнир, класиране, селекция в стационарно състояние, селекция на елитарност и други [Haldurai et al., 2016]. В предложението от докторанта алгоритъм за индивидуален подбор на най-подходящите/най-добрите индивиди е използван алгоритъмът за избор на колело на рулетка [Gen and Cheng, 1996], описан в [Haupt and Haupt, 2004; Sivanandam and Deepa, 2008]. Изчисляването на цената за всеки индивид в популацията е описано в Глава II, като целта е тя да бъде минимизирана. За изчисляване на годността за подбор се използва експоненциално пресмятане на цената.

Първоначална популация – възможно е значително да повлияе на изхода на алгоритъма, поради което се поддържат различни техники за първично генериране на популация: валидно със зависимости между преместванията; резултатът на алчния алгоритъм, описан в Глава II [Monov et al., 2022a]; комбинация от двете техники.

Случайна първоначална популация със зависимости – методът генерира случайна популация и гарантира, че се взимат предвид зависимостите между преместванията и всички хромозоми са валидни.

Първоначална популация с помощта на алчен алгоритъм, описан в Глава II – използва резултата от описания в Глава II алгоритъм и го изменя за създаване на първоначална популация.

Първоначална популация, използваща алчния алгоритъм, описан в Глава II, и случайни индивиди – идеята е да се генерира популация, базирана на резултата от евристичния алгоритъм (първоначално „добър“ индивид), но и в същото време да добави разнообразие в нея и да накара генетичния алгоритъм да изследва по-голямо поле за търсене; алгоритъмът генерира част от популацията (например половината) чрез извикване на евристичния метод за инициализация, а другата част – чрез използване на метода за произволна инициализация.

Изследване на ефекта на първоначална инициализация – чрез тестове се установява ефектът на различните методи на инициализация съответно с 23, 42 и 107 задачи за разпределение. Тестовите са изпълнени по пет пъти, за да се отчете ефектът на случайната инициализация (Таблица 3) – получените резултати доказват, че при увеличаване броя на задачите, инициализацията с данни от нашия алчен алгоритъм подобрява значително ефективността на генетичния ни алгоритъм. „Силна“ първоначална популация подобрява качеството на областите, в които генетичния алгоритъм търси решения.

Таблица 3. Сравнение цена на времето и празно движение за тест с 107 задачи

107 задачи						
Тест	Цена на времето			Празно движение		
	Случайна	Алчна	Подобрение	Случайна	Алчна	Подобрение
1	40721,00	22290,75	45%	1:37:03	0:51:22	47%
2	44525,00	25953,75	42%	1:42:09	1:06:55	34%
3	42406,75	22050,75	48%	1:38:12	0:55:14	44%
4	43091,00	22854,25	47%	1:32:16	0:51:19	44%
5	45740,75	21835,00	52%	2:00:23	0:49:44	59%
Средно			47%			46%

Кръстосване – реализираният от нас алгоритъм отчита зависимостите при изграждане на нови хромозоми. Той генерира две деца от две родителски хромозоми.

Мутация – извършва се, като се вземат предвид зависимостите на преместванията. Избира се произволен ген и се поставя на произволно място, което не нарушава зависимостите. Предложеният от нас алгоритъм работи в 7 стъпки.

Край на работа – ГА приключва своята работа, когато не е имало подобрение в популацията за определен брой итерации или е достигнат максимален лимит на поколенията, като са използвани различни условия.

Генериране на график от хромозома – при всяко генериране на нова хромозома (чрез генетични оператори като кръстосване и мутация) трябва да се произведе реален график на задачите, отчитайки всички зависимости между преместванията за правилно изчисляване цената за хромозомата; осъществено е с подходящ алгоритъм.

Резултати и изводи

Представена е авторска реализация на генетичен алгоритъм, използващ специализирани генетични оператори за инициализация, оценка на годност и подбор, кръстосване и мутации, отчитайки спецификата на проблема. Описаният алгоритъм е тестван спрямо реализирания от автора алчен алгоритъм (Глава II [Monov et al., 2022a]) за два вида оборудване за обработка на контейнери: хоризонтален транспорт (като платформи за носене) и портални кранове, работещи в блоковете за съхранение. Таблица 4 отчита подобряването на общите разходи и времената за пътуване без товар за хоризонтален транспорт.

Таблица 4. Подобрение на общите разходи и време за пътуване без товар за хоризонтален транспорт

Брой премествания	Брой ОПК единици	Подобрение на цена на време за придвижване	Подобрение на време за пътуване без контейнер
100	1	0,00 %	1,15 %
100	2	0,01 %	7,39 %
100	4	0,00 %	3,29 %
100	1	0,00 %	3,77 %
100	2	0,00 %	8,50 %
100	4	0,01 %	13,64 %

100	1	0,01 %	0,99 %
100	2	0,00 %	0,29 %
100	4	0,00 %	0,01 %
1000	20	0,00 %	-3,09 %
1000	40	0,00 %	-9,59 %
1330	21	0,00 %	-19,21 %
1001	20	-0,49 %	-89,97 %
Средно подобрене до 4 ОПК:		0,00 %	4,34 %
Средно подобрене над 20 ОПК:		-0,12 %	-30,46 %

Описаният от нас генетичен алгоритъм не успява да подобри навременните разходи за хоризонтално транспортно оборудване, но постига средно 4,34% подобрене на времето за пътуване без контейнер за до 4 (четири) единици оборудване. При планиране на повече от 20 единици алгоритъмът се представя по-лошо от разработения от нас евристичен алчен алгоритъм [Monov et al., 2022a] (-30,46%).

Таблица 5 отчита подобряването на общите разходи и времената за пътуване без товар за крановете, работящи в блоковете за съхранение.

Таблица 5: Подобрене на общите разходи и време за пътуване без товар за кранове в блоковете за съхранение

Брой премествания	Брой ОПК единици	Подобрене на цена на време за придвижване	Подобрене на време за пътуване без контейнер
42	2	12,17%	23,95%
107	2	6,93%	14,45%
23	2	9,79%	-1,96%
Средно подобрене:		9,63%	12,14%

При кранове за подреждане нашият генетичен алгоритъм постига значително подобрене спрямо разработеният от нас алчен евристичен алгоритъм, като постига средно подобрене от 9,63% за цената на времето за придвижване и 12,14% за общото време за пътуване без контейнер.

Глава IV. Табу търсене за ефективно управление на маршрути на транспортно оборудване

Локална оптимизация

Локалният оптимум е екстремумът (минимум или максимум) на целевата функция за даден регион от входното пространство. Тази функция може да има много локални оптимуми или един, като в този случай локалният оптимум е и глобалният оптимум. Алгоритъм за локална оптимизация – предназначен да намира локален оптимум; подходящ е за претърсване на даден регион от пространството за търсене и приближаване до или намиране точно на екстремумите на функцията в този регион; работят обикновено върху едно решение кандидат и включват итеративно правене на малки промени в кандидат-решението и оценяване на промяната, за да се види дали

води до подобрение, и се приема като ново кандидат-решение. Примери за такива алгоритми: Алгоритъм на Бойден-Фелтчер-Голдфарб-Шанно (BFGS Algorithm); Алгоритъм за катерене по хълм (Hill-Climbing Algorithm); Табу търсене (TS).

Глобална оптимизация

Глобалният оптимум е екстремумът (минимум или максимум) на целевата функция за цялото входно пространство за търсене. Целевата функция винаги има глобален оптимум, въпреки че може да има и локални. Наличието на локален оптимум е основен компонент за трудността на глобалния оптимизационен проблем – може да се локализира сравнително лесно локален и относително трудно глобалният оптимум. Глобалният алгоритъм за оптимизация е подходящ за преминаване през цялото входно пространство за търсене и приближаване до (или точно намиране) екстремумите на функцията; може да включва управление на едно или популация от кандидат-решения, от които итеративно се генерират и оценяват нови кандидат-решения (водят ли до подобрение), и се приемат като ново работно състояние. Примери: Генетичен Алгоритъм (Genetic Algorithm); Симулирано закаляване (Simulated Annealing); Оптимизация на рояка на частици (Particle Swarm Optimization).

Локална или глобална оптимизация

Алгоритмите за локално и глобално търсене решават различни проблеми или отговарят на различни въпроси. Алгоритъм за локална оптимизация трябва да се използва, когато се знае, че се намирате в областта на глобалния оптимум или че вашата целева функция съдържа единичен оптимум. Алгоритъм за глобална оптимизация трябва да се използва, когато се знае много малко за стойностите на целевата функция или когато се знае, че функцията съдържа локални оптимуми.

Алгоритмите за глобално търсене често са груби в навигацията си в пространството за търсене; могат да открият долина за добър локален или глобален оптимум, но може да не са в състояние да намерят най-доброто решение в рамките на долината.

Табу търсене

Табу търсене (Tabu Search, TS) е метаевристичен метод, използващ методи за локално търсене, създаден от Фред Глоувър [Glover, 1986] и формализиран през 1989 г. [Glover, 1989; Glover, 1990]; много популярен алгоритъм за решаване на оптимизационни задачи. Алгоритмите за локално търсене взимат потенциално решение и проверяват най-близките съседи (решения, които са подобни, с изключение на много малки разлики) с надеждата да намерят по-добро решение. Тези алгоритми имат тенденция да „заседнат“ в неоптимални региони, където решенията са почти еднакви (имат еднаква или много близка целева функция).

Табу търсенето подобрява стандартното локално търсене, като релаксира основното правило: На всяка стъпка по-лоши кандидати може да бъдат използвани, само ако не могат да се намерят по-добри; Използват се „забрани“ („tabu“), за да не позволяват търсенето да се връща при вече посетени решения. Условието за край може да е общ брой итерации, брой итерации без подобрение и др. Много често Табу търсенето се ползва за подобряване на резултатите от други алгоритми (алчни), както

и алгоритми за глобално търсене (Генетичен). В други случаи Табу търсене се интегрира (в нов хибриден алгоритъм) в други алгоритми за подобряване на взетите решения по време на изпълнението им (за подобряване на новите кандидат индивиди след мутация или подобряване на прото-децата, генерирани от оператора за кръстосване в Генетичните алгоритми).

Табу търсене за планиране и разпределение на задачи, свързани с движението на контейнерите

[Chen et al., 2007] представят интегриран модел за планиране на оборудването; целта е свеждане до минимум разходът или времето, необходимо за обслужване на даден набор от кораби – за решаване на проблема се предлага Табу търсене; анализира се производителността от изчислителна гледна точка. [Ling and Di, 2010] изследват междукрановата интерференция между съседни кранове, която води до блокиране движението на кран от съседни кранове; целта е да се сведе до минимум сумата от времената за завършване на задачите; предлагат нов хибриден оптимизационен алгоритъм, комбиниращ техниките на генетичния алгоритъм и табу търсене (GA-TS).

[Preston and Kozan, 2001] проектират модел за планиране, отчитайки фактори като оборудване за обработка на контейнери, трудови ресурси, капацитет за съхранение и разположение на терминала. Табу търсене се използва за сравняване на резултатите от пристанището в Бризбейн с Генетичен алгоритъм (GA). И други автори разглеждат проблема от различни гледни точки ([Zhou et al., 2020; Belle et al., 2013; Li and Tang, 2009]), прилагайки Табу търсене на различни етапи.

Авторски алгоритъм за табу търсене за подобряване на резултатите от алгоритмите, разгледани в Глава II и Глава III

Авторът предлага Табу търсене за дооптимизиране и подобряване на резултатите от алгоритмите, описани в Глава II и Глава III, за решаване на оптимизационния проблем за планиране и разпределение на задачи, свързани с движението на контейнери в пристанища на контейнерни терминали. Алчният алгоритъм, разгледан в Глава II [Monov et al. 2022a], предлага евристика, която успява да постигне много добри резултати, и е идеален кандидат за начално решение за инициализация на Табу търсене. Генетичният алгоритъм, разгледан в Глава III [Monov et al., 2022b], е алгоритъм за глобално търсене и като такъв успява да намери решение, много близко и най-вероятно намиращо се в локала на най-доброто решение на проблема. Ако резултатът от ГА се ползва за начално решение за инициализация, Табу търсенето трябва да успее да „претърси“ близките решения и да се доближи още повече (или намери точно) до най-доброто решение на проблема.

Представяне на решенията – като хромозома, използвана в Генетичния алгоритъм, описана в Глава III [Monov et al., 2022b].

Начално решение – използват се три вида: *случайно генерирано начално решение* (генерира се по алгоритъма, използван в Глава III за генериране на случайна първоначална популация със зависимости); *инициализира се от резултата на Алчния алгоритъм* [Monov et al., 2022a], описан в Глава II; *инициализира се от резултата на Генетичен алгоритъм* [Monov et al., 2022b], описан в Глава III.

Генериране на съседни решения – използват се генетичните оператори, разработени в Глава III [Monov et al., 2022b]; това автоматично прави нашия алгоритъм съвместим с всичките свойства, поддържани от генетичния алгоритъм – поддръжка на зависимости между преместванията; за намиране на съседни решения се използва генетичния оператор за мутации.

Табу списък – съставен е от табу решения (хромозоми), които вече са избирани като най-добри от съседите; има лимит на големина – всяко ново решение се добавя в края на списъка и ако списъкът се напълни (стигне лимита на големина), най-старото решение се премахва от списъка.

Резултати и изводи

Описаният алгоритъм е тестван, като за първоначално решение се ползват решенията от алчния и генетичния алгоритъм [Monov et al., 2022a; Monov et al., 2022b], и резултатите са сравнени с тези алгоритми. Тествани са сценарии за два вида оборудване: хоризонтален транспорт и кранове на подреждане в блоковете за съхранение. В Таблица 6 са представени резултатите за общи разходи на Табу търсене със случайно първоначално решение в сравнение със същите резултати от Генетичен алгоритъм.

Таблица 6. Сравнение на общи разходи между Генетичен Алгоритъм и Табу алгоритъм със случайно първоначално решение

Брой премествания	Брой ОПК единици	Генетичен алгоритъм	Табу алгоритъм със случайно първоначално решение	Подобрение %
100	1	757832,25	844528.50	-11,44%
100	2	806803	804658.75	0,27%
100	4	732615,75	774290.25	-5,69%
1000	20	1465976,5	7933832.50	-441,20%
1000	40	345075	6125428.00	-1675,10%
100	1	776876,25	810281.50	-4,30%
100	2	709919,75	838414.25	-18,10%
100	4	450952,75	770704.00	-70,91%
1330	21	2458403,25	3917569.75	-59,35%
100	1	848332,75	890424.00	-4,96%
100	2	870027,5	921938.50	-5,97%
100	4	852692,25	895250.50	-4,99%
1001	20	12010542,25	14763388.50	-22,92%
42	2	27086,75	26609.50	1,76%
107	2	38331	47407.25	-23,68%
23	2	13995,75	13801.75	1,39%
Средно подобрение:				-146,57%

Таблица 7 отчита подобряването на общите разходи и времето за пътуване без превоз на контейнер (hh:mm:ss) за хоризонтален транспорт и кранове, работещи в блоковете за съхранение, чрез използването на Табу търсене с първоначално решение резултата от Алчен алгоритъм [Monov et al., 2022a].

Таблица 7. Подобрене на общите разходи и времето за пътуване без превоз чрез Табу търсене и Алчен алгоритъм за кранове, работещи в блоковете за съхранение

Брой премествания	Брой ОПК единици	Общи разходи			Време за пътуване без превоз		
		Алчен алгоритъм	Табу + алчен алгоритъм	Подобрение %	Алчен алгоритъм	Табу + алчен алгоритъм	Подобрение %
42	2	27150.25	26363.00	2,90%	1:51:34	1:37:41	12,44%
107	2	44287.00	39723.00	10,31%	1:54:47	1:36:44	15,73%
23	2	13990.50	13286.00	5,04%	0:41:49	0:40:45	2,55%
Средно подобрене:				6,08%			10,24%

Тестовите на представения алгоритъм показват, че той успява да подобри постигнатите резултати от алгоритмите, описани в Глава II и Глава III. Това показва, че има голям потенциал в използването на Табу търсене в планиране и разпределение на задачи, свързани с движението на контейнерите в пристанищата на контейнерни терминали. Алгоритъмът е успешно интегриран в системата за планиране на оборудване за обработка на контейнери в порт за контейнерен терминал.

Глава V. Симулатор на операциите в пристанище

Ролята на симулатора на операции в пристанище е да изпълнява алгоритмите, когато са налични преместванията на контейнерите, както и да симулира изпълнението на задачите от оборудването във времето. Симулаторът дава възможност да се извежда подробна информация за изпълнението на операциите, както и генериране на статистика.

Симулаторът изпълнява сценарии, дефинирани в JSON входен файл – дефинира един или повече сценария, които се изпълняват последователно. Всеки сценарий съдържа определени параметри: **name** – име на сценария; **type** – вид на сценария: с премествания в блок за съхранение (BLOCK); с премествания на транспортните средства (TRANSPORT); **costParameters** – параметри на функциите за изчисляване на цена на времето и празно движение; **travelDistances** – дефиниция на разстоянията между блоковете, стековете и редовете; **equipmentSpeed** – дефиниране на скоростта на работа на оборудването; **equipment** – списък с оборудване; **movements** – списък с преместванията (задачите) за изпълнение.

Симулаторът е написан на Java, като се изпълнява със `simulate_jobs` команда от конзола с определени аргументи.

Симулаторът работи на итерации във времето, като при всяка итерация се изпълнява алгоритъмът за планиране за текущите налични премествания (задачи), симулира се работата на оборудването. Итерациите може да са базирани на фиксирано време или на събития. Симулаторът изпълнява подадените сценарии и извежда статистика на изпълнението (Фигура 6) в csv формат, с възможност за експорт към електронна таблица; поддържа и показване на подробна информация за

действията при изпълнение на сценарий с итерации, базирани на събития (Фигура 7). Описани са стъпките при изпълнение на симулацията. Представена е JSON схема на входния файл със сценарии. Даден е пример на JSON със сценарий с две премествания.

```
File Edit View Search Terminal Help
$ ./simulate_jobs -e tests/scenarios/scenarios.json
Име на сценарий,Генетичен обща цена,Генетичен време за пътуване без контейнер,Табу + генетичен обща цена,Табу + генетичен време за пътуване без
контейнер,Алчен обща цена,Алчен време за пътуване без контейнер,Табу + алчен обща цена,Табу + алчен време за пътуване без контейнер
Scenario Transport,12391694.75,15:51:59,12334274.0,20:30:36,12576563.25,21:34:13,12306266.25,09:00:17
Scenario_ASC_Small,102112.0,00:12:23,102112.0,00:12:23,103721.0,00:24:45,102119.5,00:12:20
Scenario_ASC_1,27216.25,01:54:03,27216.25,01:54:03,27262.75,01:52:02,26181.75,01:41:00
Scenario_ASC_2,43730.0,01:38:14,42858.75,01:36:25,44188.25,01:43:30,41747.25,01:33:45
Scenario_ASC_3,13142.0,00:40:37,13142.0,00:40:37,14083.5,00:45:42,14083.5,00:45:42
$
```

Фигура 6. Статистика при изпълнение на няколко сценария

```
File Edit View Search Terminal Help
$ ./simulate_jobs -e -l tests/scenarios/small_scenario.json
Изпълнение на сценарий Scenario_ASC_Small

-----Изпълнение на генетичен алгоритъм -----
Итерация 1 в 01:00:00
ASC-1 започва изпълнение на преместване M-4 в 01:00:00 от позиция 1-8-1 и се движи към 1-8-3-2 за преместване на контейнер ABC1111114
ASC-2 започва изпълнение на преместване M-2 в 01:00:00 от позиция 1-118-5 и се движи към 1-61-1-1 за преместване на контейнер ABC1111112
Итерация 2 в 01:00:06
ASC-1 стигна до контейнер ABC1111114 в 01:00:06 на позиция 1-8-3-2 и продължава изпълнението на преместване M-4 към позиция за доставяне 1-36-8-2
Итерация 3 в 01:02:31
ASC-1 завърши работа по преместване M-4 в 01:02:31 на позиция 1-36-8-2
ASC-1 започва изпълнение на преместване M-1 в 01:02:31 от позиция 1-36-8-2 и се движи към 1-41-4-2 за преместване на контейнер ABC1111111
Итерация 4 в 01:03:06
ASC-1 стигна до контейнер ABC1111111 в 01:03:06 на позиция 1-41-4-2 и продължава изпълнението на преместване M-1 към позиция за доставяне 1-117-2-1
Итерация 5 в 01:04:38
ASC-2 стигна до контейнер ABC1111112 в 01:04:38 на позиция 1-61-1-1 и продължава изпълнението на преместване M-2 към позиция за доставяне 1-117-3-1
Итерация 6 в 01:09:05
ASC-2 завърши работа по преместване M-2 в 01:09:05 на позиция 1-117-3-1
Итерация 7 в 01:09:06
ASC-1 завърши работа по преместване M-1 в 01:09:06 на позиция 1-117-2-1
Итерация 8 в 01:30:00
ASC-1 започва изпълнение на преместване M-5 в 01:30:00 от позиция 1-117-2-1 и се движи към 1-30-3-1 за преместване на контейнер ABC1111115
Итерация 9 в 01:36:49
```

Фигура 7. Пример за изпълнение на сценарий с итерации, базирани на събития

Резултати и изводи

Разработеният симулатор симулира операциите в контейнерни пристанища, като целта му е да се доближи възможно най-много до реалните операции, за да генерира достоверни резултати. Използва предефинирани сценарии, които са базирани на истински данни. Симулаторът е използван за тестване на резултатите от алгоритмите, описани в Глави II, III и IV.

Заклучение

От направеното проучване в дисертационния труд са постигнати следните резултати:

- Анализирани са съвременни подходи и решения за планиране и разпределение на задачи.
- Анализирани са различни алгоритми (точни (branch-and-bound) и евристични – алчни, генетични, Табу търсене и др.), използвани за намиране на ефективни решения за планиране и разпределение на задачи в контейнерни пристанища.
- Направен е сравнителен анализ на съществуващите решения и алгоритми за планиране и разпределение на задачи.

В резултат на направените изследвания са дефинирани целите на настоящия дисертационен труд, както и задачите, които трябва да се реализират за тяхното постигане.

Проектирани и разработени са три нови евристични алгоритми, които:

1. подобряват ефективността на вече съществуващите решения;
2. взимат предвид ограничения, като зависимости между задачите;
3. предлагат нов модел за изчисляване на цената на отделните задачи, отчитайки параметри като навременно свършване на работа и ненатоварено движение на оборудването;
4. предлагат възможност за интеграция помежду си, с което да подобрят допълнително своята ефективност.

Разработен е симулатор на операциите в пристанище, чиято роля е да изпълнява алгоритмите в условия, близки до реалните – когато са налични преместванията на контейнерите, както и да симулира изпълнението на задачите от оборудването във времето.

Перспективи

1. Интегриране на Табу търсенето в операциите на Генетичния алгоритъм.
2. Проектиране и разработка на комбиниран, хибриден алгоритъм (генетичен + алчен + табу търсене).
3. Проучване и реализация на алгоритми за машинно обучение за решаване на оптимизационни проблеми, свързани с движението на оборудването в контейнерните пристанища.

Основни приноси на дисертационния труд

Основните приноси в дисертационния труд са:

1. Извършен е анализ на операциите в пристанищата и съществуващи разработки на алгоритми за планиране и разпределение на задачи свързани с движението на контейнерите в контейнерни терминали.
2. Разработени са нови, подобрени алгоритми, които, за разлика от съществуващите, отчитат динамиката на процесите във времето и зависимостите между задачите и оценяват задачите съобразно реални практики за подобряване на икономическата ефективност.
3. Разработен е симулатор за тестване на новите алгоритми в реални условия, който симулира операциите в пристанищата
4. Алгоритмите са тествани и резултатите са сравнени със съществуващи решения.

Връзките между приносите, целите, задачите, мястото на описание в дисертационния труд и направените публикации, са следните:

Принос	Задачи	Глава	Публикации
1	1	I	
2	2, 3	II, III, IV	1, 2
3	4	V	
4	5	II, III, IV	1, 2

Като резултат от представеното изследване е изпълнена основната цел на дисертационния труд – да се разработят и реализират нови, по-ефективни от съществуващите вече алгоритми за планиране и разпределение на задачи, свързани с движението на контейнерите в контейнерни пристанищни терминали, както и симулатор за тяхното реалистично тестване.

Публикации по дисертационния труд

1. **Monov, S., Kostadinov, V., & Pavlov, N.** (2022), Routing Transportation Equipment Using Heuristic Algorithms, International Journal of Differential Equations and Applications, (2022), Vol. 21, No. 1, 77–84, ISSN (Print): 1311-2872; ISSN (Online): 1314-6084, doi: 10.12732/ijdea.v21i1.6, (SJR 2021:0.214). <https://www.ijpam.eu/en/index.php/ijdea/article/view/5987/272>
2. **Monov, S., Pavlov, N., & Rahnev, A.** (2022), Improving Efficiency of Routing Transportation Equipment using Genetic Algorithms, International Journal of Differential Equations and Applications, (2022), Vol. 21, No. 1, pp. 105-116, ISSN (Print): 1311-2872; ISSN (Online): 1314-6084, doi: 10.12732/ijdea.v21i1.9, (SJR 2021:0.214). <https://www.ijpam.eu/en/index.php/ijdea/article/view/5994/278>

Апробация

Резултати, получени в изследването, са използвани в следните международни, национални и университетски проекти:

- **ФП19-ФМИ-002** (2019-2020) „Иновационни ИКТ за дигитално научноизследователско пространство по математика, информатика и педагогика на обучението“ към фонд „Научни изследвания“ при ПУ - участник.
- **ФП21-ФМИ-002** (2021-2022) „Интелигентни иновационни ИКТ в научни изследвания по математика, информатика и педагогика на обучението“ към фонд „Научни изследвания“ при ПУ - участник.

Библиография

- [**Abdel-Fattah et al., 2013**] Abdel-Fattah, A. K., ElTawil, A. B., & Harraz, N. A. (2013, September). An integrated operational research and system dynamics approach for planning decisions in container terminals. In Proceedings of the International Conference on Industrial Engineering Design and Analysis. <https://doi.org/10.5281/zenodo.1088378>
- [**Ayachi et al., 2013**] Ayachi, I., Kammarti, R., Ksouri, M., & Borne, P. (2013). A Genetic algorithm to solve the container storage space allocation problem. arXiv preprint arXiv:1303.1051. <https://arxiv.org/abs/1303.1051>
- [**Bacalhau et al., 2021**] Bacalhau, E. T., Casacio, L., & de Azevedo, A. T. (2021). New hybrid genetic algorithms to solve dynamic berth allocation problem. Expert Systems with Applications, 167, 114198. <https://doi.org/10.1016/j.eswa.2020.114198>
- [**Bazzazi et al., 2009**] Bazzazi, M., Safaei, N., & Javadian, N. (2009). A genetic algorithm to solve the storage space allocation problem in a container terminal. Computers & Industrial Engineering, 56(1), 44-52. <https://doi.org/10.1016/j.cie.2008.03.012>

- [Belle et al., 2013] Van Belle, J., Valckenaers, P., Berghe, G. V., & Cattrysse, D. (2013). A tabu search approach to the truck scheduling problem with multiple docks and time windows. *Computers & Industrial Engineering*, 66(4), 818-826. <https://doi.org/10.1016/j.cie.2013.09.024>
- [Bierwirth and Meisel. 2010] Bierwirth, C., & Meisel, F. (2010). A survey of berth allocation and quay crane scheduling problems in container terminals. *European Journal of Operational Research*, 202(3), 615-627. <https://doi.org/10.1016/j.ejor.2009.05.031>
- [Bierwirth and Meisel. 2015] Bierwirth, C., & Meisel, F. (2015). A follow-up survey of berth allocation and quay crane scheduling problems in container terminals. *European Journal of Operational Research*, 244(3), 675-689. <https://doi.org/10.1016/j.ejor.2014.12.030>
- [Bish et al., 2001] Bish, E. K., Leong, T. Y., Li, C. L., Ng, J. W., & Simchi-Levi, D. (2001). Analysis of a new vehicle scheduling and location problem. *Naval research logistics (NRL)*, 48(5), 363-385. <https://doi.org/10.1002/nav.1024>
- [Cao et al., 2010] Cao, J., Shi, Q., & Lee, D. H. (2010). Integrated quay crane and yard truck schedule problem in container terminals. *Tsinghua Science & Technology*, 15(4), 467-474. [https://doi.org/10.1016/S1007-0214\(10\)70089-4](https://doi.org/10.1016/S1007-0214(10)70089-4)
- [Carlo et al., 2014a] Carlo, H. J., Vis, I. F., & Roodbergen, K. J. (2014). Storage yard operations in container terminals: Literature overview, trends, and research directions. *European journal of operational research*, 235(2), 412-430. <https://doi.org/10.1016/j.ejor.2013.10.054>
- [Carlo et al., 2014b] Carlo, H. J., Vis, I. F., & Roodbergen, K. J. (2014). Transport operations in container terminals: Literature overview, trends, research directions and classification scheme. *European journal of operational research*, 236(1), 1-13. <https://doi.org/10.1016/j.ejor.2013.11.023>
- [Chargui et al., 2021] Chargui, K., Zouadi, T., El Fallahi, A., Reghioui, M., & Aouam, T. (2021). Berth and quay crane allocation and scheduling with worker performance variability and yard truck deployment in container terminals. *Transportation Research Part E: Logistics and Transportation Review*, 154, 102449. <https://doi.org/10.1016/j.tre.2021.102449>.
- [Chen et al., 2007] Chen, L., Bostel, N., Dejax, P., Cai, J., & Xi, L. (2007). A tabu search algorithm for the integrated scheduling problem of container handling systems in a maritime terminal. *European Journal of Operational Research*, 181(1), 40-58. <https://doi.org/10.1016/j.ejor.2006.06.033>
- [Chen et al., 2020] Chen, J., Du, W., Wang, H., & Guo, D. (2020, March). Research on Integrated Scheduling Optimization of Double-trolley Quay Crane and AGV in Automated Terminal. In *IOP Conference Series: Materials Science and Engineering* (Vol. 790, No. 1, p. 012071). IOP Publishing. <https://doi.org/10.1088/1757-899X/790/1/012071>
- [Cormen et al., 2001] Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2001). *Introduction to algorithms* second edition. MIT Press, 2001, ISBN 9780262032933.
- [Dantzig and Ramser, 1959] Dantzig, G. B., & Ramser, J. H. (1959). The truck dispatching problem. *Management science*, 6(1), 80-91. <https://doi.org/10.1287/mnsc.6.1.80>
- [Date and Nagi, 2016] Date, K., & Nagi, R. (2016). GPU-accelerated Hungarian algorithms for the linear assignment problem. *Parallel Computing*, 57, 52-72. <https://doi.org/10.1016/j.parco.2016.05.012>
- [Dijkstra, 1959] Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1), 269-271. <https://doi.org/10.1007/BF01386390>
- [Edmonds and Karp, 1972] Edmonds, J., & Karp, R. M. (1972). Theoretical improvements in algorithmic efficiency for network flow problems. *Journal of the ACM (JACM)*, 19(2), 248-264. <https://doi.org/10.1145/321694.321699>
- [Egerváry, 1931] Egerváry, E. (1931). *Matrixok kombinatorius tulajdonságairol*. Matematikai és Fizikai Lapok, 38: 16–28, 1931.
- [Expósito-Izquierdo et al., 2011] Expósito-Izquierdo, C., Melián-Batista, B., & Moreno-Vega, J. M. (2011, November). Variable neighbourhood search for the quay crane scheduling problem. In

- 2011 11th International Conference on Intelligent Systems Design and Applications (pp. 463-468). IEEE. <https://doi.org/10.1109/ISDA.2011.6121699>
- [Gharehgozli et al., 2015] Gharehgozli, A. H., Laporte, G., Yu, Y., & De Koster, R. (2015). Scheduling twin yard cranes in a container block. *Transportation Science*, 49(3), 686-705. <https://doi.org/10.1287/trsc.2014.0533>
- [Glover, 1986] Glover, F. (1990). Tabu search: A tutorial. *Interfaces*, 20(4), 74-94. <https://doi.org/10.1287/inte.20.4.74>
- [Glover, 1989] Glover, F. (1989). Tabu search—part I. *ORSA Journal on computing*, 1(3), 190-206. <https://doi.org/10.1287/ijoc.1.3.190>
- [Glover, 1990] Glover, F. (1990). Tabu search—part II. *ORSA Journal on computing*, 2(1), 4-32. <https://doi.org/10.1287/ijoc.2.1.4>
- [Goldberg, 1989] Goldberg, D. (1989). *Genetic Algorithm in Search, Optimization and Machine Learning*, Reading, MA: Addison-Wesley, (1989).
- [Haldurai et al., 2016] Haldurai, L., Madhubala, T., & Rajalakshmi, R. (2016). A study on genetic algorithm and its applications. *International journal of computer sciences and Engineering*, 4(10), 139.
- [Haupt and Haupt, 2004] Haupt, R. L., & Haupt, S. E. (2004). *Practical genetic algorithms*. John Wiley & Sons, 2004.
- [He et al., 2019] He, J., Tan, C., & Zhang, Y. (2019). Yard crane scheduling problem in a container terminal considering risk caused by uncertainty. *Advanced Engineering Informatics*, 39, 14-24. <https://doi.org/10.1016/j.aei.2018.11.004>
- [He et al., 2021] He, J., Wang, Y., Tan, C., & Yu, H. (2021). Modeling berth allocation and quay crane assignment considering QC driver cost and operating efficiency. *Advanced Engineering Informatics*, 47, 101252. <https://doi.org/10.1016/j.aei.2021.101252>.
- [Huffman, 1952] Huffman, D. A. (1952). A method for the construction of minimum-redundancy codes. *Proceedings of the IRE*, 40(9), 1098-1101. <https://doi.org/10.1109/JRPROC.1952.273898>
- [König, D., 1931] König, D. (1931). Grafok es matrixok. matematikai es fizikai lapok. *Matematikai Es Fizikai Lapok*, 38, 116-119, 1931.
- [Kuhn, 1955] Kuhn, H. W. (1955). The Hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2), 83-97. <https://doi.org/10.1002/nav.3800020109>
- [Kuhn, 1956] Kuhn, H. W. (1956). Variants of the Hungarian method for assignment problems. *Naval research logistics quarterly*, 3(4), 253-258. <https://doi.org/10.1002/nav.3800030404>
- [Kumar, 2006] Kumar, A. (2006). A modified method for solving the unbalanced assignment problems. *Applied mathematics and computation*, 176(1), 76-82. <https://doi.org/10.1016/j.amc.2005.09.056>
- [Lee and Kim, 2013] Lee, B. K., & Kim, K. H. (2013). Optimizing the yard layout in container terminals. *OR spectrum*, 35(2), 363-398. <https://doi.org/10.1007/s00291-012-0298-z>
- [Lee et al., 2006] Lee, D. H., Song, L., & Wang, H. (2006). Bilevel programming model and solutions of berth allocation and quay crane scheduling (No. 06-1386).
- [Lee et al., 2010] Lee, D. H., Chen, J. H., & Cao, J. X. (2010). The continuous berth allocation problem: A greedy randomized adaptive search solution. *Transportation Research Part E: Logistics and Transportation Review*, 46(6), 1017-1029. <https://doi.org/10.1016/j.tre.2010.01.009>
- [Lengauer et al., 2017] Lengauer, P., Bitto, V., Mössenböck, H., & Weninger, M. (2017, April). A comprehensive java benchmark study on memory and garbage collection behavior of dacapo, dacapo scala, and specjvm2008. In *Proceedings of the 8th ACM/SPEC on International Conference on Performance Engineering* (pp. 3-14). <https://doi.org/10.1145/3030207.3030211>
- [Li and Tang, 2009] Li, K., & Tang, L. (2009, December). A tabu search algorithm for the integrated truck scheduling and storage allocation in quay. In *2009 Second International Symposium on Computational Intelligence and Design* (Vol. 2, pp. 275-278). IEEE. <https://doi.org/10.1109/ISCID.2009.215>

- [**Li et al., 2009**] Li, W., Wu, Y., Petering, M. E., Goh, M., & De Souza, R. (2009). Discrete time model and algorithms for container yard crane scheduling. *European Journal of Operational Research*, 198(1), 165-172. <https://doi.org/10.1016/j.ejor.2008.08.019>
- [**Li et al., 2018**] Li, H., Wu, M., & Chen, H. (2018, August). Analysis and Optimizations of Java Full Garbage Collection. In *Proceedings of the 9th Asia-Pacific Workshop on Systems* (pp. 1-7). <https://doi.org/10.1145/3265723.3265735>
- [**Ling and Di, 2010**] Ling, M. K., & Di, S. (2010). A scheduling method for cranes in a container yard with inter-crane interference. In *Electronic engineering and computing technology* (pp. 715-725). Springer, Dordrecht. https://doi.org/10.1007/978-90-481-8776-8_61
- [**Liu et al., 2004**] Liu, C. I., Jula, H., Vukadinovic, K., & Ioannou, P. (2004). Automated guided vehicle system for two container yard layouts. *Transportation Research Part C: Emerging Technologies*, 12(5), 349-368. <https://doi.org/10.1016/j.trc.2004.07.014>
- [**Mak and Sun, 2009**] Mak, K. L., & Sun, D. (2009). Scheduling Yard Cranes in a Container Terminal Using a New Genetic Approach. *Engineering Letters*, 17(4).
- [**Meisel and Bierwirth, 2009**] Meisel, F., & Bierwirth, C. (2009). Heuristics for the integration of crane productivity in the berth allocation problem. *Transportation Research Part E: Logistics and Transportation Review*, 45(1), 196-209. <https://doi.org/10.1016/j.tre.2008.03.001>
- [**Monov et al., 2022a**] Monov, S., Kostadinov, V., & Pavlov, N. (2022). Routing Transportation Equipment Using Heuristic Algorithms. *International Journal of Differential Equations and Applications*, (2022), Vol. 21, No. 1, 77–84, ISSN (Print): 1311-2872; ISSN (Online): 1314-6084, doi: 10.12732/ijdea.v21i1.6, (SJR 2020:0.103). <https://www.ijpam.eu/en/index.php/ijdea/article/view/5987/272>
- [**Monov et al., 2022b**] Monov, S., Pavlov, N., & Rahnev, A. (2022). Improving Efficiency of Routing Transportation Equipment using Genetic Algorithms, *International Journal of Differential Equations and Applications*, (2022), Vol. 21, No. 1, pp. 105-116, ISSN (Print): 1311-2872; ISSN (Online): 1314-6084 (In print), doi: 10.12732/ijdea.v21i1.9, (SJR 2020:0.103). <https://www.ijpam.eu/en/index.php/ijdea/article/view/5994/278>
- [**Msakni et al., 2018**] Msakni, M. K., Al-Salem, M., Rabadi, G., Kotachi, M., & Diabat, A. (2018). Quay crane scheduling with vessel stability. *Transportation research procedia*, 30, 60-69. <https://doi.org/10.1016/j.trpro.2018.09.008>
- [**Munapo, 2020**] Munapo, E. (2020). Development of an Accelerating Hungarian Method for Assignment Problems. *Eastern-European Journal of Enterprise Technologies*, 4(4 (106), 6–13. <https://doi.org/10.15587/1729-4061.2020.209172>
- [**Munkres, 1957**] Munkres, J. (1957). Algorithms for the assignment and transportation problems. *Journal of the society for industrial and applied mathematics*, 5(1), 32-38. <https://doi.org/10.1137/0105003>
- [**Ng et al., 2007**] Ng, W. C., Mak, K. L., & Zhang, Y. X. (2007). Scheduling trucks in container terminals using a genetic algorithm. *Engineering Optimization*, 39(1), 33-47. <https://doi.org/10.1080/03052150600917128>
- [**Nguyen and Kim, 2009**] Nguyen, V. D., & Kim, K. H. (2009). A dispatching method for automated lifting vehicles in automated port container terminals. *Computers & Industrial Engineering*, 56(3), 1002-1020. <https://doi.org/10.1016/j.cie.2008.09.009>
- [**Niu et al., 2017**] Niu, B., Zhang, F., Li, L., & Wu, L. (2017). Particle swarm optimization for yard truck scheduling in container terminal with a cooperative strategy. In *Intelligent and Evolutionary Systems* (pp. 333-346). Springer, Cham. https://doi.org/10.1007/978-3-319-49049-6_24
- [**Olteanu et al., 2018**] Olteanu, S., Costescu, D., Ruscă, A., & Oprea, C. (2018, August). A genetic algorithm for solving the quay crane scheduling and allocation problem. In *IOP Conference Series: Materials Science and Engineering* (Vol. 400, No. 4, p. 042045). IOP Publishing. <https://doi.org/10.1088/1757-899X/400/4/042045>

- [Papadimitriou and Steiglitz, 1982] Papadimitriou, C., & Steiglitz, K. (1982). Combinatorial optimization: algorithms and complexity. Prentice Hall, Englewood Cliffs.
- [Petering, 2009] Petering, M. E. (2009). Effect of block width and storage yard layout on marine container terminal performance. *Transportation Research Part E: Logistics and Transportation Review*, 45(4), 591-610. <https://doi.org/10.1016/j.tre.2008.11.004>
- [Preston and Kozan, 2001] Preston, P., & Kozan, E. (2001). A tabu search technique applied to scheduling container transfers. *Transportation planning and technology*, 24(2), 135-153. <https://doi.org/10.1080/03081060108717664>
- [Pufek et al., 2019] Pufek, P., Grgić, H., & Mihaljević, B. (2019, May). Analysis of garbage collection algorithms and memory management in java. In 2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO) (pp. 1677-1682). IEEE. <https://doi.org/10.23919/MIPRO.2019.8756844>
- [Sivanandam and Deepa, 2008] Sivanandam, S. N., & Deepa, S. N. (2008). Introduction to Genetic Algorithms, Springer-Verlag, Berlin, (2008).
- [Tasoglu and Yildiz , 2019] Tasoglu, G., & Yildiz, G. (2019). Simulated annealing based simulation optimization method for solving integrated berth allocation and quay crane scheduling problems. *Simulation Modelling Practice and Theory*, 97, 101948. <https://doi.org/10.1016/j.simpat.2019.101948>
- [Tomizawa, 1971] Tomizawa, N. (1971). On some techniques useful for solution of transportation network problems. *Networks*, 1(2), 173-194. <https://doi.org/10.1002/net.3230010206>.
- [UNCTAD, 2008] United Nations Conference on Trade and Development (UNCTAD), Review of Maritime Transport. United Nations, 2008. https://unctad.org/system/files/official-document/rmt2008_en.pdf
- [UNCTAD, 2021] United Nations Conference on Trade and Development (UNCTAD), Review of Maritime Transport. United Nations, 2021. https://unctad.org/system/files/official-document/rmt2021_en_0.pdf
- [Wang et al., 2015] Wang, Z. X., Chan, F. T., Chung, S. H., & Niu, B. (2015). Minimization of delay and travel time of yard trucks in container terminals using an improved GA with guidance search. *Mathematical Problems in Engineering*, 2015. <https://doi.org/10.1155/2015/710565>
- [Wiese et al., 2009] Wiese, J., Kliwer, N., & Suhl, L. (2009). A survey of container terminal characteristics and equipment types. Paderborn: Decision Support & Operations Research Lab, Paderborn University.
- [Wiese et al., 2011] Wiese, J., Suhl, L., & Kliwer, N. (2011). Planning container terminal layouts considering equipment types and storage block design. In *Handbook of terminal planning* (pp. 219-245). Springer, New York, NY. https://doi.org/10.1007/978-1-4419-8408-1_12
- [Wu et al., 2015] Wu, Y., Li, W., Petering, M. E., Goh, M., & Souza, R. D. (2015). Scheduling multiple yard cranes with crane interference and safety distance requirement. *Transportation Science*, 49(4), 990-1005. <https://doi.org/10.1287/trsc.2015.0641>
- [Zhang et al., 2003] Zhang, C., Liu, J., Wan, Y. W., Murty, K. G., & Linn, R. J. (2003). Storage space allocation in container terminals. *Transportation Research Part B: Methodological*, 37(10), 883-903. [https://doi.org/10.1016/S0191-2615\(02\)00089-9](https://doi.org/10.1016/S0191-2615(02)00089-9)
- [Zhang et al., 2016] Zhang, F., Li, L., Liu, J., & Chu, X. (2016, August). Artificial Bee colony optimization for yard truck scheduling and storage allocation problem. In *International Conference on Intelligent Computing* (pp. 908-917). Springer, Cham. https://doi.org/10.1007/978-3-319-42294-7_81
- [Zhao et al., 2019] Zhao, Q., Ji, S., Guo, D., Du, X., & Wang, H. (2019). Research on cooperative scheduling of automated quayside cranes and automatic guided vehicles in automated container terminal. *Mathematical Problems in Engineering*, 2019. <https://doi.org/10.1155/2019/6574582>

- [Zhou et al., 2020]** Zhou, C., Lee, B. K., & Li, H. (2020). Integrated optimization on yard crane scheduling and vehicle positioning at container yards. *Transportation Research Part E: Logistics and Transportation Review*, 138, 101966. <https://doi.org/10.1016/j.tre.2020.101966>
- [Zhu and Lim, 2006]** Zhu, Y., & Lim, A. (2006). Crane scheduling with non-crossing constraint. *Journal of the Operational Research Society*, 57(12), 1464-1471. <https://doi.org/10.1057/palgrave.jors.2602110>
- [Наков и Добриков, 2015]** Наков. П., & Добриков. П. (2015). Програмиране = ++Алгоритми, Пето издание, ISBN 954-8905-06-X.