International Journal of Industrial Engineering & Production Management (2014)

Dece 2014, Volume 25, Number 3
pp. 273-284

http://IJIEPM.iust.ac.ir/

Solving the Container Relocation Problem Using a Heuristic Algorithm

H. Eskandari*, E. Azari Sangeli

Hamidreza Eskandari, *Assistance professor of Industrial Engineering, Tarbiat Modares University*
Esmaeel Azari Sangeli, *MSc student of Industrial engineering, Tarbiat Modares University*

Keywords

Container terminals,
Heuristic algorithm,
Branch and bound,
Container relocation problem,

ABSTRACT

This study presents a heuristic algorithm to solve the container retrieval problem (CRP), which aims to retrieve all the containers from a given yard according to a predetermined sequence with a minimum number of movements. Regarding the high complexity of the problem, a decomposition approach is employed in which the main problem is decomposed into some sub-problems. The proposed heuristic benefits from the available literature on a simpler version of the problem in which relocations are restricted to be done in the bay belonging to. Experimental results indicate that the proposed heuristic is able to efficiently solve instances of up to 720 containers, which is within the range of real cases. Comparative analysis of results with those reported in recent solution approaches for the CRP shows significant improvement in both the total working time of the crane and the number of movements.

* **Corresponding author**. ***Hamidreza Eskandari***
Email: *eskandari@modares.ac.ir*

نشـــریه بین المللـــی مهنـــدسی صنـــایع و مدیـــــریت تولیـــد

شمـــــاره ۳، جلـــد ۲۵، آذر ۱۳۹۳
صفحـــــه ۲۸۴-۲۷۴

http://IJIEPM.iust.ac.ir/

نشریه بین المللی مهندسی صنایع و مدیریت تولید دانشگاه علم و صنعت ایران

ISSN: 2008-4870

حل مساله جابجایی کانتینرها با الگوریتم ابتکاری

اسماعیل آذری سنگلی، حمید رضا اسکندری

کلمات کلیدی

پایانه‌های کانتینری،
مساله جابجایی،
کانتینرها،
الگوریتم ابتکاری،
روش شاخه و کران،

چکیده:

در این مقاله، الگوریتمی ابتکاری برای حل مساله جابجایی کانتینرها در حالت مجاز بودن جابجایی کانتینر بین دسته‌های مختلف پیشنهاد می‌گردد تا در حداقل زمان کارکرد جرثقیل، بلوکی از کانتینرها با توجه به تقدم موجود بین کانتینرها تخلیه گردد. با توجه به پیچیدگی بالای مساله، برای حل آن از رویکرد شکستن مساله اصلی به زیر مسائل کوچک‌تر استفاده شده است. در حل این زیر مسائل از راهکارهای موجود در ادبیات حالت ساده‌تر مساله، که در آن جابجایی کانتینر صرفا در درون دسته مربوطه مجاز می‌باشد بهره برده‌ایم. با استفاده از رویکرد شکست، نمونه مثال‌هایی تا سقف ۷۲۰ کانتینر به گونه‌ای موثر حل شده‌اند که حاکی از قابل استفاده بودن روش پیشنهادی در موارد عملی می‌باشد. نتایج حاصله از روش پیشنهادی، با جدیدترین مطالعات موجود در ادبیات مقایسه شد که جز اندک مواردی، نتایج حاصله حاکی از حصول جواب‌هایی بهتر در مدت زمانی کمتر در مقایسه با الگوریتم‌های مشابه می‌باشد.

۱. مقدمه

یک پایانه کانتینری محلی است که در آن مسیرهایی از دریا، جاده و ریل با یکدیگر تلاقی نموده تا ارتباطی به منظور تسهیل امر جابجایی بار از طریق کانتینرها فراهم گردد. همان‌طور که در شکل (۱) ملاحظه می‌شود یکی از روش‌های مناسب برای استفاده بهینه و مناسب از فضا، بالاخص در مواردی که با کمبود فضا مواجه هستیم، انبارداری و انباشت کانتینرها به شیوه انباشت[۲] می‌باشد. همان‌طور که در شکل (۱) مشاهده می‌شود یک دسته[۳] از کانتینرها از تعدادی ستون[۴] که هر یک دارای ظرفیت مشخصی از کانتینرها هستند تشکیل شده است و از کنار هم قرار گرفتن تعدادی دسته نیز یک بلوک[۵] تشکیل می‌شود [۱].

معمولا کانتینرهای وارداتی در پایانه‌ها در بلوک‌های مجاور به خشکی و کانتینرهای صادراتی در بلوک‌های مجاور به دریا بر روی یکدیگر انباشت می‌شوند و در جریان آمد و شد کانتینرها و به دلیل طبیعت تصادفی پایانه‌های کانتینری گروهی از آن‌ها در زیر سایر کانتینرها مدفون گشته که برای دسترسی به آن‌ها بایستی کانتینرهای بالاسری به سایر ستون‌ها منتقل شوند تا دسترسی به آن‌ها ممکن شود.

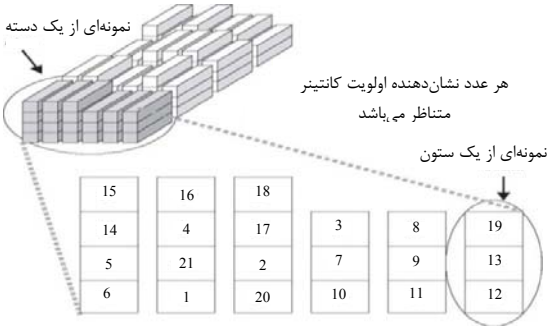
مساله جابجایی کانتینرها[۶] در هر دو بخش مجاور به دریا و خشکی به منظور کاهش تعداد جابجایی‌ها از اهمیت ویژه‌ای برخوردار می‌باشد. البته این مساله مختص پایانه‌ها نمی‌باشد و موارد مشابه دیگری از آن را در صنایع دیگر می‌توان یافت. به عنوان مثال جعبه‌ها، پالت‌ها و صفحات فلزی بزرگ که در

تاریخ وصول: ۹۰/۱۲/۱۴
تاریخ تصویب: ۹۱/۷/۹

اسماعیل آذری سنگلی، دانشجوی کارشناسی ارشد دانشگاه تربیت مدرس- تهران، پل گیشا، دانشگاه تربیت مدرس، دانشکده فنی و مهندسی
*نویسنده مسئول مقاله: **دکتر حمید رضا اسکندری** ، استادیار دانشگاه تربیت مدرس - - تهران، پل گیشا، دانشگاه تربیت مدرس، دانشکده فنی و مهندسی،

[2] Stacking
[3] Bay
[4] Stack
[5] Block
[6] Container Relocation Problem

انبارداری‌ها معمولاً به شیوه انباشت انبارداری می‌شوند با مشکل مشابهی مواجه هستند [۱].



شکل۱: تصویری از یک بلوک به همراه اجزای تشکیل دهنده آن [۱]

در یک تقسیم‌بندی کلی، کارهای انجام شده در این زمینه با توجه به محدودیت جابجایی کانتینر بین دسته‌های مختلف یک بلوک به دو گروه تقسیم می‌شوند. گروه اول متمرکز بر یک دسته بوده و امکان جابجایی کانتینر صرفا در درون همان دسته وجود دارد. طبیعتا اندازه این گروه از مسائل و نمونه‌های حل شده در آن‌ها پایین بوده و حداکثر تا سقف ۱۰۰ کانتینر را شامل می‌شوند. در گروه دوم از مطالعات محدودیت جابجایی بین دسته‌های مختلف برداشته شده و مسائلی با اندازه بزرگ‌تر (در مواردی۷۲۰ کانتینر) را شامل شده و مورد بررسی قرار می‌دهد. در مورد اول با توجه به اینکه زمان جابجایی در موارد مختلف تقریبا یکسان است تابع هدف کمینه‌سازی تعداد جابجایی‌ها می‌باشد، در حالی که در مورد دوم با توجه به زمان‌بر بودن جابجایی بین دسته‌های مختلف و لذا تفاوت قابل توجه بین زمان‌های جابجایی در موارد مختلف، تابع هدف کمینه‌سازی زمان کارکرد جرثقیل می‌باشد. در ادامه به برخی از مهم‌ترین کارهای انجام شده در این زمینه می‌پردازیم.

برای حل حالت ساده، کیم و هونگ [۱] با ارائه روشی مبتنی بر نظریه آمار سعی در برآورد تعداد جابجایی‌های باقی‌مانده برای یک بلوک از کانتینرها نمودند. آن‌ها با استفاده از یک الگوریتم شاخه و کران توانستند زمان رسیدن به جواب را تا حدود زیادی کاهش دهند ولی لزوما تعداد جابجایی‌ها را کاهش ندادند. با این حال بیشترین تعداد کانتینر گنجانده شده در نمونه‌های حل‌شده توسط آن‌ها از ۳۰ کانتینر تجاوز نمی‌کند [۴]. کاسرتا و همکاران [۷] با ارائه نمایشی ماتریسی از بلوک کانتینرها و ارائه یک روش ساده برای حل مساله، آن را به طور موثری حل نمودند. در ادامه کاسرتا و همکاران [۳] روشی پویا برای رسیدن به جواب مساله طراحی نمودند.

آن‌ها با حذف برخی از حالات ممکن برای رسیدن به جواب سعی در حل مسائل با اندازه بزرگ‌تر نمودند و در نهایت نتایج خود را با الگوریتم پیشنهادی کیم و هونگ [۱] مقایسه نمودند که حاکی از بهبود قابل توجهی بوده است. یکی از جدیدترین مطالعات انجام شده در این زمینه توسط فورستر و برتفلدت [۴] می‌باشد. آن‌ها کانتینرها را با توجه به اینکه آیا مسیر کانتینری با تقدم بیشتر از خود را مسدود می‌نمایند یا نه به دو گروه خوب و بد تقسیم نموده و با توجه به آن شش نوع جابجایی را تعریف نمودند. آن‌ها با ارائه الگوریتمی مبتنی بر روش شاخه و کران و با توجه به تقدمی که برای انواع جابجایی‌ها در نظر می‌گرفتند اقدام به جابجایی کانتینرها و جستجوی جواب‌های بهتر می‌نمایند. در نهایت آن‌ها نتایج کار خود را با تمام کارهای انجام شده در این زمینه مقایسه نمودند که حاکی از بهبود نسبی در مقایسه با اکثر روش‌های موجود می‌باشد. در یکی دیگر از جدیدترین مطالعات، کاسرتا و همکاران [۸] مدل ریاضی حالت ساده مساله را ارائه نمودند و همچنین ثابت کردند که این مساله به لحاظ پیچیدگی از نوع NP-HARD می‌باشد.

در هر صورت به دلیل پیچیدگی مدل پیشنهادی آن‌ها، حتی نمونه مثال‌های کوچکی که شامل حداکثر ۲۴ کانتینر بودند نیز قابل حل نبودند. آنلویرت و آیدین [۹] از جمله معدود مطالعاتی هستند که تابع هدف مساله را کاهش زمان کارکرد جرثقیل در نظر گرفته‌اند. آن‌ها تعدادی الگوریتم ابتکاری برای حل مساله ارائه نمودند و کارایی این الگوریتم‌ها را با توجه به جواب بهینه حاصله از یک الگوریتم شاخه کران بررسی نمودند. در هر صورت نمونه مثال‌های بررسی شده در آن مطالعه دارای ابعادی کوچک (حداکثر ۶ ستون و ارتفاع ۷) بوده و صرفا به حل مساله برای حالتی که در آن یک دسته وجود داشته باشد پرداخته‌اند.

در مورد حالت تعمیم‌یافته مساله که در آن جابجایی کانتینر بین دسته‌های مختلف مجاز می‌باشد کارهای اندکی صورت پذیرفته است. لی و چااو [۵] یک الگوریتم جستجوی محلی برای حل مساله مرتب‌سازی کانتینرها[1] ارائه نمودند که در آن ابتدا جوابی اولیه از مساله تولید می‌شود و سپس با استفاده از تکرارهای فراوان در هر مرحله مسیرهای تصادفی زیادی تولید شده و سپس با استفاده از یک مدل ریاضی اقدام به انتخاب بهترین ترکیب شدنی از مسیرهای تولید شده می‌نمودند.

در ادامه لی و لی [۲] با استفاده از این الگوریتم جستجوی محلی اقدام به حل تعمیم یافته مساله جابجایی کانتینرها نمودند و مسائلی تا سقف ۷۲۰ کانتینر را حل نمودند. در این مقاله حالت تعمیم‌یافته مساله بررسی خواهد شد و هدف از این مقاله بررسی الگوریتم‌های ابتکاری موجود در ادبیات حالت ساده مساله به منظور استفاده در حل حالت تعمیم‌یافته آن می‌باشد.

[1] Container pre-marshaling problem

در همین راستا در ادامه تعریف دقیقی از مساله آورده شده است. در بخش سوم روش ابتکاری پیشنهادی ارائه شده و سپس در بخش چهارم نتایج عددی حاصله از الگوریتم و کارایی آن بحث شده است و قسمت آخر شامل نتیجه‌گیری و معرفی زمینه‌های پژوهشی آتی می‌باشد.

۲. مساله جابجایی کانتینرها

در این مساله فرض بر این است که یک بلوک از کانتینرها با چیدمانی مشخص از کانتینرها به همراه یک دنباله از اولویت‌ها که نشان دهنده اولویت میان کانتینرها می‌باشند داده شده است. جواب این مساله به صورت دنباله‌ای از جابجایی‌ها بوده و تابع هدف آن کاهش تعداد این جابجایی‌ها می‌باشد، به طوریکه زمان لازم برای انجام این جابجایی‌ها توسط جرثقیل نیز کاهش یابد.

به طور دقیق‌تر بلوکی از کانتینرها را در نظر بگیرید که از B دسته تشکیل شده و هر دسته نیز به نوبه خود از S ستون تشکیل شده است. هر ستون حداکثر تا سقف H کانتینر ظرفیت دارد. همچنین فرض کنید تعداد N کانتینر $(N\leq BSH)$ در این بلوک جای گرفته‌اند که به G گروه تعلق دارند $(g = 1,\ldots,G;$ $G>1)$ و تفاوت این گروه‌ها در میزان تقدم عددی کانتینرهای درون هر گروه می‌باشد به نحوی که کانتینرهای با تقدم بالاتر در گروهی با شماره کمتر قرار گرفته‌اند. کانتینرها به طور دلخواه درون ستون‌ها توزیع شده‌اند و حرکات ممکن درون هر بلوک از دو نوع تشکیل شده‌اند.

گروه اول، حرکات جابجایی[1] نامیده می‌شوند که در آن کانتینری از یک ستون به ستون دیگری منتقل می‌گردد که لازمه آن این است که کانتینر مذکور در بالای ستون مبدا قرار گرفته و لذا قابل برداشته شدن باشد و ستون دوم حداقل یک ظرفیت خالی با توجه به حداکثر ظرفیت ستون داشته باشد. گروه دوم حرکت‌ها حذفی[2] نامیده می‌شوند که در آن کانتینری از ستون مربوطه برداشته شده و به خارج از بلوک منتقل می‌شود که لازمه آن این است که در آن لحظه کانتینر مذکور دارای بالاترین اولویت در بین کانتینرهای باقی‌مانده باشد و لذا امکان خروج را داشته باشد [۴].

تابع هدف در این مساله کاهش تعداد جابجایی‌ها و همچنین زمان لازم برای انجام این جابجایی‌ها توسط جرثقیل می‌باشد و محدودیت‌های آن عبارتند از:

(۱) اولویت میان کانتینرها بطوریکه در هر لحظه فقط می‌توان کانتینری را که دارای بالاترین اولویت در بین کانتینرهای باقی‌مانده باشد از بلوک حذف نمود. (۲) توجه به کانتینرهای مسدودکننده بطوریکه در هر لحظه فقط امکان دسترسی به کانتینرهایی وجود دارد که در زیر کانتینر دیگری قرار نگرفته باشد. (۳) محدودیت سوم، محدودیت جابجایی کانتینر بین دسته‌های مختلف می‌باشد که در برخی مطالعات آن را لحاظ کرده و برخی دیگر آن را حذف نموده‌اند. در این مطالعه ما به بررسی حالت تعمیم یافته مساله که در آن امکان جابجایی کانتینر بین دسته‌ها مختلف وجود دارد خواهیم پرداخت.

محاسبه زمان کارکرد جرثقیل متناظر با دنباله‌ای معین از جابجایی‌ها، با توجه به فرآیند جابجایی کانتینر صورت می‌پذیرد. اولا در سراسر مقاله فرض بر این است که فقط یک جرثقیل در درون هر بلوک از کانتینرها قرار دارد و لذا تمام جابجایی‌ها توسط همین جرثقیل انجام می‌شود. برای انجام یک جابجایی پس از مشخص شدن کانتینر مورد نظر جهت انتقال، جرثقیل به دسته حاوی آن کانتینر منتقل شده، که به آن حرکت گنتری[3] گوییم و سپس ترولی[4] را به ستون مورد نظر منتقل می‌کند که به این حرکت نیز حرکت ترولی گوییم. در نهایت جرثقیل اسپریدر[5] خود را به کانتینر مورد نظر متصل نموده و آن را بلند می‌کند، که به این حرکت نیز حرکت اسپریدر گویند. پس از اینکه جرثقیل کانتینر مورد نظر را از ستون مربوطه برداشت طی اقدام مشابهی ابتدا با حرکت گنتری کانتینر را به دسته مربوطه و سپس با حرکت ترولی به ستون مربوطه منتقل می‌کند و در نهایت با حرکت اسپریدر آن را بر روی ستون مورد نظر و یا کامیون قرار می‌دهد. زمان لازم برای انجام این دنباله از جابجایی‌ها متاثر از نوع جرثقیل، وزن کانتینرها و موارد دیگری می‌باشد. اما از آنجا که در انتهای این مقاله به دنبال مقایسه نتایج حاصله از الگوریتم پیشنهادی خود با الگوریتم پیشنهادی لی و لی [۲] که شامل جدیدترین نتایج در این حوزه می‌باشد هستیم، شرایط موجود در آن مقاله را معیار اندازه‌گیری زمان لازم برای جابجایی‌ها قرار داده‌ایم که به شرح زیر است:

۱. سرعت گنتری : ۳.۵ ثانیه به ازای هر یک دسته،

۲. سرعت ترولی : ۱.۲ ثانیه به ازای عرض یک کانتینر،

۳. زمان از دست رفته در اثر افزایش و کاهش شتاب در گنتری ۴۰ ثانیه،

۴. زمان لازم برای برداشتن (گذاشتن) کانتینر از (بر) ستون مربوطه توسط اسپریدر : ۳۰ ثانیه.

لی و لی [۲] برای محاسبه زمان جابجایی‌ها توسط جرثقیل فرمول مشخصی ارائه نکرده‌اند و همچنین به هیچ مقاله مشخصی ارجاع نداده است، اما با توجه به توضیحات درون مقاله و تحلیل داده‌های موجود در آن و همچنین با مراجعه به برخی مطالعات مرتبط [۶] به فرمول مشخص (۱) رسیده‌ایم که با توجه به نتایج

[1] Relocation
[2] Remove
[3] Gantry
[4] Trolley
[5] Spreader

مربوط به نمونه مثال حل شده در متن مقاله لی و لی [۲] قابل قبول می‌باشد:

$$\text{زمان کارکرد جرثقیل} = ۱.۲x + ۳.۵y + ۶۰z + ۴۰q \quad (۱)$$

که در آن،

z : تعداد کل حرکات،

q : تعداد کل حرکات بین دسته‌ای،

x : تعداد کل حرکات درون دسته‌ای انجام شده توسط ترولی،

y : تعداد کل حرکات بین دسته‌ای انجام شده توسط گنتری.

۳ . ارائه الگوریتم ابتکاری پیشنهادی

با توجه به این نکته که جواب مساله جابجایی کانتینرها به صورت دنباله‌ای از جابجایی‌ها می‌باشد، رویکرد ما در انتخاب روش حل مبتنی بر این اندیشه است که با داشتن طول دنباله بهینه کافی است به دنبال مشخص کردن اعضای دنباله باشیم، بطوریکه اولا اعمال این دنباله از جابجایی‌ها به بلوک مورد نظر با توجه به اولویت میان کانتینرها شدنی بوده و ثانیا منجر به برداشته شدن تمام کانتینرهای درون آن بلوک گردد. همان‌طور که گفته شد در طراحی الگوریتم پیشنهادی از روش‌های موجود در ادبیات بهره برده‌ایم. به طوریکه مشابه کیم و هونگ [۱]، همچنین لی و لی [۲] برای جستجوی جواب و حذف شاخه‌های زاید از کران پایین تعداد جابجایی‌های باقی‌مانده بهره برده‌ایم.

همچنین در ایجاد شاخه‌های جدید برای اولویت‌بندی بین شاخه-های مجاز با توجه به کران پایین جابجایی‌های باقی‌مانده، از رویکرد پیشنهادی کاسرتا و همکاران [۷] بهره برده‌ایم. در ادامه به برخی روش‌های موجود در ادبیات برای محاسبه کران پایین تعداد جابجایی‌های باقی‌مانده اشاره نموده و پس از بررسی آن‌ها بهترین روش را برای این منظور انتخاب می‌کنیم.

۳-۱. کران پایین تعداد جابجایی‌های باقی‌مانده

در ادبیات روش‌های متفاوتی برای تعیین کران پایین تعداد جابجایی‌های باقی‌مانده در هر مرحله ارائه شده است. از جمله کیم و هونگ [۱] با استفاده از نظریه احتمال روشی برای برآورد تعداد جابجایی‌های باقی‌مانده ارائه نمودند. روش قطعی مورد استفاده در برخی مطالعات به صورت فرمول (۲) می‌باشد [۲]. در این روش ابتدا کانتینرها بر حسب اینکه مسیر کانتینری با تقدم بیشتر از خود را مسدود می‌نمایند یا نه به ترتیب به دو گروه مسدودکننده و غیر مسدودکننده تقسیم می‌شوند. در ادامه با توجه به تقسیم‌بندی مذکور برای کانتینرها، کران پایین تعداد جابجایی‌های لازم برای تخلیه کامل بلوک از کانتینرهای باقی-مانده در آن در هر مرحله از فرمول زیر حاصل می‌شود:

$$\text{کران پایین تعداد جابجایی‌ها} = \text{(تعداد کانتینرهای غیر مسدودکننده)} + ۲ \times \text{(تعداد کانتینرهای مسدودکننده)} \quad (۲)$$

البته فورستر و برتفلدت [۴] نمونه کامل‌تری از این فرمول را ارائه نموده‌اند که در مواردی کران بالای پیشنهادی آن‌ها یک واحد بیشتر از کران پیشنهادی فرمول (۲) می‌باشد. از آنجا که مقدار حاصله از این دو روش تقریبا یکسان می‌باشد و با توجه به سادگی فرمول (۲) و ماهیت الگوریتم مورد استفاده ما، در این مقاله برای تعیین تعداد جابجایی‌های باقی‌مانده از فرمول (۲) استفاده می‌شود.

۳-۲. تشریح الگوریتم پیشنهادی

پارامترهای ورودی این مساله عبارتند از: تعداد دسته‌ها (B)، تعداد ستون‌ها (S) و حداکثر ارتفاع مجاز برای هر ستون (H). از آنجا که تنها مطالعه مشاهده شده در ادبیات مساله در رابطه با حالت تعمیم‌یافته مساله جابجایی کانتینرها توسط لی و لی [۲] صورت گرفته است و لذا به منظور مقایسه نتایج حاصله از الگوریتم پیشنهادی با نتایج حاصله از آن مطالعه هر سه پارامتر ورودی در این مطالعه را مشابه با لی و لی [۲] تعریف می‌نماییم. با توجه به مقاله مذکور تعداد ستون‌های درون هر دسته برابر ۱۶، حداکثر ارتفاع یک ستون ۶ و در مواردی ۸ و در نهایت تعداد دسته‌های درون بلوک را از ۱ تا ۱۰ متغیر می‌گیریم.

به منظور سهولت در بیان الگوریتم تعاریف زیر را داریم: کانتینری که در میان کانتینرهای باقی‌مانده در دسته (یا بلوک) مورد بررسی دارای بالاترین اولویت باشد کانتینر هدف نام دارد. ستون حاوی این کانتینر ستون هدف نام دارد. اگر کانتینری مسیر کانتینرهای با تقدم بیشتر از خود را مسدود کند (نکند) آن را یک کانتینر بد (خوب) می‌نامیم. در صورتی که کانتینر بدی پس از جابجایی بد (خوب) باشد، این جابجایی را بد-بد (بد-خوب) می‌نامیم.

با توجه به NP-Hard بودن مساله جابجایی کانتینرها [۸] با افزایش اندازه مساله تعداد محاسبات لازم برای رسیدن به جواب به طور نمایی افزایش می‌یابد. لذا به نظر می‌رسد که افزایش اندازه مساله (در مواردی ۷۲۰ کانتینر) مهم‌ترین چالش پیش روی حل آن می‌باشد که برای مقابله با آن رویکرد پیشنهادی ما شکستن این مساله به زیر مسئله‌های کوچک می‌باشد. حال از آنجا که یک بلوک از کانتینرها خود از چند دسته مختلف تشکیل شده است و چون با در نظر گرفتن تمام دسته‌ها با اندازه بزرگی از مساله مواجه می‌شویم، یک راه مناسب برای کاهش اندازه مساله شکستن آن به مسائل کوچک با توجه به تعداد دسته‌های درون هر بلوک می‌باشد، بطوریکه برای کانتینرهای درون هر دسته یک مساله جدید در نظر می‌گیریم و با توجه به دنباله تقدم

و تاخر موجود بین کانتینرهای درون آن دسته اقدام به حل مساله جابجایی کانتینرها در درون همان دسته می‌نماییم. سپس جواب مساله اصلی از ترکیب این جواب‌های زیر مسائل فرعی حاصل می‌گردد.

در ادامه ابتدا ساختار اصلی برنامه (شکل۲) و سپس الگوریتم جستجوی ( )Search که برای حل زیرمسائل طراحی شده است (شکل۳) را تشریح خواهیم نمود.

شروع

۱. موارد زیر را برای هر دسته موجود در بلوک کانتینرها انجام دهید:

۱.۱. Length := ∞ ؛ {$N \times H$}

۱.۲. Time-limit := 0 ؛

۱.۳. تا وقتی که Time-limit برابر صفر است موارد زیر را انجام دهید:

۱.۳.۱. با توجه به دسته جاری، دسته جدید را به صورت زیر ایجاد کنید:

۱.۳.۲. کانتینرهای درون دسته جاری را بر اساس شماره آن‌ها به طور صعودی مرتب نمایید؛

۱.۳.۳. تقدم جدید هر کانتینر را برابر با رتبه آن کانتینر در لیست مرتب شده قرار بدهید؛

۱.۳.۴. Solution-status := False ؛

۱.۳.۵. فراخوانی زیربرنامه ( )Search ؛

انتهای حلقه ۱.۳.

انتهای حلقه ۱.

۲. بهترین جواب و زمان کارکرد جرثقیل متناظر با آن را گزارش کن؛

پایان.

شکل۲: بدنه اصلی الگوریتم

همان‌طور که در شکل (۲) ملاحظه می‌شود در بدنه اصلی الگوریتم مجموعه‌ای از دستورات برای هر یک از دسته‌ها اجرا می‌شود. بطوریکه ابتدا طول اولیه دنباله (Length) را برابر با حاصل‌ضرب تعداد کانتینرهای موجود در دسته (N) در حداکثر ارتفاع ستون‌ها (H) قرار می‌دهیم. زیرا این مقدار به وضوح کران بالایی برای تعداد جابجایی‌های بهینه است. همچنین مقدار متغیر Time-limit را برابر با صفر قرار می‌دهیم. غیر صفر بودن این متغیر به معنای پایان زمان جستجو برای یافتن جواب مرتبط با دسته جاری می‌باشد. مادامی که متغیر Time-limit صفر باشد به جستجوی جواب‌های بهتر می‌پردازیم به طوری که ابتدا با توجه به دسته جاری زیر دسته‌ای جدید به صورت زیر ایجاد می‌کنیم.

ابتدا کانتینرهای واقع در دسته را به ترتیب صعودی مرتب می‌کنیم، سپس شماره جدید هر کانتینر برابر با رتبه آن کانتینر در لیست فوق قرار می‌دهیم، بطوریکه کوچک‌ترین شماره با شماره ۱ و بزرگ‌ترین شماره با شماره N (تعداد جاری کانتینرها) جایگزین می‌گردند. سپس مقدار متغیر Solution-status را برابر با False قرار می‌دهیم که به معنای این است که برای طول دنباله جاری هنوز جوابی یافت نشده است.

در ادامه الگوریتم جستجوی ( ) Search را فراخوانی می‌کنیم تا در سقف زمانی مجاز اقدام به جستجوی جواب مناسب نماید. نهایتا پس از اینکه دستورات فوق را برای هر یک از دسته‌ها انجام دادیم جواب نهایی که ترکیب زیرجواب‌های فوق است را به همراه زمان کارکرد جرثقیل متناظر با آن جواب گزارش می‌دهیم. لذا در ادامه الگوریتم جستجوی ( ) Search توضیح داده خواهیم شد.ساختار کلی الگوریتم ( ) Search در شکل (۳) قابل مشاهده است. در این الگوریتم ابتدا تجاوز از سقف زمانی جستجو بررسی می‌گردد و در صورت مثبت بودن الگوریتم به کار خود خاتمه می‌دهد.

در صورتی که شرط بالا محقق نگردد، رسیدن به جوابی شدنی از مساله بررسی می‌شود که در صورت مثبت بودن، این جواب به عنوان بهترین جواب تا این لحظه ذخیره می‌گردد و پس از اصلاح احتمالی طول دنباله، کنترل به بدنه اصلی الگوریتم برمی‌گردد تا جستجو با طول دنباله جدید از سر گرفته شود.

در صورتی که هیچ‌کدام از دو شرط فوق درست نباشند بایستی به ایجاد شاخه‌های جدید در درخت جستجو بپردازیم که در این صورت ممکن است دو حالت متفاوت رخ دهد:

اول آنکه مسیر کانتینر هدف توسط هیچ کانتینر دیگری مسدود نشده و لذا کانتینر هدف برداشته شده و پس از اصلاح وضعیت بلوک، الگوریتم جستجو برای سایر کانتینرهای باقی‌مانده در بلوک تکرار می‌شود. به هر حال در صورتی که این فراخوانی منجر به رسیدن به جواب نگردد با استفاده از دستور ۲.۶ از این شاخه صرف نظر شده و به بررسی سایر شاخه‌ها خواهیم پرداخت.

حالت دیگر بدین صورت است که کانتینر با بالاترین تقدم در زیر سایر کانتینرها مسدود شده است لذا بایستی کانتینرهای مسدودکننده به سایر ستون‌ها منتقل شوند تا این کانتینر آزاد گردد که این امر منجر به ایجاد شاخه‌های جدید در درخت جستجو می‌گردد، برای این منظور کمترین شماره (بالاترین اولویت) را در ستون‌های غیر هدف شناسایی کرده و با توجه به آن‌ها لیست Q را بدین شرح ایجاد می‌کنیم: ابتدا تمامی جابجایی‌های بد-خوب را به ترتیب صعودی بودن بر اساس شماره فوق‌الذکر وارد لیست می‌کنیم. سپس تمامی جابجایی‌های بد-بد را به ترتیب نزولی بودن بر اساس شماره فوق‌الذکر وارد لیست می‌کنیم. لیست ایجاد شده (Q) از نوع FIFO[1] می‌باشد.

در ادامه به ترتیب برای عناصر لیست Q شاخه‌های جستجو را ایجاد کرده و آن‌ها را بررسی می‌کنیم. برای این منظور ابتدا جابجایی مذکور را بر دسته اعمال نموده و در صورتی که تعداد جابجایی‌های صورت گرفته تا این لحظه (Ni) بعلاوه کران پایین

[1] First in first out

تعداد جابجایی‌های باقیمانده برای دسته جاری کمتر از کران بالای تعداد جابجایی‌های مربوط به بهترین جواب تا این لحظه باشد، شاخه جستجوی مربوط به آن را ایجاد کرده و در غیر این صورت از آن صرف نظر می‌کنیم.

زیربرنامه Search () :

شروع
۱. **اگر** سقف زمان جستجو فرارسیده است **آنگاه**:
۱.۱ Time-limit : = 1؛
۱.۲ Solution-status : = True؛
۱.۳ پایان جستجو و بازگشت به برنامه اصلی؛
پایان شرط ۱.
۲. **اگر** به یک جواب شدنی از مساله رسیده‌ایم **آنگاه**:
۲.۱. **اگر** طول جواب حاصله از Length کمتر است **آنگاه**:
۲.۱.۱. Length را برابر با طول این جواب شدنی قرار بده؛
پایان شرط ۲.۱.
۲.۲. جواب جاری را ذخیره کن ؛
۲.۳ Solution-status : = True ؛
درغیراینصورت اگر یک حرکت حذفی وجود دارد **آنگاه**:
۲.۴. حرکت حذفی مذکور را بر دسته جاری اعمال کن؛
۲.۵. فراخوانی زیربرنامه Search () ؛
۲.۶. عکس حرکت حذفی صورت گرفته در بالا را بر دسته اعمال کن؛
درغیراینصورت
۲.۷. لیست Q را به صورت زیر ایجاد کن:
۲.۷.۱. کمترین شماره (بالاترین اولویت) را در ستون‌های غیرهدف شناسایی کن؛
۲.۷.۲. ابتدا جابجایی‌های از نوع بد-خوب را به صورت صعودی (بر اساس شماره‌های حاصله از ۲.۷.۱) وارد لیست کن؛
۲.۷.۳. جابجایی‌های از نوع بد-بد را به صورت نزولی (بر اساس شماره‌های حاصله از ۲.۷.۱) وارد لیست کن؛
۲.۸. برای عناصر لیست Q به صورت FIFO موارد زیر را انجام بده:
۲.۸.۱. **اگر** Solution-status = True **آنگاه**:
۲.۸.۱.۱. بازگشت به بدنه اصلی الگوریتم؛
پایان شرط ۲.۸.۱.
۲.۸.۲. گزینه جابجایی جاری را بر روی دسته اعمال کن؛
۲.۸.۳. **اگر** Ni + (وضعیت جاری دسته) Length > lb **آنگاه**:
۲.۸.۳.۱. فراخوانی زیربرنامه Search () ؛
پایان شرط ۲.۸.۳.
۲.۸.۴. عکس حرکت جابجایی انجام شده در بالا را بر دسته اعمال کن؛
پایان حلقه ۲.۸.
پایان شرط ۲.
پایان.

شکل ۳: نمایی کلی از الگوریتم جستجو

هدف از این کار جلوگیری از انجام حرکات غیر مفید می‌باشد و برای این کار از تابع ( ) Lb که وضعیت جاری دسته را به عنوان ورودی دریافت می‌کند و کران پایین تعداد جابجایی‌های باقی‌مانده را با توجه به دسته مورد نظر و معادله (۱) برمی گرداند، استفاده می‌کنیم. در هر صورت اگر شاخه ایجاد شده منجر به جواب نگردد با دستور ۲.۸.۴ از آن صرف‌نظر شده و شاخه های دیگر بررسی خواهند شد. نحوه اولویت‌بندی میان شاخه‌های جستجو در قالب شکل (۴) ارائه شده است. به هر حال قبل از ایجاد شاخه جدید سقف زمانی جستجو بررسی خواهد شد. این روش اولویت‌بندی با اندکی تفاوت توسط کاسرتا و همکاران [۷] و همچنین توسط فورستر و برتفلدت [۴] نیز مورد استفاده بوده است.

۱ :	۱۹	۵	۵۵	۱۲	۳۲	۱۴	۱۷	۸	۱۳	

۱: کمترین شماره (بیشترین اولویت) در هر یک از ستون‌های ورودی
۲: ترتیب بهترین شاخه‌های جستجو از چپ به راست
کانتینری که باید جابجا گردد : ۱۵

۲ :	۵	۸	۱۲	۱۳	۱۴	۵۵	۳۲	۱۹	۱۷	
			<۱۵			>۱۵				

شکل۴: مثالی از نحوه اولویت‌بندی بین شاخه‌های جدید

به عنوان آخرین نکته، در صورتی که به جوابی شدنی از مساله برسیم یا سقف زمانی فرا رسد مقدار Solution-status از False به True تغییر نموده و مانع از ایجاد شاخه‌های جدید شده و لذا موجب بازگشت سریع به تابع اصلی می‌شود.

						۶	۱۸		۲۶	۲۲							۵۰
۲۷		۵۵				۴	۵۳	۴۷	۴۲	۹		۱۲					۳۹
۵۴	۳۳	۲۴				۳۰	۶۷	۱۶	۵۸	۳۵	۶۴	۷۰					۶۵
۶۳	۶۱	۸	۳۶			۵۶	۵۹	۴۴	۵۱	۲۵	۲۳	۴۰	۵				۴۶
۲	۶۸	۵۲	۱۷	۱		۴۱	۱۴	۱۹	۱۰	۶۹	۶۰	۲۸	۳۴	۱۵	۶۲		۷
۴۸	۳۲	۱۳	۵۷	۴۹		۲۱	۶۶	۳۱	۳۸	۳۷	۴۳	۳	۴۵	۲۹	۲۰		۱۱
۱	۲	۳	۴	۵		۶	۷	۸	۹	۱۰	۱۱	۱۲	۱۳	۱۴	۱۵	۱۶	شماره ستون :

شکل ۵: تصویر دسته‌ای از کانتینرها و اولویت بین آن‌ها

۴. نتایج عددی

الگوریتم پیشنهادی ما توسط کامپایلر Borland C++ V5.02 نوشته شده و سپس بر روی یک کامپیوتر شخصی با مشخصات RAM: 3GB، CPU: Intel(R) Core(TM) i3 اجرا شده است. از آنجا که مقاله لی و لی [۲] جدیدترین مقاله مرتبط در این زمینه می‌باشد، نمونه مسائل حل شده در آن مقاله از طریق لینک موجود در آن مقاله تهیه شد تا با استفاده از آن‌ها نتایج حاصله از دو تحقیق امکان مقایسه‌پذیری داشته باشند. در تمام مسائل حل شده در این مطالعه حداکثر زمان جستجو برای رسیدن به جوابی شدنی با فرض معین بودن طول دنباله جواب ۳ ثانیه می‌باشد. البته این سقف زمانی با توجه به پارامترهای ورودی مساله اعم از ابعاد بلوک، تعداد جرثقیل‌ها و غیره، همچنین نوع کامپیوتر مورد استفاده برای اجرای الگوریتم متفاوت خواهد بود ولی در مقاله جاری ما سقف زمانی مذکور را برابر با ۳ ثانیه قرارداد می‌کنیم.

در ادامه ابتدا یک نمونه از مسائل حل شده توسط لی و لی [۲] را توسط الگوریتم پیشنهادی خود حل و تشریح نموده‌ایم. همان‌طور که در شکل (۵) مشاهده می‌شود بلوک مربوط به این مثال که در جدول (۱) با شناسه "R011606_0070_004" آورده شده است به صورت یک دسته حاوی ۷۰ کانتینر می‌باشد. این دسته دارای ۱۶ ستون بوده که هر ستون حداکثر ظرفیتی برابر با ۶ کانتینر دارد. حداقل تعداد جابجایی‌های حاصله از الگوریتم لی و لی [۲] برای این نمونه مثال ۱۵۸ مورد می‌باشد، در حالی که تعداد این جابجایی‌ها توسط الگوریتم پیشنهادی ما تا ۱۱۶ مورد کاهش یافته که حاکی از بهبود ۸۴ درصدی در تعداد جابجایی‌ها با توجه به کران پایین مساله یعنی ۱۰۸ می‌باشد. همچنین زمان جابجایی کانتینرها توسط جرثقیل نیز از ۱۳۸۲۳ ثانیه به ۸۸۶۷ ثانیه کاهش یافته که حاکی از بهبود نسبی ۳۶ درصدی می‌باشد.

قرارداد می‌کنیم که منظور از (a:b,c)، انتقال کانتینر شماره a از ستون شماره b به ستون شماره c باشد، همچنین در صورتی که مقدار c برابر صفر باشد منظور خروج کانتینر از ستون مربوطه می‌باشد. با توجه به توضیحات فوق جواب حاصله از الگوریتم پیشنهادی ما به صورت زیر می‌باشد:

(۱:۵,۰)، (۲۷:۱,۲)، (۵۴:۱,۵)، (۶۳:۱,۵)، (۲:۱,۰)، (۱۲:۱۲,۱۴)، (۷۰:۱۲,۵)، (۴۰:۱۲,۱)، (۲۸:۱۲,۱)، (۳:۱۲,۰)، (۶:۶,۳)، (۴:۶,۰)، (۵:۱۳,۰)، (۶:۳,۰)، (۵۰:۱۶,۱۲)، (۳۹:۱۶,۵)، (۶۵:۱۶,۱۲)، (۴۶:۱۶,۱۲)، (۷:۱۶,۰)، (۵۵:۳,۱۲)، (۲۴:۳,۲)، (۸:۳,۰)، (۲۲:۱۰,۱۱)، (۹:۱۰,۰)، (۲۶:۹,۱)، (۴۲:۹,۱۲)، (۵۸:۹,۱۲)، (۵۱:۹,۵)، (۱۰:۹,۰)، (۱۱:۱۶,۰)، (۱۲:۱۴,۰)، (۵۲:۳,۱۶)، (۱۳:۳,۰)، (۱۸:۷,۱۵)، (۵۳:۷,۱۶)، (۶۷:۷,۳)، (۵۹:۷,۳)، (۱۴:۷,۰)، (۱۵:۱۴,۰)، (۴۷:۸,۱۶)، (۱۶:۸,۰)، (۳۶:۴,۱۶)، (۱۷:۴,۰)، (۱۸:۱۵,۰)، (۴۴:۸,۴)، (۱۹:۸,۰)، (۶۲:۱۵,۷)، (۲۰:۱۵,۰)، (۳۰:۶,۸)، (۵۶:۶,۷)، (۴۱:۶,۴)، (۲۱:۶,۰)، (۲۲:۱۱,۰)، (۶۴:۱۱,۶)، (۲۳:۱۱,۰)، (۲۴:۲,۰)، (۳۵:۱۰,۱۶)، (۲۵:۱۰,۰)، (۲۶:۱,۰)، (۲۷:۲,۰)، (۲۸:۱,۰)، (۲۹:۱۴,۰)، (۳۰:۸,۰)، (۳۱:۸,۰)، (۳۳:۲,۱۳)، (۶۱:۲,۶)، (۶۸:۲,۸)، (۳۲:۲,۰)، (۳۳:۱۳,۰)، (۳۴:۱۳,۰)، (۳۵:۱۶,۰)، (۳۶:۱۶,۰)، (۶۹:۱۰,۲)، (۳۷:۱۰,۰)، (۳۸:۹,۰)، (۵۱:۵,۷)، (۳۹:۵,۰)، (۴۰:۱,۰)، (۴۱:۴,۰)، (۵۸:۱۲,۶)، (۴۲:۱۲,۰)، (۶۰:۱۱,۸)، (۴۳:۱۱,۰)، (۴۴:۴,۰)، (۴۵:۱۳,۰)، (۵۵:۱۲,۴)، (۴۶:۱۲,۰)، (۴۷:۱۶,۰)، (۴۸:۱,۰)، (۷۰:۵,۱)، (۶۳:۵,۲)، (۵۴:۵,۴)، (۴۹:۵,۰)، (۶۵:۱۲,۱)، (۵۰:۱۲,۰)، (۵۱:۷,۰)، (۵۳:۱۶,۴)، (۵۲:۱۶,۰)، (۵۳:۴,۰)، (۵۴:۴,۰)، (۵۵:۴,۰)، (۵۶:۷,۰)، (۵۷:۴,۰)، (۵۸:۶,۰)، (۵۹:۳,۰)، (۶۰:۸,۰)، (۶۱:۶,۰)، (۶۲:۷,۰)، (۶۳:۲,۰)، (۶۴:۶,۰)، (۶۵:۱,۰)، (۶۶:۷,۰)، (۶۷:۳,۰)، (۶۸:۸,۰)، (۶۹:۲,۰)، (۷۰:۱,۰).

جدول ۱. مقایسه نتایج حاصله از الگوریتم پیشنهادی با نتایج لی و لی [۲] برای نمونه مسائل ساده

شناسه نمونه مساله	تعداد دسته	حداکثر ارتفاع	تعداد کانتینرها	کران پایین تعداد حرکات	تعداد حرکات		زمان کارکرد جرثقیل (ثانیه)		زمان پردازش (ثانیه)		بهبود نسبی زمان کارکرد جرثقیل	بهبود نسبی در تعداد حرکات
					لی و لی [۲]	نتایج پژوهش جاری	لی و لی [۲]	نتایج پژوهش جاری	لی و لی [۲]	نتایج پژوهش جاری		
R011606_0070_001	۱	۶	۷۰	۱۰۰	۱۱۸	۱۰۷	۱۰۸۳۲	۸۲۵۷	۶۳۰۸	<۳	۲۳.۸	۶۱.۱
R011606_0070_002	۱	۶	۷۰	۱۰۴	۱۱۷	۱۰۸	۱۰۸۴۰	۸۳۱۳	۱۱۰۸۶	<۳	۲۳.۳	۶۹.۲
R011606_0070_003	۱	۶	۷۰	۱۰۴	۱۱۰	۱۰۸	۱۰۳۲۶	۸۴۰۲	۵۵۰۶	<۳	۱۸.۶	۳۳.۳
R011606_0070_004	۱	۶	۷۰	۱۰۸	۱۵۸	۱۱۶	۱۳۸۲۳	۸۸۶۷	۹۰۳۱	<۳	۳۵.۹	۸۴
R011606_0070_005	۱	۶	۷۰	۱۰۶	۱۲۴	۱۱۰	۱۱۴۰۵	۸۴۳۰	۹۱۱۷	<۳	۲۶.۱	۷۷.۸
R021606_0140_001	۲	۶	۱۴۰	۲۰۸	۲۲۸	۲۱۸	۲۱۷۷۱	۲۰۲۳۱	۲۱۶۰۰	<۶	۷.۱	۵۰
R021606_0140_002	۲	۶	۱۴۰	۱۹۷	۲۲۴	۲۱۰	۲۱۴۳۵	۱۹۱۳۱	۲۱۶۰۰	<۶	۱۰.۷	۵۱.۹
R021606_0140_003	۲	۶	۱۴۰	۲۱۱	۲۴۷	۲۲۰	۲۳۲۰۷	۲۰۰۸۵	۲۱۶۰۰	<۶	۱۳.۵	۷۵
R021606_0140_004	۲	۶	۱۴۰	۲۱۹	۲۳۵	۲۳۱	۲۲۳۴۷	۲۱۲۲۷	۲۱۶۰۰	<۶	۵	۲۵
R021606_0140_005	۲	۶	۱۴۰	۲۱۰	۲۱۷	۲۲۰	۲۰۹۸۵	۱۹۶۹۸	۲۱۵۹	<۶	۶.۱	-۳۰

								۶				
R041606_0280_001	۴	۶	۲۸۰	۴۳۹	۵۰۲	۴۶۵	۴۹۰۷۴	۴۶۰۴۶	۲۱۵۹۹	<۱۲	۶.۲	۵۸.۷
R041606_0280_002	۴	۶	۲۸۰	۴۲۳	۴۵۰	۴۴۶	۴۵۴۴۷	۴۳۴۵۶	۲۱۶۰۰	<۱۲	۴.۴	۱۴.۸
R041606_0280_003	۴	۶	۲۸۰	۴۱۵	۴۵۰	۴۳۶	۴۵۱۷۲	۴۳۱۱۶	۲۱۶۰۰	<۱۲	۴.۶	۴۰
R041606_0280_004	۴	۶	۲۸۰	۴۲۶	۴۳۰	۴۴۰	۴۴۰۸۰	۴۳۵۶۶	۲۱۶۰۰	<۱۲	۱.۲	-۷۱.۴
R041606_0280_005	۴	۶	۲۸۰	۴۳۱	۴۳۹	۴۴۵	۴۴۵۴۳	۴۴۳۳۲	۲۱۶۰۰	<۱۲	۰.۵	-۴۲.۹
R061606_0430_001	۶	۶	۴۳۰	۶۶۰	۷۵۶	۶۹۷	۷۸۲۸۲	۷۱۶۶۳	۲۱۶۰۰	<۱۸	۸.۵	۶۱.۵
R061606_0430_002	۶	۶	۴۳۰	۶۵۴	۶۹۵	۶۹۱	۷۴۰۱۱	۷۰۸۴۲	۲۱۶۰۰	<۱۸	۴.۳	۹.۸
R061606_0430_003	۶	۶	۴۳۰	۶۵۶	۶۹۸	۶۸۲	۷۳۵۵۸	۶۹۹۰۸	۲۱۶۰۰	<۱۸	۵	۳۸.۱
R061606_0430_004	۶	۶	۴۳۰	۶۴۸	۶۹۹	۶۸۶	۷۴۰۳۸	۶۹۹۹۵	۲۱۶۰۰	<۱۸	۵.۵	۲۵.۵
R061606_0430_005	۶	۶	۴۳۰	۶۶۰	۷۰۱	۶۹۱	۷۴۵۳۴	۷۰۷۳۱	۲۱۶۰۰	<۱۸	۵.۱	۲۴.۴
R081606_0570_001	۸	۶	۵۷۰	۸۶۹	۹۲۴	۹۰۴	۱۰۳۱۲۸	۹۵۱۶۰	۲۱۶۰۰	<۲۴	۷.۷	۳۶.۴
R081606_0570_002	۸	۶	۵۷۰	۸۷۴	۹۳۰	۹۰۲	۱۰۳۸۱۰	۹۵۴۴۸	۲۱۶۰۰	<۲۴	۸.۱	۵۰
R081606_0570_003	۸	۶	۵۷۰	۸۹۱	۹۸۱	۹۴۶	۱۰۷۵۱۵	۹۸۷۷۴	۲۱۶۰۰	<۲۴	۸.۱	۳۸.۹
R081606_0570_004	۸	۶	۵۷۰	۸۷۱	۹۵۲	۹۲۱	۱۰۵۲۹۷	۹۷۳۲۳	۲۱۶۰۰	<۲۴	۷.۶	۳۸.۳
R081606_0570_005	۸	۶	۵۷۰	۸۷۳	۹۴۰	۹۱۳	۱۰۵۰۲۲	۹۶۶۵۹	۲۱۶۰۰	<۲۴	۸	۴۰.۳
R101606_0720_001	۱۰	۶	۷۲۰	۱۱۰۷	۱۱۶۳	۱۱۵۰	۱۳۶۵۸۳	۱۲۵۰۰۷	۲۱۶۰۰	<۳۰	۸.۵	۲۳.۲
R101606_0720_002	۱۰	۶	۷۲۰	۱۰۸۵	۱۱۳۲	۱۱۲۸	۱۳۳۳۲۷	۱۲۲۰۸۶	۲۱۶۰۰	<۳۰	۸.۴	۸.۵
R101606_0720_003	۱۰	۶	۷۲۰	۱۱۰۲	۱۲۲۵	۱۱۵۱	۱۴۰۹۳۳	۱۲۳۶۸۶	۲۱۶۰۰	<۳۰	۱۲.۲	۶۰.۲
R101606_0720_004	۱۰	۶	۷۲۰	۱۰۸۱	۱۱۶۸	۱۱۳۲	۱۳۷۲۱۰	۱۲۳۰۳۷	۲۱۶۰۰	<۳۰	۱۰.۳	۴۱.۴
R101606_0720_005	۱۰	۶	۷۲۰	۱۰۸۵	۱۱۵۸	۱۱۵۲	۱۳۵۵۲۷	۱۲۴۱۴۸	۲۱۶۰۰	<۳۰	۸.۴	۸.۲
R011608_0090_001	۱	۸	۹۰	۱۴۳	۱۹۰	۱۵۴	۱۶۷۷۲	۱۱۸۳۹	۱۳۲۸۳	<۳	۲۹.۴	۷۶.۶
R011608_0090_002	۱	۸	۹۰	۱۳۹	۱۹۱	۱۵۱	۱۶۸۸۳	۱۱۵۹۷	۱۱۱۴۴	<۳	۳۱.۳	۷۶.۹
R011608_0090_003	۱	۸	۹۰	۱۴۲	۲۱۶	۱۵۸	۱۸۸۵۶	۱۲۰۷۲	۲۱۶۰۰	<۳	۳۶	۷۸.۴
R011608_0090_004	۱	۸	۹۰	۱۴۳	۱۷۸	۱۵۱	۱۵۹۲۱	۱۱۵۴۵	۷۰۵۱	<۳	۲۷.۵	۷۷.۱
R011608_0090_005	۱	۸	۹۰	۱۴۳	۱۸۲	۱۵۱	۱۶۲۸۱	۱۱۸۴۱	۱۳۷۵۰	<۳	۲۷.۳	۷۹.۵
R021608_0190_001	۲	۸	۱۹۰	۳۰۵	۴۲۳	۳۴۳	۳۸۱۰۱	۳۰۴۰۷	۲۱۶۰۰	<۶	۲۰.۲	۶۷.۸
R021608_0190_002	۲	۸	۱۹۰	۳۰۹	۳۵۹	۳۲۸	۳۳۵۱۱	۲۹۱۶۶	۲۱۶۰۰	<۶	۱۳	۶۲
R021608_0190_003	۲	۸	۱۹۰	۳۰۲	۳۷۳	۳۲۵	۳۴۵۶۰	۲۹۴۳۰	۲۱۶۰۰	<۶	۱۴.۸	۶۷.۶
R021608_0190_004	۲	۸	۱۹۰	۳۰۳	۳۵۱	۳۲۵	۳۲۷۶۲	۲۸۹۷۸	۲۱۶۰۰	<۶	۱۱.۵	۵۴.۲
R021608_0190_005	۲	۸	۱۹۰	۳۱۰	۳۳۳	۳۴۳	۳۱۷۹۴	۳۰۰۰۶	۲۱۶۰۰	<۶	۵.۶	-۳۰.۳
R041608_0380_001	۴	۸	۳۸۰	۶۰۲	۸۳۰	۶۶۷	۷۷۷۷۸	۶۴۹۶۷	۲۱۶۰۰	<۱۲	۱۶.۵	۷۱.۵
R041608_0380_002	۴	۸	۳۸۰	۶۱۷	۸۰۴	۶۸۵	۷۷۰۵۳	۶۵۲۷۳	۲۱۶۰۰	<۱۲	۱۵.۳	۶۳.۶
R041608_0380_003	۴	۸	۳۸۰	۶۰۳	۶۸۴	۶۵۲	۶۷۶۳۴	۶۳۴۳۹	۲۱۶۰۰	<۱۲	۶.۲	۳۹.۵
R041608_0380_004	۴	۸	۳۸۰	۶۱۴	۷۵۵	۶۶۴	۷۲۹۳۲	۶۳۹۴۴	۲۱۶۰۰	<۱۲	۱۲.۳	۶۴.۵
R041608_0380_005	۴	۸	۳۸۰	۶۱۷	۷۷۳	۶۶۸	۷۴۴۹۳	۶۴۰۲۴	۲۱۶۰۰	<۱۲	۱۴.۱	۶۷.۳
R061608_0570_001	۶	۸	۵۷۰	۹۰۴	۱۱۴۳	۱۰۰۰	۱۱۵۹۲۴	۱۰۰۰۸۴	۲۱۶۰۰	<۱۸	۱۳.۷	۵۹.۸
R061608_0570_002	۶	۸	۵۷۰	۸۹۷	۱۳۵۳	۹۹۹	۱۲۹۵۹۳	۹۹۶۴۹	۲۱۶۰۰	<۱۸	۲۳.۱	۷۷.۶
R061608_0570_003	۶	۸	۵۷۰	۹۱۳	۱۱۳۹	۱۰۰۱	۱۱۵۷۸۱	۱۰۰۲۲۷	۲۱۶۰۰	<۱۸	۱۳.۴	۶۱.۱
R061608_0570_004	۶	۸	۵۷۰	۹۰۲	۱۲۴۲	۱۰۱۱	۱۲۲۶۱۴	۱۰۲۰۲۹	۲۱۶۰۰	<۱۸	۱۶.۸	۶۷.۹
R061608_0570_005	۶	۸	۵۷۰	۹۱۴	۱۳۳۳	۱۰۱۷	۱۲۹۱۵۰	۱۰۲۶۱۶	۲۱۶۰۰	<۱۸	۲۰.۵	۷۵.۴
میانگین											۱۳.۴	۴۵.۲

تمامی نمونه مثال‌های بررسی شده در این مطالعه از طریق لینک ارائه شده در مقاله لی و لی [۲] در دسترس است. این نمونه مسائل شامل دو گروه مسائل ساده و مسائل پیچیده می‌باشند. تفاوت این دو سری مسائل در نوع چینش کانتینرها می‌باشد به طوری که در مسائل پیچیده‌تر انسداد مسیر کانتینرهای با تقدم بیشتر توسط کانتینرهای با تقدم کمتر به میزان بیشتری لحاظ شده است. در اکثر مسائل بلوک‌ها حداکثر تا سقف ۷۵ درصد ظرفیت خود پر شده‌اند، که اندکی از مقدار مشاهده شده در دنیای واقعی بیشتر می‌باشد [۲].

جدول ۲. مقایسه نتایج حاصله از الگوریتم پیشنهادی با نتایج لی و لی [۲] برای نمونه مسائل پیچیده

شناسه نمونه مساله	تعداد دسته	حداکثر ارتفاع	تعداد کانتینر ها	کران پایین تعداد حرکات	تعداد حرکات		زمان کارکرد جرثقیل (ثانیه)		زمان پردازش (ثانیه)		بهبود نسبی زمان کارکرد جرثقیل	بهبود نسبی در تعداد حرکات
					لی و لی [۲]	نتایج پژوهش جاری	لی و لی [۲]	نتایج پژوهش جاری	لی و لی [۲]	نتایج پژوهش جاری		
U011606_0070_001	۱	۶	۷۰	۱۲۵	۱۲۵	۱۲۵	۱۱۳۴۲	۹۳۹۱	۱۷۳۳۴	<۳	۱۷.۲	۰
U011606_0070_002	۱	۶	۷۰	۱۲۴	۱۳۰	۱۲۸	۱۱۷۳۷	۹۶۰۹	۱۱۲۵۳	<۳	۱۸.۱	۳۳.۳
U021606_0140_001	۲	۶	۱۴۰	۲۴۸	۲۵۵	۲۵۳	۲۳۶۸۲	۲۲۰۶۳	۲۱۵۹۹	<۶	۶.۸	۲۸.۶
U021606_0140_002	۲	۶	۱۴۰	۲۴۸	۲۵۹	۲۵۳	۲۴۲۸۲	۲۲۶۰۹	۲۱۶۰۰	<۶	۶.۹	۵۴.۵
U041606_0280_001	۴	۶	۲۸۰	۴۹۷	۵۰۵	۴۹۹	۴۹۹۹۵	۴۷۵۹۵	۲۱۶۰۰	<۱۲	۴.۸	۷۵
U041606_0280_002	۴	۶	۲۸۰	۴۹۶	۵۱۰	۵۰۵	۵۰۰۹۰	۴۷۲۳۹	۲۱۶۰۰	<۱۲	۵.۷	۳۵.۷
U061606_0430_001	۶	۶	۴۳۰	۷۶۴	۷۸۹	۷۷۰	۸۱۷۰۰	۷۵۱۰۲	۲۱۶۰۰	<۱۸	۸.۱	۷۶
U061606_0430_002	۶	۶	۴۳۰	۷۶۴	۷۹۷	۷۷۷	۸۲۰۸۹	۷۵۷۳۳	۲۱۶۰۰	<۱۸	۷.۷	۶۰.۶
U081606_0570_001	۸	۶	۵۷۰	۱۰۱۳	۱۰۵۹	۱۰۱۹	۱۱۵۰۵۶	۱۰۲۵۲۵	۲۱۶۰۰	<۲۴	۱۰.۹	۸۷
U081606_0570_002	۸	۶	۵۷۰	۱۰۱۴	۱۰۲۹	۱۰۲۱	۱۱۲۳۵۵	۱۰۰۱۲۵	۲۱۶۰۰	<۲۴	۱۰.۹	۵۳.۳
U101606_0720_001	۱۰	۶	۷۲۰	۱۲۸۱	۱۳۶۸	۱۲۹۳	۱۵۵۰۹۵	۱۳۱۹۶۸	۲۱۶۰۰	<۳۰	۱۴.۹	۸۶.۲
U101606_0720_002	۱۰	۶	۷۲۰	۱۲۸۱	۱۳۲۲	۱۲۹۱	۱۵۱۱۰۷	۱۳۲۲۴۱	۲۱۶۰۰	<۳۰	۱۲.۵	۷۵.۶
U011608_0090_001	۱	۸	۹۰	۱۶۴	۱۷۵	۱۶۶	۱۵۷۱۶	۱۲۷۷۳	۲۱۶۰۰	<۳	۱۸.۷	۸۱.۸
U011608_0090_002	۱	۸	۹۰	۱۶۴	۱۸۰	۱۶۸	۱۶۲۳۶	۱۳۱۵۲	۸۰۴۳	<۳	۱۹	۷۵
U021608_0190_001	۲	۸	۱۹۰	۳۴۸	۳۸۷	۳۵۴	۳۵۲۲۹	۳۰۵۱۹	۲۱۶۰۰	<۶	۱۳.۴	۸۴.۶
U021608_0190_002	۲	۸	۱۹۰	۳۴۸	۴۱۴	۳۵۸	۳۷۶۷۷	۳۱۳۸۹	۲۱۶۰۰	<۶	۱۶.۷	۸۴.۸
U041608_0380_001	۴	۸	۳۸۰	۶۹۶	۸۴۸	۷۱۰	۷۹۳۰۵	۶۸۲۳۴	۲۱۶۰۰	<۱۲	۱۴	۹۰.۸
U041608_0380_002	۴	۸	۳۸۰	۶۹۶	۷۵۲	۷۰۳	۷۲۹۲۱	۶۶۸۳۴	۲۱۶۰۰	<۱۲	۸.۳	۸۷.۵
U061608_0570_001	۶	۸	۵۷۰	۱۰۴۴	۱۲۳۰	۱۰۶۲	۱۲۱۹۴۹	۱۰۳۴۸۰	۲۱۶۰۰	<۱۸	۱۵.۱	۹۰.۳
U061608_0570_002	۶	۸	۵۷۰	۱۰۴۴	۱۲۳۸	۱۰۵۸	۱۲۳۱۳۵	۱۰۳۳۸۱	۲۱۶۰۰	<۱۸	۱۶	۹۲.۸
میانگین											۱۲.۳	۶۷.۷

نتایج حاصله از لی و لی [۲] و الگوریتم پیشنهادی ما برای مسائل ساده و پیچیده به ترتیب در جداول (۱) و (۲) آورده شده‌اند، به طوریکه ستون‌های سمت چپ شامل مشخصات نمونه مثال‌ها، ستون‌های میانی شامل نتایج حاصله از دو الگوریتم و سه ستون سمت راست شامل نتایج مربوط به مقایسه دو روش می‌باشد. جدول (۱) شامل ۵۰ نمونه مساله مختلف می‌باشد تا توانایی الگوریتم در ازای پارامترهای ورودی مختلف مساله سنجیده شود. به طور مشابه جدول (۲) نیز شامل ۲۰ نمونه مساله نسبتا مشکل‌تر می‌باشد.

نتایج حاصله از الگوریتم پیشنهادی از سه جهت کاهش زمان کارکرد جرثقیل، کاهش تعداد جابجایی‌ها و همچنین زمان لازم برای رسیدن به جواب با الگوریتم پیشنهادی لی و لی [۲] که جدیدترین مطالعه موجود در ادبیات می‌باشد مقایسه شده است. همان‌طور که ملاحظه می‌شود برای هر دو نوع مسائل ساده و پیچیده نتایج حاصله نشان دهنده بهبودهای قابل توجهی می‌باشند به طوری که برای مسائل ساده و پیچیده، زمان رسیدن به جواب به طور اساسی کاهش یافته است. این در حالی است که در اکثر موارد جواب‌های گزارش شده در کمتر یک ثانیه حاصل شده‌اند و همان‌طور که در مورد نمونه مثال ارائه شده توضیح داده شد زمان‌های گزارش شده معمولا صرف جستجوی اضافی برای یافتن جواب‌های بهتر شده است که در صورت صرف نظر نمودن از آن بهبود مذکور بسیار بیشتر از مقدار گزارش شده می‌باشد و زمان حصول همین جواب‌ها یا جواب‌هایی تقریبا برابر با آن‌ها تقریبا در تمام موارد کمتر از یک یا دو ثانیه خواهد بود.

در مورد زمان کارکرد جرثقیل که در این مطالعه به عنوان تابع هدف اصلی مورد نظر بوده است (هر چند که به طور غیر مستقیم به آن پرداخته شده است) بهبود قابل توجهی را شاهد می‌باشیم به طوری که میانگین درصدهای بهبود نسبی در مسائل ساده و پیچیده به ترتیب برابر با ۱۳.۴ و ۱۲.۳ درصد می‌باشد.

تعداد جابجایی‌ها که در مطالعات مربوط به حالت ساده مساله [۱و۴] به عنوان تابع هدف اصلی لحاظ شده است، جز اندک مواردی برای تمام نمونه مسائل حل شده نشان دهنده بهبود قابل توجهی می‌باشد زیرا به طور میانگین، به ترتیب برای مسائل ساده

و مشکل ۴۵.۲ و ۶۷.۷ درصد بهبود را نشان می‌دهد. در محاسبه این درصد بهبود به کران پایین هر نمونه مساله توجه شده است به طوری که در هر مورد میزان بهبود حاصله را بر حداکثر بهبود ممکن (که با توجه به کران پایین تعداد جابجایی‌های در مساله مربوطه محاسبه می‌شود) تقسیم نموده‌ایم. همچنین در جدول (۳) نتایج حاصله از این مطالعه را با نتایج حاصله از لی و لی [۲] و همچنین کیم و هونگ [۱] برای ۵ نمونه مساله ساده و ۵ نمونه مساله پیچیده مقایسه نموده‌ایم. از آنجا که تابع هدف در مطالعه صورت گرفته توسط کیم و هونگ [۱] صرفا کاهش تعداد جابجایی‌ها بوده است مقایسات موجود در جدول (۳) مربوط به تعداد جابجایی‌ها می‌شود و شامل زمان کارکرد جرثقیل نمی‌گردد. مجددا نتایج آورده شده در جدول (۳) حاکی از بهبود قابل توجهی می‌باشند به طوریکه نسبت تعداد جابجایی‌های حاصله از روش‌های پیشنهادی کیم و هونگ [۱] و لی و لی [۲] به تعداد جابجایی‌های حاصله از این مطالعه به طور متوسط و به ترتیب ۱.۷۵ و ۱.۱۹ می‌باشد.

در هر صورت توجه به این نکته ضروری است که تابع هدف تعداد جابجایی‌ها در حضور تابع هدف کاهش زمان کارکرد جرثقیل هرگز معیار مناسبی برای مقایسه الگوریتم‌های مورد مقایسه نمی‌باشد زیرا همانطور که برای نمونه در مورد مثال "R041606_0280_004" مشاهده می‌گردد، اگرچه تعداد جابجایی‌های حاصله از الگوریتم پیشنهادی ما ۱۰ مورد بیشتر از تعداد جابجایی‌های حاصله از الگوریتم لی و لی [۲] می‌باشد ولی زمان کارکرد جرثقیل متناظر با این جواب نسبت به لی و لی [۲] مقدار کمتری می‌باشد.

جدول ۳. مقایسه نتایج حاصله از الگوریتم پیشنهادی با مطالعات [۱ و۲]

شناسه نمونه مساله	R011606_0070_01	R011606_0070_02	R011606_0070_03	R011606_0070_04	R011606_0070_05	R011608_0090_01	R011608_0090_02	R011608_0090_03	R011608_0090_04	R011608_0090_005	میانگین نسبت‌ها
کران پایین	۱۰۰	۱۰۴	۱۰۴	۱۰۸	۱۰۶	۱۴۳	۱۳۹	۱۴۲	۱۴۳	۱۴۳	
نتایج پژوهش جاری	۱۰۷	۱۰۸	۱۰۸	۱۱۶	۱۱۰	۱۵۴	۱۵۱	۱۵۸	۱۵۱	۱۵۱	
لی و لی [۲]	۱۱۸	۱۱۷	۱۱۰	۱۵۸	۱۲۴	۱۹۰	۱۹۱	۲۱۶	۱۷۸	۱۸۲	
کیم و هونگ [۱]	۱۷۳	۱۷۴	۱۷۶	۱۸۲	۱۸۴	۳۰۳	۲۵۳	۳۱۵	۲۸۳	۲۸۳	
مقایسه با لی و لی [۲]	۱.۱	۱.۰۸	۱.۰۲	۱.۳۶	۱.۱۳	۱.۲۳	۱.۲۶	۱.۳۷	۱.۱۸	۱.۲۱	۱.۱۹
مقایسه با کیم و هونگ [۱]	۱.۶۲	۱.۶۱	۱.۶۳	۱.۵۷	۱.۶۷	۱.۹۷	۱.۶۸	۱.۹۹	۱.۸۷	۱.۸۷	۱.۷۵

به طور کلی نتایج حاصله هم به لحاظ زمان رسیدن به جواب و هم از نظر جواب‌های حاصله نشان دهنده بهبود قابل توجهی می‌باشند.

۵. نتیجه‌گیری و جمع‌بندی

در این مقاله ادبیات مساله جابجایی کانتینرها را بر حسب محدودیت جابجایی کانتینر بین دسته‌های مختلف به دو گروه ساده و تعمیم یافته تقسیم نمودیم. هدف از این مقاله ارائه الگوریتمی ابتکاری برای حل تعمیم یافته مساله جابجایی کانتینرها می‌باشد. برای این منظور با توجه به عمر بیشتر حالت ساده مساله در مقایسه با حالت تعمیم یافته از روش‌های موجود در ادبیات حالت ساده مساله برای حل تعمیم یافته مساله که اخیرا مورد توجه بوده است استفاده نمودیم. حذف محدودیت جابجایی کانتینر بین دسته‌های مختلف منجر به مسائل با اندازه بزرگ‌تری خواهد شد، به طوریکه در اکثر مسائل مربوط به حالت ساده مساله حداکثر تعداد کانتینرهای موجود در مسائل از ۱۰۰ کانتینر تجاوز نمی‌نماید، در حالی که در مطالعه جاری مسائلی با اندازه ۷۲۰ کانتینر نیز به نحوی موثر حل شدند. با توجه به اندازه بزرگ مساله و ماهیت آن که منجر به استفاده از روش شاخه و حد می‌شود، در طراحی الگوریتم پیشنهادی از رویکرد شکستن مساله به مسائل کوچک‌تر استفاده شده است.

سهم ما در این تحقیق از دو جنبه می‌باشد. اولا تعداد جابجایی-های مورد نیاز برای تخلیه بلوک و لذا زمان لازم برای انجام این جابجایی‌ها توسط جرثقیل را جز اندک مواردی به طور قابل توجهی در مقایسه با جدیدترین کارهای موجود در ادبیات حالت تعمیم یافته مساله کاهش داده‌ایم، به طوریکه متوسط بهبود نسبی در زمان کارکرد جرثقیل برای مسائل ساده و پیچیده نسبت به الگوریتم پیشنهادی لی و لی [۲] به ترتیب ۱۳.۴ و ۱۲.۳ درصد می‌باشد. همچنین تعداد جابجایی‌ها که در مطالعات مربوط به حالت ساده مساله به عنوان تابع هدف اصلی مطرح بوده است با توجه به کران پایین مربوطه به طور متوسط برای مسائل ساده و پیچیده به ترتیب ۴۵.۲ و ۶۷.۷ درصد از فاصله موجود تا کران پایین مسائل را پوشش داده است. در هر صورت با توجه به نتایج حاصله از این تحقیق ادعا می‌شود در مواردی که بلوک کانتینرها شامل بیش از یک دسته باشد تابع هدف کاهش تعداد جابجایی لزوما تابع هدف مناسبی نمی‌باشد و بایستی با تابع هدف کاهش زمان کارکرد جرثقیل جایگزین گردد. برای مثال همانطور که برای نمونه در مورد مثال "R041606_0280_004" از جدول (۱) مشاهده می‌گردد، اگرچه تعداد جابجایی‌های حاصله از الگوریتم پیشنهادی ما ۱۰ مورد بیشتر از تعداد جابجایی‌های حاصله از الگوریتم لی و لی

[۲] می‌باشد ولی زمان کارکرد جرثقیل متناظر با این جواب نسبت به لی و لی [۲] مقدار کمتری می‌باشد.

همچنین زمان رسیدن به جواب به طور چشمگیری کاهش یافته است، به طوریکه زمان لازم برای رسیدن به جواب در مسائل ساده و پیچیده گاها تا چند هزار برابر کاهش یافته است. این در حالی است در اکثر موارد زمان لازم برای رسیدن به جواب گزارش شده کمتر از یک یا دو ثانیه بوده است و باقی زمان‌ها صرف جستجوی اضافی جهت یافتن جواب‌های بهتر شده است. به عبارت دیگر هر چند زمان‌های گزارش شده به اندازه کافی برای اهداف عملیاتی کاهش یافته‌اند ولی در صورت لزوم می‌توان با اعمال تغییراتی اندک در الگوریتم زمان لازم برای رسیدن به همین جواب‌ها یا جواب‌هایی تقریبا برابر با این جواب‌ها را برای اکثر نمونه مثال‌های حل شده به کمتر از یک یا دو ثانیه کاهش داد. در صورتی که در انجام جابجایی‌ها از بیش از یک جرثقیل استفاده شود با توجه به امکان تداخل مسیر آن‌ها، همچنین با در نظر گرفتن جرثقیل‌هایی با امکان انتقال بیش از یک کانتینر به طور همزمان، طراحی الگوریتم‌هایی متناسب با این شرایط جدید می‌تواند در تعریف زمینه‌های پژوهشی جدید مد نظر قرار گیرد. همچنین بررسی امکان استفاده از الگوریتم استفاده شده در این مقاله در حل مسائلی با ماهیتی مشابه مساله جابجایی کانتینرها همچون مساله مرتب سازی کانتینرها می‌تواند در تعریف زمینه‌های پژوهشی جدید مورد توجه قرار گیرد. پژوهش آتی می‌تواند بر ارائه روشی برای رسیدن به جواب بهینه و یا بهبود جواب فعلی از نظر کاهش زمان کارکرد جرثقیل متمرکز باشد.

منابع

[1] Kim, K. H. and G.-P. Hong (2006). "A heuristic rule for relocating blocks." Computers & Operations Research., Vol. 33, No. 4, PP. 940-954.

[2] Lee, Y. and Y.-J. Lee (2010). "A heuristic for retrieving containers from a yard." Computers & Operations Research., Vol. 37, No. 6, PP. 1139-1147.

[3] Caserta, M., Voß, S., & Sniedovich, M. (2011). "Applying the corridor method to a blocks relocation problem." OR Spectrum., Vol. 33, No. 4, PP. 915-929.

[4] Forster, F. and A. Bortfeldt (2012). "A tree search procedure for the container relocation problem." Computers & O*perations Research*, Vol. 39, No. 2, PP. 299-309.

[5] Lee, Y. and S.-L. Chao (2009). "A neighborhood search heuristic for pre-marshalling export containers." European Journal of Operational Research., Vol. 196, No. 2, PP. 468-475.

[6] Hwang, H. and S.-B. Lee (1990)."Travel-time models considering the operating characteristics of the storage and retrieval machine." International Journal of Production Research, Vol. 28, No. 10, PP. 1779-1789.

[7] Caserta, M., Schwarze, S., & Voß, S. (2009). A New Binary Description of the Blocks Relocation Problem and Benefits in a Look Ahead Heuristic, Proc., 9rd European Conference on Evolutionary Computation in Combinatorial Optimization., Springer-Verlag Berlin, Heidelberg, PP. 37-48.

[8] Caserta, M., S. Schwarze, et al. (2012). "A mathematical formulation and complexity considerations for the blocks relocation problem." European Journal of Operational Research 219(1): 96-104.

[9] Ünlüyurt, T. and C. Aydın (2012). "Improved rehandling strategies for the container retrieval process." Journal of Advanced Transportation: n/a-n/a.