International Core Journal of Engineering Volume 7 Issue 8, 2021
ISSN: 2414-1895 DOI: 10.6919/ICJE.202108_7(8).0004

A New Algorithm for the Container Pre-marshalling Problem

Weihong Shi

School of Shanghai Maritime University, Shanghai 201306, China.

1115299726@126.com

Abstract

Container pre-marshalling is a kind of transportation operation. In container terminal operation, containers are usually stacked with a certain number of layers. If the containers to be picked up are pressed down by other containers, the barrier boxes on them must be removed at this time. Due to the uncertainty in wharf operation, unpacking is almost inevitable. The average number of unpacking is an indicator to measure the level of wharf operation management. Different from previous literature, this paper designs a new algorithm to solve this problem by integrating heuristic algorithm and rule constraint. All kinds of indicators to evaluate the current layout are used as the coding part of the genetic algorithm, and the whole unpacking process is realized by constantly iterating to find the appropriate weight coefficient and emptying the stack rules. Finally, an example is given to prove the effectiveness of the algorithm.

Keywords

Container Terminal; Container Pre-marshalling.

1. Introduction

The CPMP (container pre-marshalling problem) has been studied for many years. Lots of approaches published can fall into three categories: Heuristic method,which is the main method to solve the problems. Yusin Lee a and Shih-Liang Chao [1] proposed an algorithm composed of two major tasks and three minor ones. The former mainly aims to find a reasonable and feasible solution; the latter is to enlarge the search area, Shorten sequence and guide the search direction. Since its random search, it will take longer time to find a feasible solution. Andre Hottung and Kevin Tierney [2] present a new way to solve the problem by three steps using a biased random genetic algorithm(BRKAG). The algorithm avoid the problem of uncertain length of solution and the three steps guarantee the generation of feasible solution. According the BRKAG to tune the parameter for constructing a solution in few time. Kevin Tierney, Pacino and Voß [3] use the A* and IDA* algorithms and reduce the search area by several branching and breaking rules. Shan-Huen Huang and Tsan-Hwan Lin [4] propose six heuristic steps to fill and empty the stack. First, label three types of stacks. Then determine and fill the stacks that can be filled or potentially filled to a certain height. Decompose the stacks cannot reach the height. When an empty stack appears, the container with the highest priority in the wrong stack is selected to move to the empty stack. Repeat these steps until the exit conditions are met. Ning Wang, Bo Jin and Andrew Lim [5] put forward a novel way that fixing the container in the slot one by one in descending order of priority and it only can move again in one situation. The solution is constructed by evaluation scheme and its branches are reduced by beam search. Ning Wanga, Bo Jinb, Zizhen Zhangc and Andrew Limd [6] develop a new concept on the previous basis, feasibility to ensure current layout is solvable and a scheme is designed to avoid entering into no solution. Tree search, Andreas Bortfeldt, Florian Forster [7] give the lower bound for the number of “BX” and “GX” moves and their sum is the lower bound of a layout,which can, to a certain extent, reduce the number of branches. At present, the container pre-marshalling algorithm falls into three

ISSN: 2414-1895 DOI: 10.6919/ICJE.202108_7(8).0004

main categories, tree search, heuristic algorithms, and rule-based search. In this paper, a new algorithm is developed by mixing rules and heuristics, and its effectiveness is proved by an example.

2. The pre-marshalling problem

Container pre-marshalling is a kind of transportation operation. In container terminal operation, containers are usually stacked with a certain number of layers. If the containers to be picked up are pressed down by other containers, the barrier boxes on them must be removed at this time.

The operation of moving the obstacle box away and back again is called unpacking. Due to the uncertainty in wharf operation, unpacking is almost inevitable. The average number of unpacking is an indicator to measure the level of wharf operation management.

The pre-marshalling problem focuses on sorting the containers of a single bay. Formally, a bay contains a given number of stacks S, and a maximum number of tiers, T. The function group (s,t) provides the group of the container in stack s at tier t, with group (s, t)=0 if no container is located at position s, t. The group represents the retrieval time of the container, i.e., the time point in which the container must exit the stacks. The goal is to remove all misoverlays from the bay, meaning each stack is sorted in descending order from the ground up according to the group of each container. A bay has no misoverlays if group (s, t) ≥group (s, t+1), for all $1 \leq s \leq S$, $1 \leq t \leq T$. At the same time, you can only take the top container and drop it on top of other stacks. Figure 1 illustrates a simple container pre-marshalling work project.

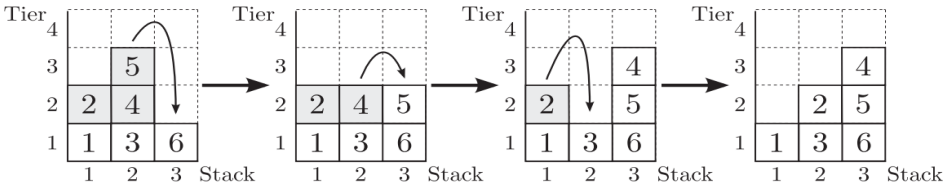

Fig. 1 An example sequence of moves to solve the pre-marshalling problem. Mis-overlaid containers are shown in gray.

3. Algorithm process

Different from the previous methods, this paper uses heuristic algorithm and rules to solve the whole problem. Fig. 2 is the main composition of the whole algorithm.

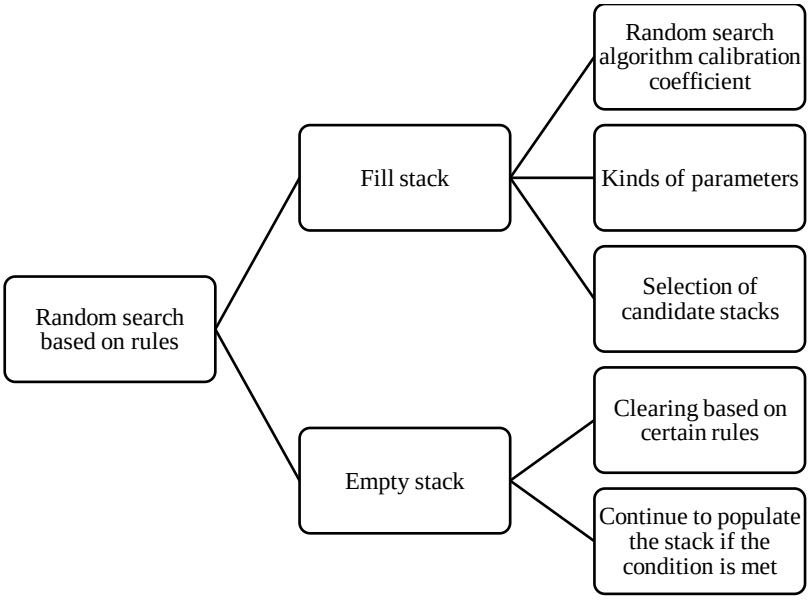

Fig. 2 The main composition of the whole algorithm

ISSN: 2414-1895 DOI: 10.6919/ICJE.202108_7(8).0004

3.1 Various parameters

Table 1 shows all the parameters required to calculate the fitness value for determining whether to move or not. D(d,r) represents the difference in priority between the container at the top of d stack and the container at the top of r stack. Δ_o is the increment of class o stacks, $o = \{R, E, W\}$, respectively, the stack with no misoverlays, the stack with no container and the stack with at least one misoverlay. The meanings of H_d and H_r are shown in Table 1.

Table 1. Parameter of the GA

Symbol	Meaning
D(d,r)	The difference between the priority of the donor stack and the receiving stack
Δ_R	The increment of class r stack
Δ_C	The number of correct container changes
H_d	Height of donation stack before moving
H_r	The height of the receiving stack before moving
Δ_W	Increases in class W stations

This paper uses these parameters to determine the good or bad of the entire bay after the current movement. There is a coefficient in front of each parameter, and this coefficient will be calibrated by genetic algorithm to determine which parameters are more important.

3.2 Empty stack

This article uses rule determination to determine whether a stack needs to be emptied and populated after emptying.

The rules are as follows:

1. When the number of defferred stacks is greater than 1, calculate the number of slots that are free except the remaining defferred stacks in the current stack and the slots needed to make the current stack into a good stack or an empty stack. If the former is less than the latter, the current stack needs to be cleared through the good stack, otherwise enter Step 2.

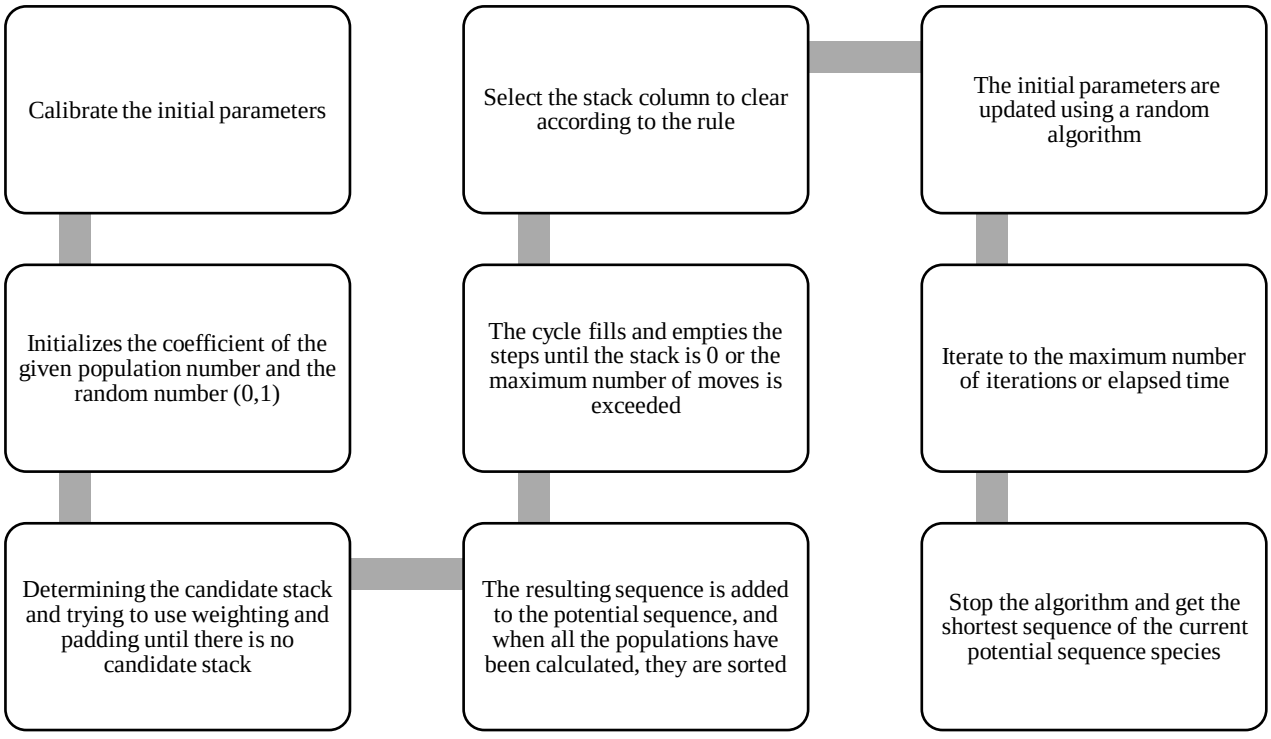

Fig. 3 The algorithm flow

ISSN: 2414-1895 DOI: 10.6919/ICJE.202108_7(8).0004

2. Clear the lowest differential stack. If there are any differential stacks of the same height, select the one with the higher priority on the top of the stack to clear it (because a lower priority stack of the same height may be used to fill the created empty stack to prevent repeated moves). If no empty stack can be obtained by performing this step, add meaningless steps to the sequence first to break out of the loop.

3. If the difference stack number is 1, use the good stack to empty the difference stack. (Attempts to fill at each step of the stack emptying process)

The actual algorithm process is shown in Figure 3. The potential sequence refers to the sequence formed by all the solutions obtained after the weighted evaluation of the parameters of all the populations and the cleaning of the stack step.

4. Numerical examples

In order to verify the effectiveness of the algorithm, this paper chooses an algorithm in a paper[4] and compares its calculation examples. Figure 4 shows the bay with S=12, T=5. The number in the grid represents the priority of the container

		4		6		5		6			
2		2		3		1	1	5			3
2	5	6		4	1	5	2	3	6	6	5
4	1	3	5	6	6	2	6	1	3	4	4
4	2	6	3	2	1	6	1	2	2	1	6

Fig. 4 The initial layout

The solution obtained in the literature is (6,1), (4,2), (4,12), (6,4), (10,4), (11,4), (5,4), (9,4), (6,8), (2,6), (2,6), (9,6), (7,6), (3,6), (7,2), (8,2), (8,2), (11,8), (7,11), (3,11), (5,11), (10,11), (9,11), (10,7), (3,7), (8,7), (3,10), (8,3), (8,10), (5,10), (12,10), (12,10), (8,9), (5,8), (12,8), totally 35 moves.

However, this paper uses C++ to implement the algorithm in Visual Studio 2019 on Windows 10 platform. The moves calculated by the method in this paper are (4,2), (4,12), (6,1), (6,4), (9,4), (2,4), (2,4), (11,4), (6,2), (10,6), (11,2), (5,6), (9,11), (9,11), (7,11), (7,1), (7,11), (3,11), (3,7), (3,6), (12,3), (12,3), (12,6), (10,12), (9,12), (5,12), (8,9), (8,7), (10,7), (8,10), (5,6), (5,10), totally 32 moves. 3 moves less than the algorithm in the above literature. Figure 5 shows the results obtained by the algorithm in this paper.

1	1		4		4	2				4	3
2	1	3	5		5	2				5	3
2	1	3	5		6	2		1		5	3
4	1	3	6		6	2		1	6	5	4
4	2	6	6	2	6	6	1	2	6	6	6

Fig. 5 The final layout

5. Conclusion

This paper proposed a new algorithm to solve the problem of container inverted box, unlike previous simple tree search, heuristic algorithm, and based on the rules of search, this paper comprehensive use of rules on top of this calibration fitness value function, using genetic algorithms for all kinds of cases, the best fitness value of the mobile, to obtain the optimal solution. Finally, the algorithm presented in this paper is compared with the algorithms in other literatures by a numerical example, and the effectiveness of this algorithm is proved.

References

[1] Lee Y, Chao S L. A neighborhood search heuristic for pre-marshalling export containers[J]. European Journal of Operational Research, 2009, 196(2):468-475.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202108_7(8).0004

[2] Hottung A, Tierney K. A biased random-key genetic algorithm for the container pre-marshalling problem [J]. Computers & Operations Research, 2016, 75(nov.):83-102.

[3] Tierney K, Pacino D, Vo S. Solving the Pre-Marshalling Problem to Optimality with A* and IDA*[J]. Flexible Services and Manufacturing Journal, 2014:223-259.

[4] Huang S H, Lin T H. Heuristic algorithms for container pre-marshalling problems[J]. Computers & Industrial Engineering, 2012, 62(1):13-20.

[5] Wang N, Jin B, Zhang Z, et al. A feasibility-based heuristic for the container pre-marshalling problem[J]. European Journal of Operational Research, 2017, 256(1):90-101.

[6] Wang N, Jin B, Lim A. Target-guided algorithms for the container pre-marshalling problem[J]. Omega, 2015, 53(jun.):67-77.

[7] Bortfeldt A, F Forster. A tree search procedure for the container pre-marshalling problem[J]. European Journal of Operational Research, 2012, 217(3):531-540.