Dear Editor,

We would like to express our gratitude for the great effort you and the anonymous referees have put in. We found most of the critiques and feedback very constructive. We have strived to address all the issues as thoroughly as possible. Consequently, we have rewritten many sections of the manuscript. The main changes are summarized as follows:

1. We added a section (Section 4.3) to substantiate why the proposed lower bounds dominate the existing lower bound of Bortfeldt and Foster.

2. We have removed feasibility check method which was in Section 4.2, as it is not the original contribution of this paper; we have mentioned it in our another paper. In addition, it is not closely related to the key ideas expressed in this paper. Section 4 only talks about lower bounds now. Since only the algorithm TGH in Section 5 uses the concept of movable (immovable) containers, we have moved this concept (Section 4.1 of old version) to the first part of Section 5.

3. We reorganized the computational experiment section.

We hope our revision meets with your satisfaction, and we look forward to your favorable response.

Regards,

Ning Wang

Bo Jin

Andrew Lim

Reviewer 1

1. In section 4.2.1, on page 8, line about 48 (the numbers on the left do not match the lines of the text in my version): The authors say "If a stack has zero immovable container, then add *H*-*im* to element *SG*. Notation *im* is the number of immovable containers in a stack...". In other words *im* = 0 in this case. Lemma 1 says, that *im* is the same for all stacks. In my opinion, in this case it should be *SG* = H × S, because there are zero immovable containers in all stacks. The formulation used by the authors seems to be right but misleading.

Answer: As Reviewer 2 suggests, the feasibility check is not the original contribution of this paper; it has appeared in our another paper. And the feasibility issue is not closely related to the key ideas of this paper, so we have removed the section about feasibility from this paper. But the suggestion of Reviewer 1 is right, thus we have revised the paper discussing feasibility accordingly.

2. In section 4.3.2, on page 11, *LP* number 4: The authors write "*xi* ϵ {0, 1}; *i* = 1,…,S". In my opinion it should be either *xi* ϵ {0, 1}; *i* = {1,…,S} or *xi* ϵ {0, 1}; *i* = 1,…,S.

Answer: We have revised the error in typing this expression and similar ones.

3. The authors present a lower bound for (CPMP) and (CPMPDS). For (CPMP) their lower bound consists of three parts. Two of them are already presented by Bortfeldt and Foster. The third part is new. The authors claim, that their lower bound dominates the lower bound of Bortfeldt and Foster, but they do not substantiate this claim. Further, they propose a maximum knapsack method to approximate their third part of the lower bound. Again, they claim, that even their approximated lower bound dominates the one of Bortfeldt and Foster without any substantiation.

Answer: We have added a Section (Section 4.3) to substantiate why our proposed lower bounds dominate that of Bortfeldt and Foster.

4. The authors say nothing about the quality of the lower bound. In my opinion, they should provide the calculated lower bound in the computational study. Otherwise it is not possible to see how useful this bound is.

Answer: We have added the comparison of two proposed lower bound computation methods with that of Bortfeldt and Foster. The results on three data sets are displayed in Section 7.1.

5. All computational results (except the ones for CPMPDS) are compared with the results of Bortfeldt and Foster. The problem is, that the authors use a much faster PC than Bortfeldt and Foster. This makes the comparison of runtimes useless. In my opinion, the authors need to test the algorithm of Bortfeldt and Foster on the same machine to get a fair comparison.

Answer: TODO

p\_ub = 1.75

Details: tabu list for the invest to former moves in BG compound moves.

The paper does not require positive clean supply for a normal move, while it requires positive clean supply for an extra move. But there is opportunity that a normal move has zero clean supply (clean stacks are full)

Sorting compound moves, the lower priority criterion is misunderstanding. Figure does not show this criterion.

Selecting the best move mbest of extra compound move is not clear.

Reviewer 2:

1. I actually doubt that the dummy stack extension of the CPMP (CPMPDS) can be applied in practice. The 'dummy stack' is placed at the truck lane that spans the whole length of a block. Since there is only one such lane (see Fig. 2), trucks cannot bypass a dummy container stack. Hence, it is impossible to serve any trucks at a block while the pre-marshalling is performed at one of the block's bays. I expect that this severely disturbs the operations of truck handling at a container terminal and that the pre-marshalling process may actually lower the productivity of the terminal.

Answer: The scenario mentioned by the reviewer is only one possible scenario. There are at least three scenarios where the CPMPDS is useful.

Scenario 1. The cranes are idle. Pre-marshalling is usually performed in cranes’ idle time, which has been mentioned by past literature.

Scenatio 2. In the truck lane, the crane which serves trucks (Crane A) is ahead of the crane which performs pre-marshalling (Crane B). When trucks get out of the block, pre-marshalling does not stand on their ways because truck lanes are unidirectional. In addition, the bays from which Crane A retrieves containers are near to each other and Crane A does not need to move to bays behind Crane B because containers destined for the same place are usually stored near to each other.

Scenario 3. There is more than one truck lane. According to the paper “Storage yard operations in container terminals: Literature overview, trends, and research directions” by Carlos et al. in EJOR, 2014 which is recommended by the reviewer 2, there may be more than one truck lane beside a block, which makes pre-marshalling possible while serving trucks.

2. According to the references, several parts of Section 4 stem from Wang et al. (2013). It needs to be stated more clearly, which parts of the material are reproduced from Wang et al. and which parts are actually new contributions of the OMEGA submission.

Answer: in the old version, the lower bound computation is the original contribution of this paper, while the feasibility check method is reproduced from Wang et al. (2013). As the feasibility check method is not closely related to the key ideas expressed in this paper, we decide to remove that part from this paper. Section 4 only talks about lower bounds. Since only the algorithm TGH in Section 5 uses the concept of movable (immovable) containers, we have moved this concept to the first part of Section 5.

3. The paper provides comprehensive computational experiments but, from my perspective, the most relevant research questions are not covered by the experimental evaluation:

* The authors propose new lower bounds and they claim that these bounds are better than those proposed by other researchers. However, the quality of these bounds is not evaluated against each other.

Answer: We have added the comparison of two proposed lower bound computation methods with that of Bortfeldt and Foster. The results on three data sets are displayed in Section 7.1.

* The heuristics are compared with each other. An objective evaluation of their capabilities would require to take into account proven optimal solution or lower bounds. Actually, the bounds proposed in this paper do not appear at all in the computational study.

Answer: we have added lower bounds of instances of data set LV, CV and BF in the last columns of Table 3, Table 4, and Table 5, respectively. For BF data groups, BS-G and BS-B can solve six and eight groups to optimality as resultants solutions is equal to lower bounds.

TODO: solution gap

* This paper proposed to extent the CPMP to the CPMPDS. However, there is no experiment, which shows how many container moves a terminal can actually save from switching to the more complex problem. Therefore, it remains an open question, whether (and to which extent) the CPMPDS provides a benefit for a terminal.