Dear Editor,

We would like to express our gratitude for the great effort you have put in. We hope our revision meets your satisfaction, and we look forward to your favorable response.

Regards,

Ning Wang

Bo Jin

Andrew Lim

Editor’s comments

If you visit OMEGA website you will notice that during the last couple of years we published extensively on this topic. Please ensure that you have included recent and relevant papers from OMEGA and other OR journals.

Answer: we have added more reference (five from Omega, and one from transportation science) to make our statements more convincing.

Reviewer 1

Minor issues

1. Page 3: "Transfer lanes are called dummy stacks": The temporary stacks that are located at the transfer lanes are called dummy stacks.

Answer: we have revised the sentence accordingly.

2. Page 3: Figure 2: There are two trucks in 2(a). These trucks are on different lanes. As far as I know, there are parking and driving lanes. Maybe, the authors should mark the lane, they are calling transfer lane. Are both lanes transfer lanes?

Answer: we have revised the captions of Figure 2, indicating that I/O points can have two forms: transfer lanes (Figure 2(a)) and transfer bays (Figure 2(b)). To keep consistent with Figure 2(b), more detailed information of transfer lanes, such as the concrete names of the two lanes, are omitted, as the intention of Figure 2 is to show scenarios where the CPMPDS is used.

3. Page 3: "hence, they cannot be implemented directly at terminal layouts as shown in Figure 2(a)": Algorithms for CPMP can be implemented for these layouts. They do not use any dummy stack, but these algorithms are able to find feasible solutions for CPMP.

Answer: we have revised the sentence to better represent the disadvantage of applying CPMP algorithms to the CPMPDS.

4. Page 5 (and other): "S \_H": In my opinion, the ×-symbol is misleading. I always read it as the Cartesian product. Better is "S · H".

Answer: we have revised all the multiplications to ·.

5. Page 15: ". . . guarantees a solution for any feasible instance.": What is a feasible instance and how is it defined? Do the authors mean "'. . . guarantees a feasible solution for any instance."?

Answer: we meant “guarantees a feasible solution for any instance”. We have revised accordingly.

Actually, not all the layouts can be rearranged to clean layouts; some layouts can never be rearranged to clean layouts. Thus, only those initial layouts that can be arranged to clean layouts are called “feasible instances” in our sense. We will discuss how to check feasible and infeasible instances in another paper. In this paper, “instances” refer to feasible instances.

6. Page 18: "The second evaluation scheme prefers small two-tuple (f(c; s); h(s) uf(s)). . . ": How do the authors compare two-tuples? They have to define something like "(a; b) < (c; d); iff a < c V a = c; b < d".

Answer: we have revised the description of the first and the second evaluation schemes on Page 18.

Reviewer 2

1. The added results show that the new lower bound yields hardly any improvement compared with bounds proposed in earlier papers (see Table 2) and I therefore recommend that the authors reduce the praising of their bound in the abstract and the conclusions.

Answer: as the performance of the proposed lower bound is not very attractive, and both Reviewer 1 and Reviewer 2 suggest that it is not the focus of the paper, we have decided to remove the new lower bound from the paper. We would like to present it after further improvement in the future.

The lower bounds in the experiments are substituted by the lower bounds of Bortfeldt and Foster (2012), and the gaps are recomputed accordingly.

Another minor issue is to replace 'receptively' by 'respectively' on page 18.

Answer: we have corrected the error.