UNIVERSITY OF BATH

User's Manual for CLP Spreadsheet Solver

Dr Güneş Erdoğan
School of Management, University of Bath.

Download URL: http://people.bath.ac.uk/ge277/index.php/clp-spreadsheet-solver/
Video tutorial: https://www.youtube.com/watch?v=-0MCkd-CjD0

Developed by Dr. Güneş Erdoğan, 2017.
School of Management, University of Bath.

TABLE OF CONTENTS

1. INTRODUCTION

CLP Spreadsheet Solver is the fourth open-source spreadsheet solver I have developed using Excel and VBA, after VRP Spreadsheet Solver (for vehicle routing problems), FLP Spreadsheet Solver (for facility location problems), and CLP Spreadsheet Solver (for 1D and 2D bin packing problems). All three solvers are available for download at no cost on my personal website.

CLP Spreadsheet Solver is, in broad terms, an extension of BPP Spreadsheet Solver to 3D. Is there a need for it? Academically, yes, since visualizing 3D container loading plans requires a high degree of effort. In practice, container loading planning is usually done using specialized software, and Excel may not be appealing. I am hoping the it will be used for load planning of trucks. Time will tell. Please send your bug reports, comments, and suggestions to G.Erdogan@bath.ac.uk.

2. HOW TO SOLVE A CONTAINER PACKING PROBLEM

CLP Spreadsheet Solver has been designed for simplicity above all. Using the customized ribbon named "CLP Spreadsheet Solver" (in older versions of Mac Office, the commands may be displayed in the "Tools" menu), you may issue the commands in their increasing numerical index, filling in data to each worksheet as it is generated. A step by step guide is given below.

- Enter all relevant data in the "CLP Solver Console" worksheet.
- Execute "1.1 Setup Items Worksheet" and enter the names and dimensions of each item type, as well as the number of items of each type. Do not forget to state which rotations are allowed, and if the item type must be packed, may be packed, or cannot be a packed (this last option is for what-if analysis). If an item type "may be packed", do not forget to give it a positive profit, else it will not be packed.
- Optionally, you may execute "1.2 Optional – Sort items alphabetically" for easier access through the solution worksheet (to be setup later).
- Optionally, you may execute "1.3 Optional – Setup item-item compatibility worksheet". This worksheet, if setup, contains information about which item types can be packed together into a container. For example, you may not want to pack hazardous chemicals and food crates in the same container.
- Execute "2.1 Setup Containers Worksheet", and enter the names, dimensions, and costs of each container type, as well as the number of containers available for each container type. Do not forget to state if a container type may be used or should not be used (this second option is for what-if analysis).
- Optionally, you may execute "2.2 Optional – Sort containers alphabetically" for easier access through the solution worksheet (to be setup later).
- Optionally, you may execute "2.3 Optional – Setup container-item compatibility worksheet". This worksheet, if setup, contains information about which item types can be packed into the container types. For example, you would not pack dairy items into a non-refrigerated container.

- Execute “3. Setup Solution Worksheet”. This may take some time, based on the size of the problem.
- Optionally (and I think you will), you may execute “4.1. Setup Visualization Worksheet”. The items are semi-transparent, so that you can have a better ide of how to place the items.
- “5.1 Engage the CLP Spreadsheet Solver” and wait for the run to end. For best results, allow the solver the time to perform a few thousand iterations at least. As a fair warning, l must state that I have found that the Mac version of Excel is orders of magnitude slower than the Windows version.
- Check the solution and modify it to suit your objectives. Optionally, you can execute “5.2 Optional – Feasibility Check” function to see if the modified solution is still feasible.
- Optionally (and I think you will), you may execute “4.2. Animate the loading plan”. This should give you a good idea about how to actually load items into their locations.
- The “Send feedback / ask a question” command will start an e-mail addressed at me. All (positive) feedback is welcome. Please provide details of your problem when you are asking a question.
- The “Watch tutorial on YouTube” command will direct your web browser to the related video on my YouTube channel. The comments section may contain answers to some of your questions.
- The “About” command will display the current version of the workbook as well as my contact information. Please cite the software and this manual in all projects they have been used.

3. SCOPE AND LIMITATIONS

Among many others, CLP Spreadsheet Solver operates on the following assumptions:

- No safety distance is required between items in a container.
- All items and containers are assumed to be rectangular cubes.
- No items can be prepositioned in a container.
- In addition to the no-overlap constraint and the (optional) compatibility constraints, every item must be supported fully on the surface it is resting.

4. STRUCTURE OF THE WORKSHEETS

CLP Spreadsheet Solver adopts an incremental flow of information, with subsets of data being kept in separate worksheets, as depicted in Figure 1. Initially, the workbook only contains the worksheet named CLP Solver Console. The remaining worksheets should be generated in the sequence denoted by their indices. The names of the worksheets are hardcoded within the code, so you are advised against renaming them. All cell references are absolute, so you are advised against inserting or deleting cells (or columns, or rows) in the worksheets. If a worksheet is modified then the worksheets with a larger index will need to be generated again, and the previous information will be permanently overwritten. If you would like to do a what-if analysis on the parameters, you are strongly advised to make copies of the worksheets before doing so.

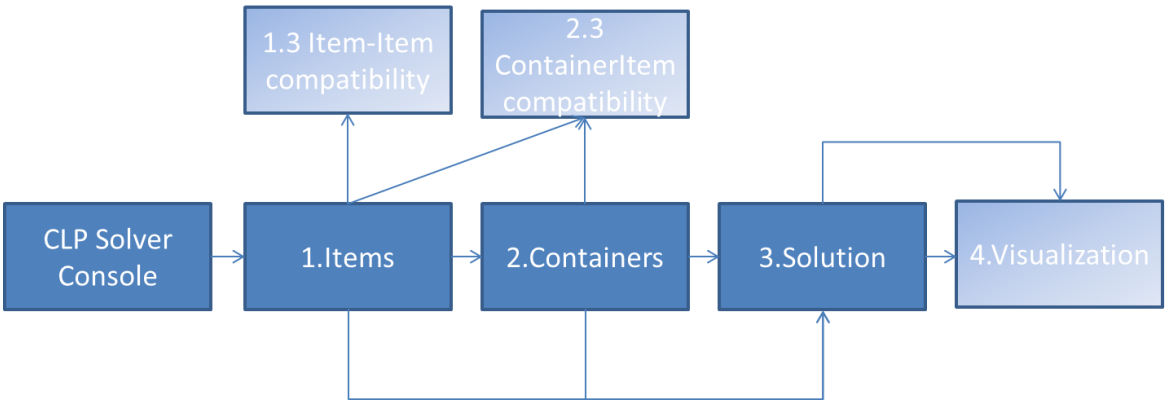


Figure 1: The flow of information between the worksheets. The optional worksheets are denoted with a lighter colour.

The cells containing the data in the worksheets are colour-coded with the following scheme:

The cells with a black background are set by the worksheets and should not be modified.
The cells with a green background are parameters to be set by the user.
The cells with a yellow background are to be computed by the worksheets (but can be edited).
The cells with a blue background will be the result of the solver (but can be manually input).
The cells with an orange background signal a warning.
The cells with a red background signal an error.

Note that some of the colour-coding features are not available for Excel 2007.

4.1 CLP Solver Console

This worksheet is central to the workbook, and **it should not be deleted**. If it is deleted, please run the macro *SetupConsoleWorksheet*, or close and reopen the workbook to generate it again. The parameters defined within the worksheet are described below.

Sequence: Instead of having a wizard interface, which is very easy to use but also very restrictive, the workbook numbers the worksheets in the order of progress. The parameters related to each worksheet are presented along with their sequence number. Please stick to the sequence unless you know what you are doing.

Number of item types: Each item type has a width, height, length, associated profit, and the number of items available. The allowed rotations for the item type, and if the item must be, may be, or cannot be packed may be input later.

Number of types of containers: Each container type has a width, height, length, associated cost, and the number of containers available. You may also input if the container type must be used or cannot be used.

Loading order: This regards the animation to come more than the solution algorithm. The most straightforward option is “Lengthwise”. It does not consider the base support of items, so the animation may look like there will be items hanging on a ledge, only to be saved by an item to be loaded later. The second option is “Diagonal”, which may be easier to follow visually in some instances, but base support is still not considered. The option “Loading ensured” makes sure that each item being placed is properly supported, and no items are blocked. On the other hand, it may result in some counter-intuitive loading orders, in which the type of item being loaded changes unexpectedly.

Item labels: If set to “Yes”, writes the type of the item into the rectangle representing it in the visualization worksheet.

First Fit Decreasing based on: An algorithmic parameter defining the order of items being packed within initial the constructive heuristic. Different settings may result in alternative solutions, especially for low CPU time allowances. The default setting is *Volume*, which works quite well in general. In some instances, *Weight* or *Max{width, height, length}* may result in a better solution.

Show progress on status bar?: If “Yes” is selected, the solution algorithm will display the iteration number and the best solution value obtained in the bottom left of the Excel window. This takes a fraction of a second for every time it is displayed, and may steal the CPU time from the solver. It is better suited for larges instances where fewer iterations will be performed.

CPU time limit (seconds): The amount of time after which the algorithm will end. As a general rule: the longer time you allow, the better the results will be. I recommend at least two seconds per item. Approximately half this time will be spent to ensure that the organization of items within each container is sensible. I recommend a much longer time for Mac Excel (more like a minute per item), due to the mismatch between Microsoft and Apple products.

4.2 1.Items

The columns in this worksheet are explained below.

Item Type ID: This column is automatically generated, and must not be deleted or altered.

Name: The entries in this column must be unique. Being specific at this stage will help you later.

Image file: If you put a jpg file and enter the name here, the cubes will display the image on their visible face.

Width (x): How wide the item is.

Height (y): How tall the item is.

Length (z): How long the item is.

Volume: This field is automatically calculated based on the width, height, and length.

Rotations: All items are allowed to be placed on their x/z surface. The other surfaces are also allowed by default. Here is a figure that will hopefully make things clearer, which shows the default placement on the x/z surface.

Weight: Quite intuitive, this one.

Must be packed? : The item types that "must be packed" will have absolute priority over the items that "may be packed". If an item type "may be packed", make sure that it has a positive profit, otherwise the solver will not attempt to pack it.

Profit: The benefit of packing one item of this type. The objective function is computed as the sum of the profits of the items packed minus the sum of the costs of the containers used.

Number of items: How many items of this type must or may be packed. If you have a type of item for which you have to pack a minimum amount and the rest is optional, I recommend that you split it into two types: the first "must be packed", the other "may be packed".

4.3 2.Containers

The columns in this worksheet are explained below.

Container Type ID: This column is automatically generated, and must not be deleted or altered.

Name: The entries in this column must be unique. Being specific at this stage will help you later.

Width (x): How wide the container type is.

Height (y): How tall the container type is.

Length (z): How long the container type is.

Volume: This field is automatically calculated based on the width and the length.

Weight capacity: Again, quite intuitive.

May be used? : The option of "Do not use" is provided for the possibility of what-if analysis.

Cost: The cost of using one container of this type. The objective function is computed as the sum of the profits of the items packed minus the sum of the costs of the containers used.

Number of containers: How many containers of this type are available.

4.4 3.Solution

For each container a set of columns detailing the items packed into it will be generated. You may scroll right to see the items in the other containers. Column A also contains the *List of detected infeasibilities*, below the items in the first container. The columns in this worksheet are explained below.

Item type name: The name of type of the item. Can be chosen from the drop-down menu populated with the names of the types of items.

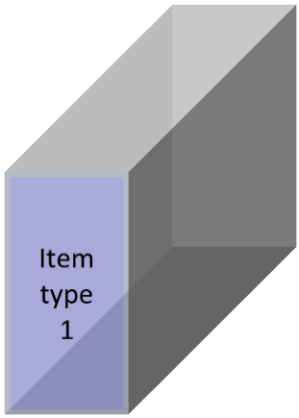
x coordinate: The *x* coordinate for the origin corner of the item.

y coordinate: The *y* coordinate for the origin corner of the item.

z coordinate: The *z* coordinate for the origin corner of the item.

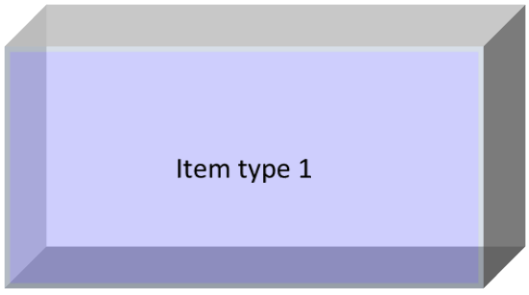
Orientation: The orientation of the item, consisting of the surface it is standing and if it is rotated on that surface. The first and the third letters show the surface the item is placed on, hence the default orientation is xyz. Any other order refers to a rotation where the associated letters imply the type of rotation, e.g. zyx refers to the case where the item is still resting on the x/z surface, with the item is rotated so that the width became the length and vice versa. Examples of all orientations for an item with width 1, height 2, and depth 4 is provided below.

Orientation xyz (default):



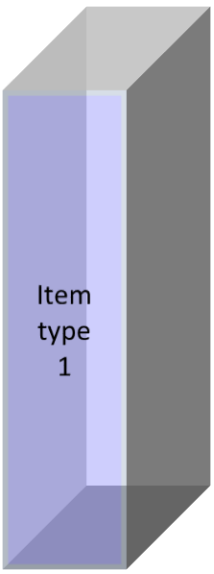
Orientation zyx:

(x and z are swapped, meaning the width and depth are swapped)

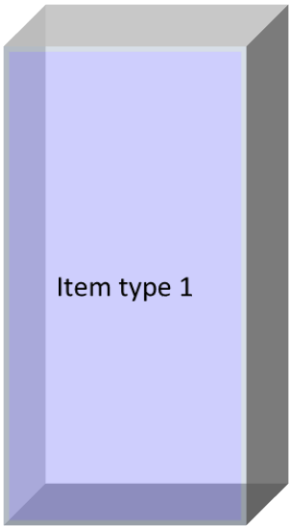


Orientation xzy:

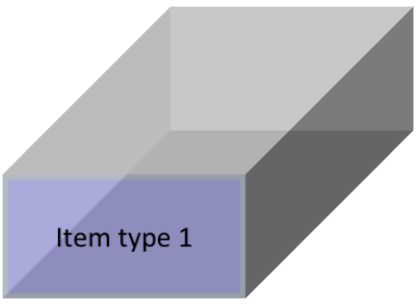
(y and z are swapped, meaning that height and depth are swapped)



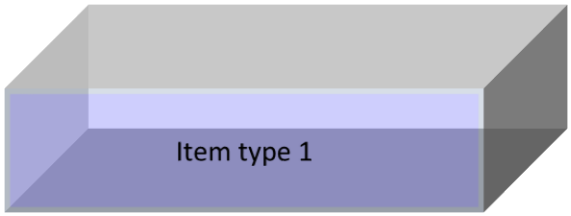
Orientation yzx:



Orientation yxz:



Orientation zxy:



Total volume: The sum of the column *Volume*. In a feasible solution, *Total Volume* should be less than or equal to the area of the container.

Net profit: The sum of the profits of the items in the container minus the cost of the container (if there is at least one item in it).

4.5 4.Visualization

This worksheet is optional, and if generated, it contains rectangular cube shapes showing the containers and the items in the containers. You can move the shapes around to see if you can arrive at a better or better looking solution, but unfortunately the current design will not automatically write your solution into the solution worksheet.

5. FUNCTIONS

5.1 Optional – Reset the workbook

This is a quick way of deleting the data worksheets and resetting the console worksheet. Be careful, it is irreversible. Corresponds to the macro *ResetWorkbook*.

5.2 Setup Items Worksheet

Corresponds to the macro *SetupItemsWorksheet*.

5.3 Optional - Sort items alphabetically

The drop-down menus of the solution worksheet are populated by the list of item types. To have alphabetically ordered drop-down menus, you need to sort the item types. Corresponds to the macro *SortItemTypes*.

5.4 Optional – Setup item-item compatibility worksheet

Two item types are compatible if they can be packed into the same container. If you do not generate this worksheet, the solver will assume that all item types are compatible with each other.

If you generate it, you may choose which item types are incompatible. Corresponds to the macro *SetupItemItemCompatibilityWorksheet*.

5.5 Setup Containers Worksheet

Corresponds to the macro *SetupContainersWorksheet*.

5.6 Optional - Sort containers alphabetically

The containers in the solution and the visualization worksheets will follow the order of the container types in the containers worksheet. In general, it is a good idea to keep them in an alphabetical order. If you have a certain order of containers in your mind, do not use this option. Corresponds to the macro *SortContainerTypes*.

5.7 Optional – Setup container-item compatibility worksheet

A container type is compatible with an item type if the item type can be packed into the container type (e.g. frozen foods and refrigerated / non-refrigerated trucks). If you do not generate this worksheet, the solver will assume that all item types are compatible with all container types. If you generate it, you may choose which item types and container types are incompatible. Corresponds to the macro *SetupContainerItemCompatibilityWorksheet*.

5.8 Setup Solution Worksheet

Corresponds to the macro *SetupSolutionWorksheet*.

5.9 Optional - Setup Visualization Worksheet

Corresponds to the macro *SetupVisualizationWorksheet*.

5.10 Engage CLP Spreadsheet Solver

Corresponds to the macro *CLP_Solver*.

5.11 Optional – Feasibility Check

This function is supplied for checking the feasibility of the data and the solution after manual alterations. Corresponds to the macro *FeasibilityCheckDataAndSolution*.

6. SOLUTION ALGORITHM

The field of CLP research mostly focuses on heuristic algorithms. The details of the literature would be beyond the scope of this document, and we do not claim that a single algorithm can successfully solve all variants of the CLP. Let it suffice to state that a variant of the Large Neighbourhood Search is implemented within the CLP Spreadsheet Solver. An outline of the algorithm is given below.

Step 1 (Initialization): Sort the items with respect to their priority, size, and profit. Sort containers with respect to their size and cost.

Step 2 (Constructive step): Use the First-Fit-Decreasing heuristic to pack the items into the containers.

Step 3 (Perturbation): Randomly remove items from containers, and randomly empty a number of containers. Sort the containers in decreasing order of the volume packed into them.

Step 4 (Reoptimization): Use a constructive heuristic to repack the removed items into the containers.

Step 5 (Solution update): If the new solution is better than the best known solution, update the best known solution. Otherwise, revert back to the best known solution. If the time limit is not exceeded, go to Step 3.

7. CONCLUSION

Despite its shortcomings, I hope that the CLP Spreadsheet Solver will be used as a minor decision support system for small and medium enterprises as well as for teaching and undergraduate and postgraduate levels. I am almost sure that academics will disagree with some design choices, but they can be corrected. I am also quite certain that practitioners will find features that do not completely fit their needs, but it can provide you starting points that lead to an actual solution. Please send bug reports, comments, and suggestions to G.Erdogan@bath.ac.uk.