# User's Manual for BPP Spreadsheet Solver

**Prof Güneş Erdoğan**
**School of Management, University of Bath.**

**Developed by Prof Güneş Erdoğan, 2015.**
**School of Management, University of Bath.**

**DISCLAIMER: THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE ORIGINAL DEVELOPER BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,  WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.**

**TABLE OF CONTENTS**

## 1. INTRODUCTION

In the past two years I have developed two open-source spreadsheet solvers using Excel and VBA: VRP Spreadsheet Solver (for vehicle routing problems) and FLP Spreadsheet Solver (for facility location problems). Both solvers are available for download at no cost on the VeRoLog website. As of the time of this writing, my video tutorial channel for these solvers on YouTube attracts 50 viewers per day on the average. I am receiving about one e-mail per week with a question or a bug report. The total number of downloads per month is more than 100. A number of colleagues in Austria, Canada, Netherlands, and Spain are using the solvers in their lectures. I must admit that this is more than what I have expected.

In the introduction of the user's manual of the FLP Spreadsheet Solver, I have stated that I would be looking into developing a spreadsheet solver for Bin Packing Problems (BPPs). I have realized that there is a need for such a solver, and I have found this solver to be much easier to design with respect to the previous ones, since there was no connection with an external data source. In addition, this solver works both on Windows and Macs, increasing its accessibility. It is capable of solving two-dimensional (and by reduction, one-dimensional) BPPs with up to 50 bins and 200 items. For larger sizes, the solution algorithm still performs reasonably, but setting up the worksheets takes more time than the solution algorithm itself.

At this point, I do not have a clear idea about the next spreadsheet solver, or if there is a need for any other spreadsheet solver. Please send your bug reports, comments, and suggestions to G.Erdogan@bath.ac.uk.

## 2. HOW TO SOLVE A BIN PACKING PROBLEM

BPP Spreadsheet Solver has been designed for simplicity above all. Using the menu item "FLP Spreadsheet Solver" in the tab "Add-ins" (in Macs, the commands are 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. This menu is automatically generated when the file is opened, and deleted when it is closed. If the menu is not available for some reason, you can run the macro *SetupMenuItems* to refresh it. A step by step guide is given below.

- Enter all relevant data in the "BPP 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 if the item type can be rotated, 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 bin.

- Execute "2.1 Setup Bins Worksheet", and enter the names, dimensions, and costs of each bin type, as well as the number of bins available for each item type. Do not forget to state if a bin 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 bins alphabetically" for easier access through the solution worksheet (to be setup later).
- Optionally, you may execute "2.3 Optional – Setup bin-item compatibility worksheet". This worksheet, if setup, contains information about which item types can be packed into the bin types.
- 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. Optional – Setup Visualization Worksheet".
- If you have set up the Visualization worksheet, you may also want to animate it, especially if you are using Guillotine cuts.
- "5.1 Engage the BPP 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, I must state that I have found that the Mac version of Excel is orders of magnitude solver 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.
- 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 "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, BPP Spreadsheet Solver operates on the following assumptions:

- No safety distance is required between items in a bin.
- All items and bins are assumed to be rectangles.
- No items can be prepositioned in a bin.
- There are no constraints other than the no-overlap constraint and the (optional) compatibility constraints.

## 4. STRUCTURE OF THE WORKSHEETS

BPP 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 FLP 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.
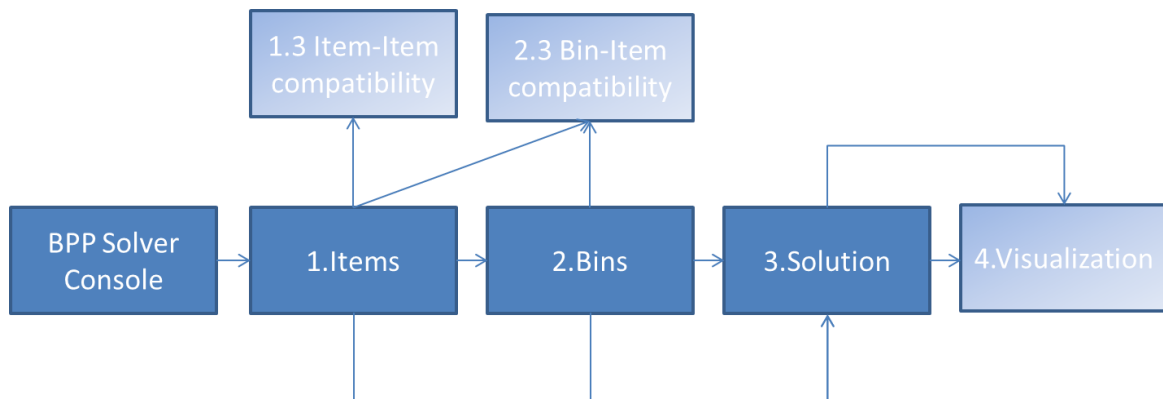


**Figure 1:** The flow of information between the worksheets. The optional worksheets are denoted with a lighter color.

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 BPP 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, length, associated profit, and the number of items available. The option of rotating the item, and if the item must be, may be, or cannot be packed may be input later.

**Number of types of bins:** Each bin type has a width, length, associated cost, and the number of bins available. If the bin type may be used or cannot be used will be required.

**Guillotine cuts?:** This parameter with a colourful name comes from *cutting* rather than packing problems. If the aim of placing the items in the bins is to cut them, and the material or the cutting apparatus cannot stop and turn at the corners, it will need to perform its cuts across the bins. The items to placed later will need to fit into the resulting smaller pieces. An example is depicted below. Enabling the Guillotine cuts will accommodate this cutting / packing constraint.
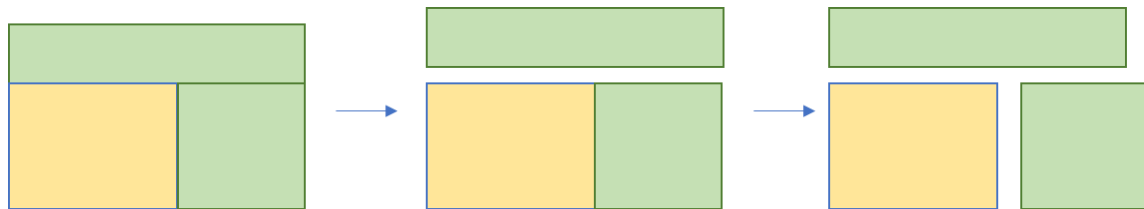


**Item labels:** If set to "Yes", writes the type of the item into the rectangle representing it in the visualization worksheet.

**First Fit Decreasing based on:** An algorithmic parameter defining the order of items being packed within the constructive heuristic. Different settings may result in alternative solutions. The default setting is *Area*, which works well on the instances where most of the items can be rotated. For instances where most of the items cannot be rotated, *Circumference*, *Width* or *Height* may result in a better solution.

**Also minimize distance between identical items?:** If set to "Yes", attempts to minimize the number of identical items being placed in separate bins, and the distance between identical items within each bin. This option may slightly slow down the algorithm due to the distance computation step. The main algorithm is not built for this purpose, and the users are warned not to rely on this option. For effective results with this option, allow more time to the algorithm.

**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 0.5 seconds per item and bin, e.g. for a problem with 20 items and 2 bins, 11 seconds. I recommend a much longer time for Mac Excel, 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.

**Width (x):** How wide the item is.

**Length (y):** How tall the item is.

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

**Can be rotated? :**  If set to "Yes", the solver will look into possibility of rotating the item. It is recommended to set it to "No" whenever applicable to speed up the algorithm.

**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 bins 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.Bins

The columns in this worksheet are explained below.

**Bin 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 bin type is.

**Length (y):** How tall the bin type is.

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

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

**Cost:** The cost of using one bin 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 bins used.

**Number of bins:** How many bins of this type are available.

### 4.4   3.Solution

For each bin a set of columns detailing the items packed into it will be generated. You may scroll right to see the items in the other bins. Column A also contains the *List of detected infeasibilities*, below the items in the first bin. 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 bottom left corner of the item.

**y coordinate:** The *y* coordinate for the bottom left corner of the item.

**Rotated:** If the item is rotated or not.

**Max number of items:** This value is precomputed based on the area of the items and the bins. The idea is to minimize the number of cells required for the solution spreadsheet.

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

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

### 4.5   4.Visualization

This worksheet is optional, and if generated, it contains rectangle shapes showing the bins and the items in the bins. 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 bin. 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 Bins Worksheet

Corresponds to the macro *SetupBinsWorksheet*.

### 5.6 Optional - Sort bins alphabetically

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

### 5.7 Optional – Setup bin-item compatibility worksheet

A bin type is compatible with an item type if the item type can be packed into the bin 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 bin types. If you generate it, you may choose which item types and bin types are incompatible. Corresponds to the macro *SetupBinItemCompatibilityWorksheet*.

### 5.8 Setup Solution Worksheet

Corresponds to the macro *SetupSolutionWorksheet*.

### 5.9 Optional - Setup Visualization Worksheet

Corresponds to the macro *SetupVisualizationWorksheet*.

### 5.10 Optional - Animate Visualization Worksheet

Corresponds to the macro *AnimateVisualizationWorksheet*.

### 5.11 Engage BPP Spreadsheet Solver

Corresponds to the macro *BPP_Solver*.

### 5.12 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 BPP 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 BPP. Let it suffice to state that a variant of the Large Neighbourhood Search is implemented within the BPP 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 bins with respect to their size and cost.

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

**Step 3 (Perturbation):** Randomly remove items from bins, and randomly empty a number of bins. Sort the bins with respect to the area packed into them and their cost per unit area.

**Step 4 (Reoptimization):** Use the First-Fit-Decreasing heuristic to repack the removed items.

**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.   MODIFYING THE INTERFACE

As of version 1.5, BPP Spreadsheet Solver supports the following languages in addition to English: Türkçe (Turkish), Deutsche (German), Espanola (Spanish), Française (French), Italiano (Italian), Português (Portuguese), and عربى (Arabic). I do not speak all these languages, so the translations are by Google Translate. I expect that there are some horrible mistranslations, which I hope will be corrected by the help of altruistic users. However, if you are in a hurry to correct your own copy or add a new language, here is how to do it.

There are two hidden worksheets, MultiLanguageMenuItems and MultiLanguageInterface. The names are self-explanatory. You can unhide them one at a time by right clicking the name tab of CLP Solver Console worksheet, clicking "Unhide…", selecting the worksheet to be unhidden, and clicking OK. In both worksheets, the parts that you should not change are marked with a black background, and the parts you can modify are marked with a green background, as per the general colour scheme of the solver. In short, the first columns of the former and the first column of the latter **must not be modified**.
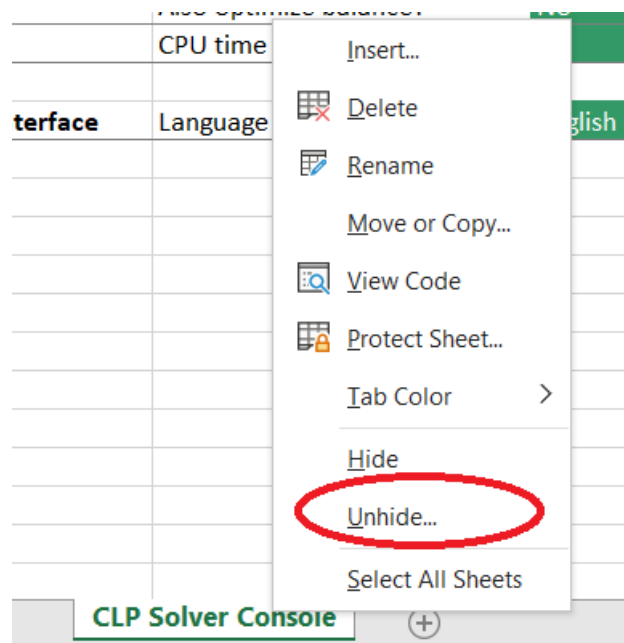


Simply changing the part of the interface in these worksheets will get you what you need. The message boxes will respond immediately. For the menu items to respond, you may need to save,

close, and reopen the workbook. For the worksheets to respond, you will need to set them up again. The modifications are best done when there is no data in the solver.

If you need to add a new language, you will need to replace one of the existing languages. Pick a language that you will never use and modify the header of the column (in both worksheets) to the name of the language. Then, modify the contents of the new language column as appropriate. Google Translate can handle 100 to 150 entries at one time. Adding I suggest that you hide these worksheets once you are done, by right clicking their name tabs and clicking "Hide".

If you need to modify the English interface so that you can implement your own jargon, follow the steps described in the previous paragraph. Modify the header of the column (in both worksheets) to a meaningful name that is different than English, e.g. MyCompany. Then, modify the contents of the MyCompany column (in both worksheets) to your heart's desire.

## 8. CONCLUSION

Despite its shortcomings, I hope that the BPP 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.