

Excel for Two-Sided Matching:
A Users' Manual (short version 1)*

*The original version of this note was written by Shuya Abe (Graduate School of Information Science, Osaka University) and Naoki Watanabe (Graduate School of Business Administration, Keio University). The authors thank Jun Wako (Faculty of Economics, Gakushuin University) for his advice on software improvement.

1 Introduction

This is a users' manual (short version) for *Excel for Two-Sided Matching* to solve two-sided matching problems with a computer. As of September 2017, *Excel for Two-Sided Matching* version 3 (sample3.xlsm) is available from the following URL.

<http://labs.kbs.keio.ac.jp/naoki50lab/sample3.xlsm>

The screen-shots in this manual were taken in the environment shown in Table 1.

Table 1: Environment of screen-shots

OS	Windows 10 Enterprise
Excel	Microsoft Office 2016 Excel
CPU	Intel(R) Core(TM) i7-3770K CPU @ 3.50GHz 3.90GHz
Memory	8 GB
Disk	1 TB

Excel for Two-Sided Matching runs on computers that can use Excel VBA (Visual Basic for Applications). The recommended environment is listed in Table 2.

Table 2: Recommended environment

OS	Windows 7–, Mac OS X 10.8–
Excel	Microsoft Office 2013–, Microsoft Office for Mac 2016
CPU	no requirement if Excel works
Memory	at least 2GB
Disk	no in particular

Notes

- (1) *Excel for Two-Sided Matching* consists of 4 Excel sheets (man, group, conf, result) and 1 Visual Basic for Applications (VBA) program. If assistance is needed for reading and writing VBA codes, refer to an appropriate textbook.

- (2) Two-sided matching is an assignment problem. Matching workers and divisions in a company is an example, and the following information is used.
 - (a) Workers and divisions in a company: each worker has a preference over the divisions to which he or she is to be assigned, and each division conducts evaluations of the workers. The preferences and evaluations are represented by ranking without ties.
 - (b) Quota: the maximal number of workers that can be accepted by each division, i.e., capacity.
- (3) Users need to input numerical values in **Arabic numerals (not double-byte characters) without multiple numbers and without missing values** for initial settings, workers' preferences, divisions' evaluations, and quotas of each division.
- (4) *Excel for Two-Sided Matching* uses a program written in Visual Basic. Thus, unlike editing a regular Excel sheet, users cannot restore states by pressing the “backspace” key or pressing the “Z” key while holding down the “Ctrl” key. It is possible to return the Excel file to the intended state before editing by closing the Excel file without saving the current state.
- (5) The computation results are overwritten each time the program is run. Thus, users need to make a copy of the computation results in another file.

2 Using *Excel for Two-Sided Matching*

2.1 Getting Started

Initially, set the numbers of workers and divisions in the VBA window. See Fig. 1. By default, the number of workers is set as 107 and the number of divisions is set as 11.

1. Click on the **Visual Basic** button on the **Developer** ribbon.
2. Find **VBAProject (sample3.xlsm)** by **Project** explorer.
3. Open **Module 1** in the **Standard Module** that is under **VBAProject (sample3.xlsm)**.

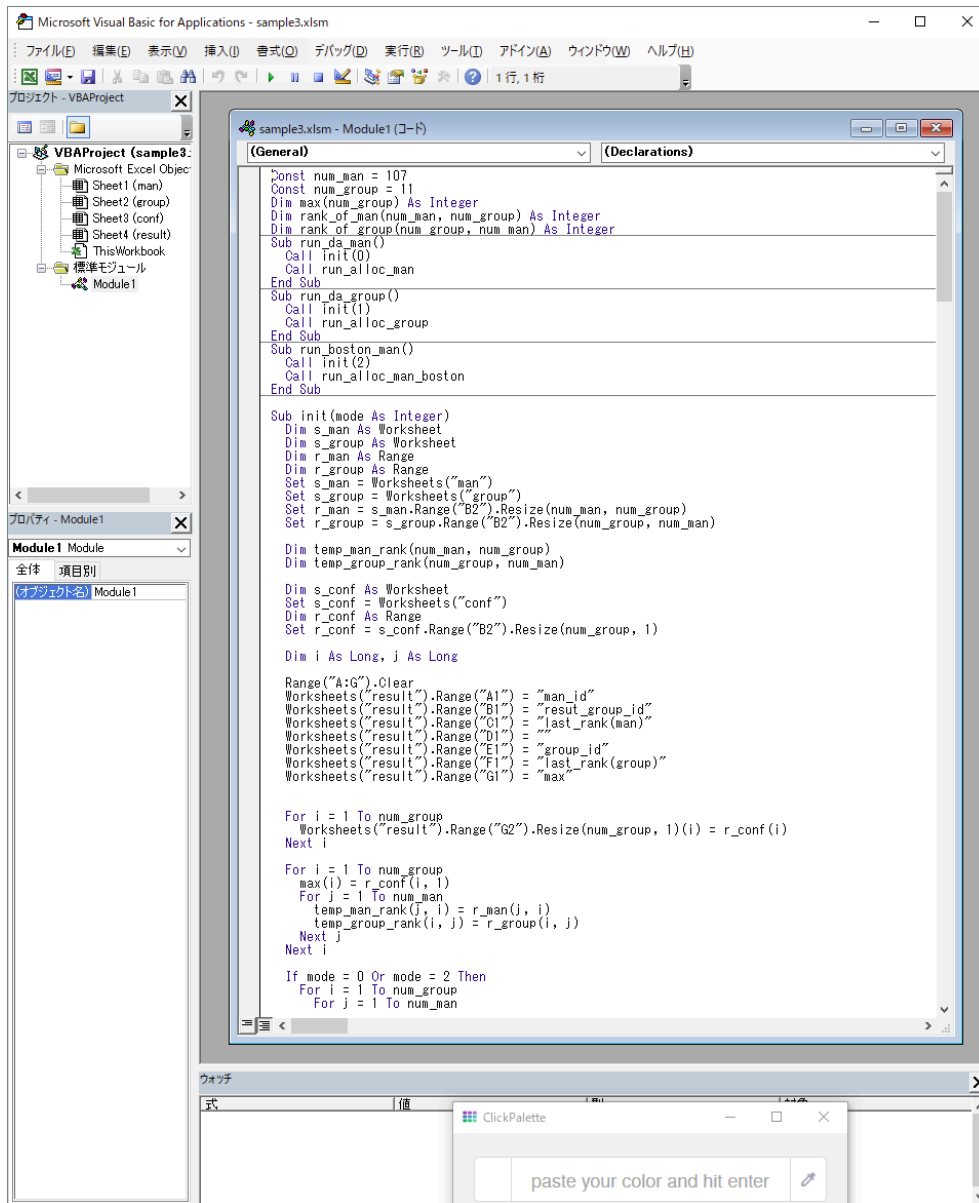


Figure 1: Visual Basic window

4. Change the numbers in `Const num_man` to set the number of workers, and change the numbers in `Const num_group` to set the numbers of divisions. These are the first two lines in **Module 1**.
5. Save to the same file and close the Visual Basic window.

2.2 Running the Program

1. Complete the initial setting for the numbers of workers and divisions in the Visual Basic window.
2. Input numerical data on workers' preferences, divisions' evaluation, and quotas in the (1) man sheet, (2) group sheet, and (3) conf sheet.
3. Open the (4) result sheet.
4. Choose one button out of three in the result sheet.
 - Click on the **da (man)** button for computing a matching with the worker-proposing DA algorithm.
 - Click on the **da (group)** button for computing a matching with the division-proposing DA algorithm.
 - Click on the **boston (man)** button for computing a matching with the worker-proposing DA algorithm.
5. View computation results in the result sheet.

(1) man sheet

Input workers' preferences over divisions in Arabic numerals. See Fig.2, where 20 workers all report their preferences over 5 divisions.

Rows in column A (man_id): Input worker IDs in ascending order **without multiple numbers and without missing values.**

Rows in column B and thereafter (g_rank): Each row is used for a worker.

Input the division ID worker i prefers at the j th rank in the cell corresponding to row (man_id) i and column (g_rank) j . (In Fig.2, the second-best division (g_rank2) for a worker with ID=5 (man_id=5) is a division with ID=4.) The value put in cell 6C is thus 4.

randomize button: Use this to generate the workers' preferences randomly.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	man_id	g_rank1	g_rank2	g_rank3	g_rank4	g_rank5							
2	1	2	3	5	1	4							
3	2	1	4	3	5	2							
4	3	4	5	3	1	2							
5	4	1	3	4	5	2		randomize					
6	5	2	4	5	1	3							
7	6	2	4	3	5	1							
8	7	1	2	5	3	4							
9	8	5	4	1	2	3							
10	9	4	1	3	5	2							
11	10	5	1	2	3	4							
12	11	3	1	5	2	4							
13	12	1	5	3	4	2							
14	13	4	2	5	1	3							
15	14	2	4	5	1	3							
16	15	1	5	3	2	4							
17	16	4	2	1	3	5							
18	17	4	3	5	1	2							
19	18	5	2	4	1	3							
20	19	3	5	1	4	2							
21	20	5	3	1	2	4							
22													

Figure 2: man sheet

(2) group sheet

Input divisions' evaluations for workers in Arabic numerals. See Fig.3, where 5 divisions report their evaluations for 20 workers (although the 13th ranked worker and thereafter cannot be seen in the screen-shot).

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	group_id	m_rank1	m_rank2	m_rank3	m_rank4	m_rank5	m_rank6	m_rank7	m_rank8	m_rank9	m_rank10	m_rank11	m_rank12
2	1	12	16	2	8	17	14	4	5	6	13	18	20
3	2	18	7	6	10	3	15	4	16	17	20	11	5
4	3	12	3	11	9	5	6	18	10	4	15	1	13
5	4	7	14	20	12	9	2	1	19	17	10	13	6
6	5	17	8	13	6	5	19	11	4	14	18	20	15
7													
8													
9													
10													
11													
12													
13													
14													
15		randomize											
16													
17													
18													
19													
20													
21													
22													

Figure 3: group sheet

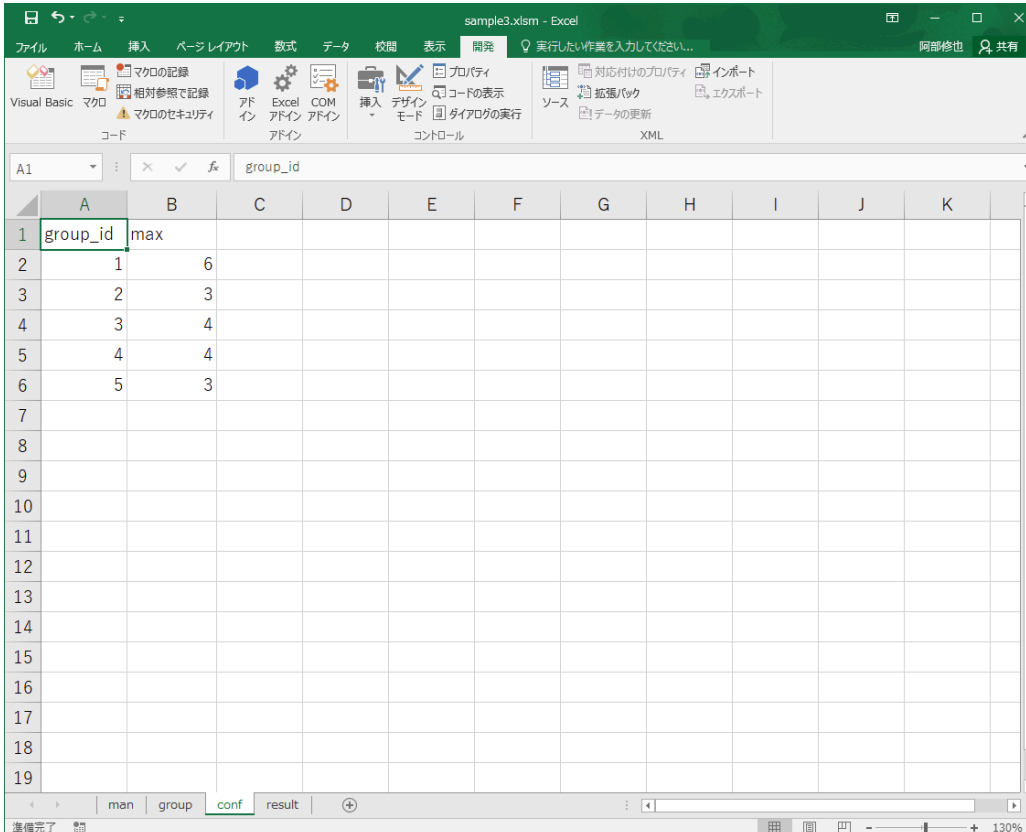
Rows in column A (group_id): Input worker IDs in ascending order **without multiple numbers and without missing values.**

Rows in column B and thereafter (m_rank): Each row is used for a division. Input the worker ID division i evaluates at the j th rank in the cell corresponding to row (group_id) i and column (man_rank) j . (In Fig.2, the fourth-best worker (man_rank4) for a division with ID=3 (group_id=3) is a worker with ID=9.) The value put in cell 4E is thus 9.

randomize button: Use this to generate the workers' preferences randomly.

(3) conf sheet

Input quotas of divisions in Arabic numerals. See Fig.3, where 5 divisions report 20 openings in total.



	A	B	C	D	E	F	G	H	I	J	K
1	group_id	max									
2	1	6									
3	2	3									
4	3	4									
5	4	4									
6	5	3									
7											
8											
9											
10											
11											
12											
13											
14											
15											
16											
17											
18											
19											

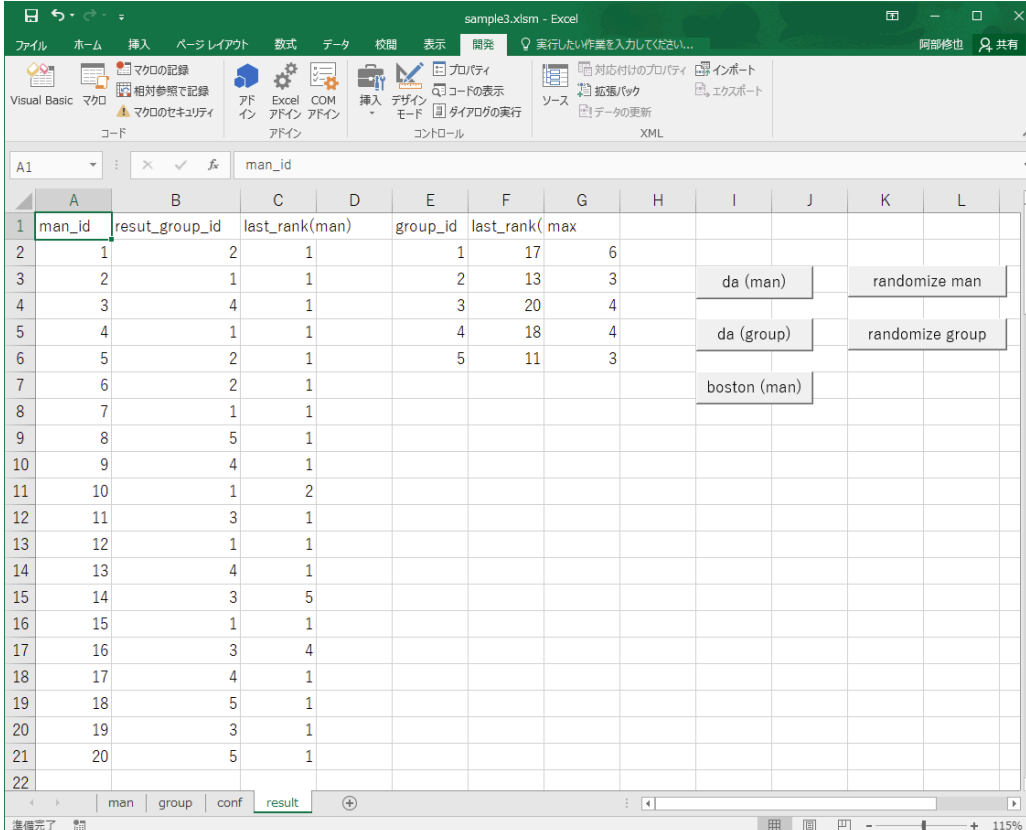
Figure 4: conf sheet

Rows in column A (group_id): Input worker IDs in ascending order **without multiple numbers and without missing values.**

Rows in column B (max): Each row is used for a division. Input the quota of division i in the cell corresponding to “max” in row (group_id). **The total of all quotas should coincide with the total number of workers.** In Fig.4, the quota of a division with ID=1 is 6. The value put in cell 2B is thus 3.

(4) result sheet

The computation results are shown in the result sheet. The buttons in this sheet are used to run each computation. See Fig.5, which shows a matching of 20 workers to 5 divisions by rankings according to the workers' preferences and the divisions' evaluations.



man_id	resut_group_id	last_rank(man)	group_id	last_rank(max	
1	2	1	1	17	6
2	1	1	2	13	3
3	3	1	3	20	4
4	4	1	4	18	4
5	5	2	5	11	3
6	6	2	1		
7	7	1	1		
8	8	5	1		
9	9	4	1		
10	10	1	2		
11	11	3	1		
12	12	1	1		
13	13	4	1		
14	14	3	5		
15	15	1	1		
16	16	3	4		
17	17	4	1		
18	18	5	1		
19	19	3	1		
20	20	5	1		
21					
22					

Figure 5: result sheet

Rows in column A (man_id): Worker IDs are shown in ascending order.

Rows in column B (result_group_id): Each division matching each worker is shown with the ID.

Rows in column C (last_rank (man)): The rank of the division that matches a worker in his or her preference is shown for each worker. In Fig. 5, worker 14 is matched with division 3, which is the 5th rank of his or her preference. The value put in cell 15C is thus 5.

Rows in column E (group_id): Division IDs are shown in ascending order.

Rows in column F (last_rank (group)): The lowest rank of worker among workers matched with a division according to the division's evaluation is given. As shown in the middle of Fig.5, division 3 with quota 4 is matched with the worker who is 20th in the evaluation of the division. In division 3, no worker is lower ranked in the evaluation. The value put in cell 4F is thus 20.

Row in column G (max): The quotas put in conf sheet are shown.

(**Caveat**) When the workers' preferences have missing values, an error message appears if worker-proposing algorithms are used, whereas the division-proposing DA algorithms may generate a matching. In the latter case, **it should be an error message if last_rank (man)=0**. The same thing applies to the worker-proposing algorithms when the divisions' evaluations have missing values.

da (man)button: Press when worker-proposing DA algorithm is used.

da (group) button: Press when division-proposing DA algorithm is used.

boston (man) button: Press when worker-proposing Boston mechanism is used.

randomize man button: The same function button that appears in the man sheet.

randomize group button: The same function button that appears in the group sheet.

3 Troubleshooting

3.1 Developer ribbon does not appear

In the default, Microsoft Office Excel does not show the **Developer** ribbon. Complete the following procedure (in the case of Microsoft Office Excel 2016).

1. Click on the **File** ribbon.
2. Click on the **Option** ribbon.
3. Click on the **Customize Ribbons** in **Excel's Option** window.
4. Confirm whether **Main Tab** appears in the **Customize Ribbons** field, and then look for **Developer** in the list of ribbons in **Main Tab**.
5. Click on the **OK** button.
6. Close the **Excel's Option** window, and then the **Developer** button appears. If the button does not, then open the Excel file again.

3.2 Error messages appear

Confirm the following points, when error messages appear after pressing the buttons to run the program.

- Did you input data in Arabic numerals with single-byte characters in the VBA window, man sheet, group sheet, and conf sheet?
- Do the numbers of workers and divisions initially set in the VBA window coincide with those put in man sheet and group sheet?
- Does the number of workers initially set in the VBA window coincide with the total of all quotas put in conf sheet?

3.3 No error message but incorrect matching

Confirm the following points, when error messages appear but the software does not generate a correct matching after pressing the buttons to run the program.

- Did you open your Excel file in Protected View (Fig.6)? In that case, click on the **To Enable Editing** button to end Protected View.
- Does the total of all quotas coincide with the total number of workers?

- Are there any multiple numbers or missing values in man sheet, group sheet, or conf sheet?

	A	B	C	D	E	F	G	H	I	J	K	L
1	man_id	resut_group_id	last_rank(man)		group_id	last_rank(max)						
2	1	2	2		1	12	6					
3	2	4	1		2	18	3		da (man)		randomize man	
4	3	3	3		3	8	4					
5	4	1	1		4	6	4		da (group)		randomize group	
6	5	3	1		5	16	3					
7	6	5	1						boston (man)			
8	7	4	3									
9	8	2	1									
10	9	4	2									
11	10	3	1									
12	11	3	1									
13	12	5	1									
14	13	1	1									
15	14	4	1									
16	15	2	2									
17	16	1	1									
18	17	1	3									
19	18	1	2									
20	19	5	1									
21	20	1	1									
22												
23												
24												

Figure 6: Screen-shot of Protected View.