A NEW APPROACH TO REDUCE THE BURDEN OF PAIRWISE-COMPARISON ON THE AHP

SangHeon Han * Tsutomu Hiroshima † Toru Yarita ‡

Received April 18, 2017 ; revised May 15, 2017, July 4, 2017

Abstract. The pairwise comparisons in AHP (Analytic Hierarchy Process) are made using a scale list that indicates the importance of one entity over another entity with respect to a given criteria. Moreover, the pairwise comparison matrix represents the intensities of the decision maker's preference between individual pairs of alternatives. The matrix is usually determined from the 1-point to 9-point scale. Various methods for paired comparison method have been proposed, making more intuitive and highly accurate decision making possible. However, the number of pairwise comparisons increases as the number of criteria increases. Therefore, the burden of decision makers would become heavier.

In this paper, we propose an algorithm for the allocation problem of the burden and verify the algorithm by using a programming language called Haskell[1], which is specialized in the functional programming. This research contributes not only to allocation algorithm, but also aids researchers and decision makers in applying the AHPs effectively.

1 Introduction Pairwise Comparison Method is always used in comparing entities in pairs, whereby there are more than two entities to evaluate. Because the process of evaluating is very simple, the pairwise comparison has been broadly used in studies of preference, precedence and social choice etc.. Methods which utilize pairwise comparison focus on superiority/inferiority between entities or its differences. Pairwise Comparison Method is proposed as a typical method in AHP. In 1971, Thomas L. Saaty proposed AHP as a supporting tool for decision making to connect human's subjective evaluation with reasonable decision. In AHP, decision maker regards a problem as hierarchy relation of criteria /alternatives levels for systematic approaches to problem. As a result, AHP is useful in multiple criteria problem and in subjective decision-making problem. In real, AHP has been broadly applied to decision making in business or to consensus building in public works, etc.. However, AHP uses a great number of pairwise comparisons to calculate numerical weights of criteria and alternatives in general. While pairwise comparison methods have merits of simple process and accurate evaluation, many studies have indicated demerits that decision maker's comparing burden is heavy if comparing entities increase in Cowan (2001). There exists similar demerit in AHP because AHP needs to compare with many pairs of criteria if the number of criteria increases for more exact evaluation. Furthermore, if the number

2010 *Mathematics Subject Classification.* 91A10, 91A40, 91A50, 91A65.

Key words and phrases. Analytic Hierarchy Process (AHP), Combinatorial design method, Pairwise comparison .

*Department of Management Administration, Nagoya University of Commerce & Business, 4-4 Sagamine, Komenoki-cho, Nisshin-shi, Aichi-ken, Japan 470-0193, Phone: +81-(0)561-73-2111, E-mail: han-sh@nucba.ac.jp

†Department of Management Administration, Nagoya University of Commerce & Business, 4-4 Sagamine, Komenoki-cho, Nisshin-shi, Aichi-ken, Japan 470-0193, Phone: +81-(0)561-73-2111, E-mail: hiroshima@nucba.ac.jp

‡Department of Economics, Nagoya University of Commerce & Business, 4-4 Sagamine, Komenoki-cho, Nisshin-shi, Aichi-ken, Japan 470-0193, Phone: +81-(0)561-73-2111, E-mail: yarita@nucba.ac.jp

[1]Haskell is a relatively new language that was born in 1990 (Haskell 1.0) as a "standard word" in purely functional programming language based on lazy evaluation. In 1999, as a stable language definition Haskell 98 was enacted, and gradually became popular.

of alternatives increases then it is necessary to compare with many pairs of alternatives for each criterion at a higher level. Han (2014) pointed out the heavy burden of pairwise comparison in AHP and proposed a method to decrease the value of C.I..

Therefore, in this study we concentrate on the problem of sharing the heavy burden of pairwise comparison work with multiple decision makers under the assumption of making consensus building. The problem is transformed into group AHP problem, which is a necessary consensus building process to solve the problem.

Next, we outline the configuration of the study. Section 2 describes the background and the objective of the study and treat of past studies. In section 3, we describe basic theory related to burden sharing of pairwise comparison works. Section 4 considers fair sharing burden problem and propose efficient solution procedure. Section 5 discusses the conclusion of the study and subjects in future.

2 Review of Related study During last three decades, many attempts were made in directing investors so that AHP and Fuzzy are proposed as tools for portfolio optimization for financial decision making.

Saaty (1980) and Durer et al. (1994) analyzed a complex portfolio system. Furthermore, to select securities using data envelopment analysis (DEA), Tiryaki (2001) evaluate the financial performance of companies. Bolster et al. (1995) used AHP to determine investor suitability, based on age, to select among seven investment securities. The results of their study showed varying patterns of investment for the different age groups.

Thalheimer and Ali (1979) applied AHP to time series analysis and to portfolio selection on mutual savings banks to determine the optimal portfolio choice. They opined that an investor should invest in short-term stocks, risk-less assets, and well diversified investment to achieve the highest utility from investable fund. Oyatoye et al. (2010) applied AHP to investment portfolio selection in the banking sector. They underlines the importance of different criteria, factors and alternatives that are essential to successful investment decisions in the financial crisis in 2008. We find that each criterion plays an important role in the portfolio. Nevertheless, the effect of the AHP-portfolio on the total returns is trivial. Therefore, we determine that the decision maker prefers the government bond to the others. We also determine that the portfolio generated by AHP accords with the decision maker's preference.

3 Background and objective of study For proper number of criteria and alternatives in case single decision maker, concept of magical number is introduced to be 7 ± 2, as suggested by Miller (1956).

After that, magical number of 5 to 9 has been changed to number of less than equal to 4 in order to satisfy transitive law as possible in Cowan (2001). In short, proper number of criteria and alternatives is 3 or 4 in AHP for one decision maker. The number is regarded as physical limitation of decision maker in this study. Let's consider an interview test in entrance examination as an example. If the number of candidates for ranking increases then candidates will be split into multiple interviewers due to the physical limitation of interviewer. The ranking is decided by consensus building in meeting. In the above example, interview time is limited by physical situation of interviewer for one group of candidates. If proper interview time is given then the necessary numbers of interviewers and interview rooms are fixed.

In AHP pairwise comparison, decision maker continues to evaluate by referring to scale list. For example, when he/she considers comparison between alternative A and B, if A is more important than B then he/she evaluates as corresponding integer from 1 to 9 referred to scale list. In the contrary case, he/she evaluate as corresponding fraction from 1/1 to 1/9.

This procedure has different complexity of evaluating from questionaries' format. In AHP pairwise comparison, evaluating value is easy to lead into confusion, which results from violation of transitive law. It is necessary to investigate efficient method such as hybrid type of using interval and precedence criteria, simultaneously.

In this study, we suppose a decision maker's physical constraints and focus on sharing heavy burden of AHP pairwise comparison work with multiple decision makers for improving process of pairwise comparison. The objective of this study is to propose an efficient solution procedure to support AHP pairwise comparison with decision maker's physical limitation and to investigate what pairwise comparisons should be.

4 Pairwise comparison and combinatorial design theory In this section, we consider a wine-tasting problem as a series of allocation problem. There are 7 wines of different brands, such as A, B, C, D, E, F, G which are evaluated based on some criteria by an expert sommelier.

In addition, we suppose a physical constraint of sommelier that they can test only 3 brands. That is, one sommelier can compare in three combinatorial pairs from three types of wines. The number of all pairs is from seven types of wine. Because one sommelier can test three pairs of wines, it is necessary to split all pairwise comparisons into seven sommeliers. All wines are named by A, B, ..., G and only three types of wines must be assigned to each sommelier. How to assign all types of wines to seven sommeliers fairly? The above problem has been studied in the design theory and some methods are proposed such as cyclic design method. As a result of utilizing cyclic design method, optimal design is obtained as follows;

$$\{\mathrm{A,B,C}\},\{\mathrm{A,D,E}\},\{\mathrm{A,F,G}\},\{\mathrm{B,D,F}\},\{\mathrm{B,E,G}\},\{\mathrm{C,D,G}\},\{\mathrm{C,E,F}\}.$$

In Table 1, we illustrate the above optimal design as matrix, which called incidence matrix. Titles of row and column are wine and sommelier in matrix, respectively. If there is a design then 1 is assigned, otherwise 0. As a result, we obtain 7×7 matrix in Table 1.

Table 1: Incidence matrix of wine and sommelier

wine	1	2	3	4	5	6	7
wine A	1	1	1	0	0	0	0
wine B	1	0	0	1	1	0	0
wine C	1	0	0	0	0	1	1
wine D	0	1	0	1	0	1	0
wine E	0	1	0	0	1	0	1
wine F	0	0	1	1	0	0	1
wine G	0	0	1	0	1	1	0

In incidence matrix, there are three 1 in any column and it means that any sommelier test three brands of wines. And there are three 1 in any row and it means that any brand of wines tested by three sommeliers. Furthermore, we know that the matrix reflects all pairs of wine and sommelier. Partially, the combinatorial design theory traces its origin to recreational mathematics in middle of 19th century. However, the theory has been activated and developed by the design of experiments (DOE). It is the start of DOE that R. A. Fisher studied an agricultural test in Rothamsted in 1919. He introduced terms of plot, treatment, block and variety to his experiment. These terms and related symbols due to R. A. Fisher are used as they stand in design theory. For example, symbol of treatment for each plot is used to denote variety in DOE, which was suggested by Fisher (1937) and Ishii (1972).

Now we consider representation of design by symbol. The number of treatments is denoted by v. In the example of wine test, the number of wine brands is denoted by $v = 7$ and number of sommeliers is $b = 7$. Next, r denotes the number of blocks for each treatment and $r = 3$ in the wine test problem. Symbol k denotes the number of plots for each block and $k = 3$ in the wine test problem. Finally, λ denotes the number of blocks in the case there are any treatment pairs and $\lambda = 1$ in the wine test problem. (Table 2)

Table 2: Symbols in wine test problem

	plot (k)	treatment (v)	block (b)	r	λ
wine test	Sommelier's Constraints	Number of wines	Number of sommeliers	Number of Sommeliers on each wine	One testing
	3	7	7	3	1

Under the above setting of notation, (v, b, r, k, λ) is used to describe combinatorial design problems suggested by Mazur (2010). Note that v, b, r, k must satisfy the following equation:

$$ {}_v\mathrm{C}_2 = bk = vr. \tag{1} $$

For instances, wine test problem is expressed by $(7, 7, 3, 3, 1)$ design. The above design problem is efficiently utilized to sharing heavy burden of comparing work to multiple sommeliers for the AHP problem constrained by physical limitation of decision maker.

5 Fair sharing problem of burden Suppose the physical limitation k and the number of wines v are given, the required number of sommeliers b and the number of blocks of each wine r are determined by (1). Therefore, there exists a fair sharing if and only if

$$ \frac{{}_v\mathrm{C}_2}{k} \text{ is integer and } v \text{ is odd.} \tag{2} $$

In this study, we suppose $k = 3$, that is, each decision maker can cover with only 3 entities because of physical limitation. The assumption is based on a minimum number of entities for testing exactly and there are 3 combinatorial pairwise comparisons. For example, if wines of $\{\mathrm{A}, \mathrm{B}, \mathrm{C}\}$ are assigned then decision maker test three pairwise comparison of (A, B), (B, C) and (A, C). However, there is no meaning to share comparison work when decision maker test only two wines of $\{\mathrm{A}, \mathrm{B}\}$ because there is only one pairwise comparison.

In general, accuracy of AHP pairwise comparison is calculated based on CI (Consistency Index) and Han (2014) pointed out that the greater the number of options, the greater the burden of decision-makers by pairwise comparisons.

Table 3 considers wine test problem with 7 wines ($v = 7$) and 3 physical limitations ($k = 3$). There are ${}_v\mathrm{C}_2 = 21$ pairwise comparisons for each wine and design of assigning 3 wines to 7 sommeliers ($b = 7$). In short, sommelier 1 tests 3 wines of $\{\mathrm{A}, \mathrm{B}, \mathrm{C}\}$ and compares with 3 pairs of (A, B), (B, C) and (A, C). Figure 4 describes wine test problem with 9 wines ($v = 9$) and 3 physical limitations ($k = 3$). In case 8 wines ($v = 8$) there are all pairwise comparisons of ${}_v\mathrm{C}_2 = \frac{8\times 7}{1\times 2} = 28$ and necessary number of sommeliers is obtained $b = 28/3 = 9.33\ldots$ from the assumption of physical limitation. Therefore, there is no optimal design because the number of sommeliers is not positive integer.

Now Table 4 describes necessary number of sommeliers and the existence of optimal design based on situation with 9 wines ($v = 9$) and 3 physical limitations ($k = 3$).

Table 5 describes the judgements of existence of design. Column 6, 7 and 8 describe whether the condition (2) is satified.

Table 3: All pairwise comparisons and optimal design for fair sharing — Example of 7 wines constrained to 3 physical limitations

all pairwise comparisons					
AB	BC	CD	DE	EF	FG
AC	BD	CE	DF	EG	
AD	BE	CF	DG		
AE	BF	CG			
AG	BG				
AF					

assignment
sommelier-1:$\{A, B, C\}$
sommelier-2:$\{A, D, E\}$
sommelier-3:$\{A, F, G\}$
sommelier-4:$\{B, D, F\}$
sommelier-5:$\{B, E, G\}$
sommelier-6:$\{C, D, G\}$
sommelier-7:$\{C, E, F\}$

Table 4: All pairwise comparisons and optimal design for fair sharing — Example of 9 wines constrained to 3 physical limitations

all pairwise comparisons							
AB	BC	CD	DE	EF	FG	GH	HI
AC	BD	CE	DF	EG	FH	GI	
AD	BE	CF	DG	EH	FI		
AE	BF	CG	DH	EI			
AF	BG	CH	DI				
AG	BH	CI					
AH	BI						
AI							

assignment
Sommelier-1 :$\{A, B, C\}$
Sommelier-2 :$\{A, D, E\}$
Sommelier-3 :$\{A, F, G\}$
Sommelier-4 :$\{A, H, I\}$
Sommelier-5 :$\{B, D, F\}$
Sommelier-6 :$\{B, E, H\}$
Sommelier-7 :$\{B, G, I\}$
Sommelier-8 :$\{C, D, I\}$
Sommelier-9 :$\{C, E, G\}$
Sommelier-10:$\{C, F, H\}$
Sommelier-11:$\{D, G, H\}$
Sommelier-12:$\{E, F, I\}$

In Figure 1, we list up the optimal design by cyclic method after judging the existence of design. For detail cyclic method, refer to note of Rosa (1991).

In this study we consider an AHP problem to rank wines. The AHP problem is illustrated by AHP chart in Figure 1. For simplicity, we suppose that there are wines of less than 10 and one decision maker (sommelier, appraiser) has physical limitation of 3 wines for testing. For the problem, we propose an efficient algorithm to share heavy testing burden fairly with multiple tester (sub-sommeliers) based on the combinatorial design theory. It is necessary to investigate the existence of optimal design and construction of the design for fair burden sharing. The above AHP problem is revised by add sub-sommeliers layer such as Figure 1.

Solution procedure of fair sharing problem

- Step1: Set the wine test problem to classic AHP chart.
- Step2: Define a physical limitation of decision maker
- Step3: Find necessary number of sub decision makers
- Step4: Apply our algorithm to find optimal design assigning each wine to each sub decision maker, fairly.

Table 5: Judgement of existence of design

no. of pairwise comparison (t)	required no. of sommeliers (b)	Physical limitation (k)	no. of wines (v)	no. of blocks of each wine (r)	Final Judgement (Fairness Condition)
3	1	3	3	1	Passed
6	2	3	4	1.5	Failure
10	3.33	3	5	2	Failure
15	5	3	6	2.5	Failure
21	7	3	7	3	Passed
28	9.33	3	8	3.5	Failure
36	12	3	9	4	Passed
45	15	3	10	4.5	Failure

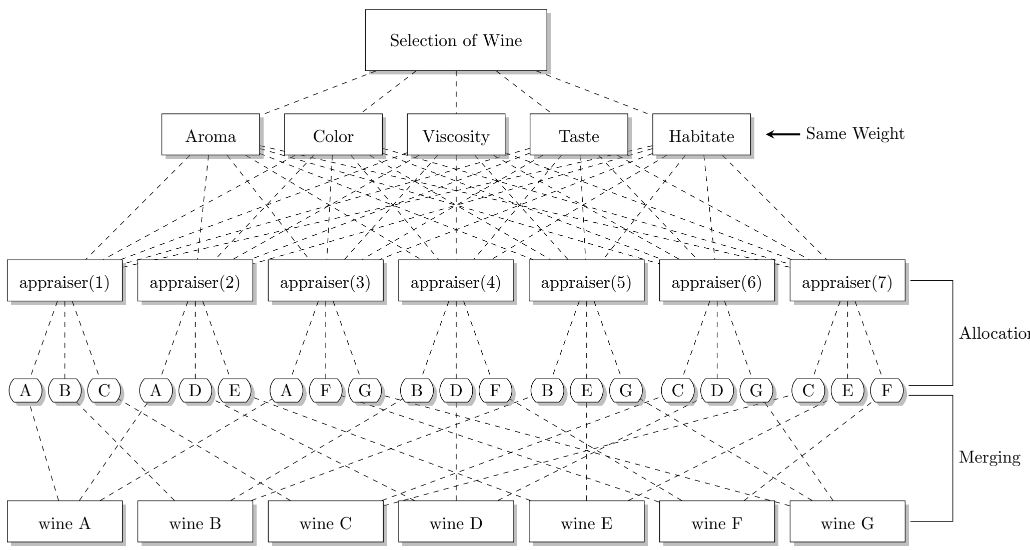

Figure 1: Result of assignment in AHP

- Step5: Revise the classic AHP chart by adding layer of sub decision makers
- Step6: Solve the problem by Group AHP

For the above solution procedure, computational complexity and validity are clear based on theory of combinatorial design and AHP.

6 Theoretical Approach and Methodology Let us consider optimal designs for $k = 3$ and $v = 7$. Identifying matrixes which become the same by permutation of their rows, we have 30 incidence matrixes as in Table 6.

Assume that there are 15 brands of wine ($v = 15$)

$$\mathrm{A, B, C, D, E, F, G, H, I, J, K, L, M, N, O.}$$

Table 6: All incidence matrixes for the case $k = 3$ and $v = 7$

$$\begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}$$

Since $bk = {}_v\mathrm{C}_2 = 105$, we needs 35 sommeliers. An example of the optimal designs is

$$\{\mathrm{A,B,C}\},\{\mathrm{A,D,E}\},\{\mathrm{A,F,G}\},\{\mathrm{A,H,I}\},\{\mathrm{A,J,K}\},\{\mathrm{A,L,M}\},\{\mathrm{A,N,O}\},$$
$$\{\mathrm{B,D,F}\},\{\mathrm{B,E,G}\},\{\mathrm{B,H,J}\},\{\mathrm{B,I,K}\},\{\mathrm{B,L,N}\},\{\mathrm{B,M,O}\},\{\mathrm{C,D,G}\},$$
$$\{\mathrm{C,E,F}\},\{\mathrm{C,H,K}\},\{\mathrm{C,I,J}\},\{\mathrm{C,L,O}\},\{\mathrm{C,M,N}\},\{\mathrm{D,H,L}\},\{\mathrm{D,I,M}\},$$
$$\{\mathrm{D,J,N}\},\{\mathrm{D,K,O}\},\{\mathrm{E,H,M}\},\{\mathrm{E,I,L}\},\{\mathrm{E,J,O}\},\{\mathrm{E,K,N}\},\{\mathrm{F,H,N}\},$$
$$\{\mathrm{F,I,O}\},\{\mathrm{F,J,L}\},\{\mathrm{F,K,M}\},\{\mathrm{G,H,O}\},\{\mathrm{G,I,N}\},\{\mathrm{G,J,M}\},\{\mathrm{G,K,L}\}$$

and the corresponding incidence matrix is shown in Table 7.

It is very troublesome to do this procedure manually. We intend to apply these matrixes to results of questionnaires and verify the effectiveness of our methods by simulating allocation of the decision-makers. Threfore, we develop a computer program for this purpose.

In this section, we only consider the cases where v and k satisfy the fair sharing condition (2).

Let W be a finite set with $\#(W) = v$ and $b = \frac{{}_v\mathrm{C}_2}{k}$. Optimal design $\mathcal{D}$ of W is a set of subsets of W, such that

1. $\#(\mathcal{D}) = b$,
2. $\#(X) = k$ for all $X \in \mathcal{D}$,
3. $\#(X \cap Y) \leq 1$ for all X, $Y \in \mathcal{D}$, $X \neq Y$.

Let $\mathcal{W} = \{X | X \subset W, \#(X) = k\}$, then, the set of all optimal designs is defined by

$$\Delta = \{\mathcal{D} | \mathcal{D} \subset \mathcal{W}, \#(\mathcal{D}) = b, \#(X \cap Y) \leq 1 \text{ for all } X, Y \in \mathcal{D}, X \neq Y\},$$

which we would like to enumerate as a sequence of incidence matrixes. However, this definition is not suitable for computer algorithms. So we have to rebuild Δ recursively.

We proceed with our argument on more general assumptions. Suppose a finite set $U = \{\alpha, \beta, \ldots\}$ and a symmetric relation $\sim$ between two elements of U are given.

For a nonnegative integer a and a subset $S \subset U$, we define $\Delta(S, a)$ by

$$\Delta(S, a) = \{A | A \subset S, \#(A) = a, \alpha \sim \beta \text{ for all } \alpha, \beta \in A, \alpha \neq \beta\}.$$

Notice that

$$\Delta(S, 0) = \{\emptyset\}$$

and if $\#(S) < a$

$$\Delta(S, a) = \emptyset.$$

Now we assume $S \neq \emptyset$ and $a > 0$. Choosing $\alpha \in S$ arbitrarily, $\Delta(S, a)$ is expressed as a disjoint union as follows:

$$\Delta(S, a) = \{A | A \in \Delta(S, a), A \ni \alpha\} \cup \{A | A \in \Delta(S, a), A \not\ni \alpha\}. \tag{3}$$

For a subset $T \subset U$, we define by $\alpha \sim T$ the subset

$$\alpha \sim T = \{\beta | \beta \in T, \beta \sim \alpha\}.$$

Table 7: An example of incidence matrix for the case $k = 3$ and $v = 15$

$$
\begin{pmatrix}
1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0
\end{pmatrix}
$$

Then, the first term of the right hand side of (3) is given by

$$\{A|A\in\Delta(S,a),A\ni\alpha\}=\{\{\alpha\}\cup B|B\in\Delta\left(\alpha\sim\left(S\setminus\{\alpha\}\right),a-1\right)\},$$

since

$$\{A\setminus\{\alpha\}|A\in\Delta(S,a),A\ni\alpha\}=\Delta\left(\alpha\sim\left(S\setminus\{\alpha\}\right),a-1\right).$$

For the second term of the right hand side of (3), we have

$$\{A|A\in\Delta(S,a),A\not\ni\alpha\}=\Delta\left(S\setminus\{\alpha\},a\right),$$

and accordingly

$$\Delta(S,a)=\{\{\alpha\}\cup B|B\in\Delta\left(\alpha\sim\left(S\setminus\{\alpha\}\right),a-1\right)\}\cup\Delta\left(S\setminus\{\alpha\},a\right).$$

Now we obtain

$$\Delta(S,a)=\begin{cases}\{\emptyset\} & \text{if } a=0,\\ \emptyset & \text{if } \#(S)<a,\\ \{\{\alpha\}\cup B|B\in\Delta\left(\alpha\sim\left(S\setminus\{\alpha\}\right),a-1\right)\}\cup\Delta\left(S\setminus\{\alpha\},a\right) & \text{otherwise.}\end{cases} \tag{4}$$

By using the programming language Haskell, the above construction (4) can be easily translated to a program code as below. That is the reason why we adopt Haskell.

```
delta :: [Int] -> Int -> [[Int]]
delta _ 0 = [[]]
delta ss@(alpha:sa) a
  | length ss < a = []
  | otherwise = (map (alpha:) $ delta (alpha \~ sa) (a - 1))
                ++
                (delta sa a)
```

We present the whole code in Appendix.

7 Concluding Remarks In this study we considered the case where one decision maker cannot compare all entities due to the heavy burden of pairwise comparison work. The necessity to allocate the burden to multiple decision makers leads us to the group-AHP problem. To solve this problem, we proposed an algorithm to generate an incidence matrix based on combinatorial design theory. For simplicity, our algorithm covers limited cases in which the fair sharing condition is satisfied. We will discuss more general cases with real data and verify the efficiency of our method in our further research.

A Source Code Now we present the whole source code.

```
module Main where

import System.Environment (getArgs)
import Data.Array ((!), array, listArray)

generateVectors :: Int -> Int -> [[Int]]
generateVectors v k
    | k == 0 = [replicate v 0]
    | v == k = [replicate v 1]
    | v > k = (map (1:) $ generateVectors (v - 1) (k - 1))
            ++
```

```
              (map (0:) $ generateVectors (v - 1) k)
    | otherwise = error "generateVectors"

relationR :: [Int] -> [Int] -> Bool
relationR w w' = (sum $ zipWith (*) w w') <= 1

takeTheFirstFromDelta :: Int -> Int -> Int -> [[Int]]
takeTheFirstFromDelta b v k
    = indexesToVectors $ head $ delta indexSet b
    where
      vectors = generateVectors v k
      l = length vectors
      indexSet = [1 .. l]
      indexedVectors = listArray (1, l) vectors
      indexesToVectors = map (indexedVectors !)

      matrixOfR = array (1, l)
                  [ (n, q n) | n <- [1 .. l] ]
          where
            q n = array (n + 1, l)
                  [ (m, r n m) | m <- [n + 1 .. l] ]
            r n m = relationR
                    (indexedVectors ! n)
                    (indexedVectors ! m)

      a \~ s = filter (matrixOfR ! a !) s

      delta :: [Int] -> Int -> [[Int]]
      delta _ 0 = [[]]
      delta ss@(alpha:sa) a
        | length ss < a = []
        | otherwise = (map (alpha:) $ delta (alpha \~ sa) (a - 1))
                      ++
                      (delta sa a)
      delta [] _ = []

readArgs :: [String] -> (Int, Int, Int)
readArgs = p . map read
    where
      p [] = (1, 3, 3)
      p [v] = p [v, 3]
      p [v, k]
          | m == 0 && n == 0 = (b, v, k)
          | otherwise = error "readArgs"
          where
            c = v*(v-1) `quot` 2
            (b,m) = c `quotRem` k
            n = c `rem` v
      p _ = error "readArgs"

main :: IO ()
main = do
  (b, v, k) <- fmap readArgs getArgs
  let s = takeTheFirstFromDelta b v k
```

```
    t = foldr1 (zipWith (+)) s
  putStrLn $ unlines $ map show s
  putStrLn $ show t
```

```
[1,1,1,0,0,0,0,0,0,0,0,0,0]
[1,0,0,1,1,0,0,0,0,0,0,0,0]
[1,0,0,0,0,1,1,0,0,0,0,0,0]
[1,0,0,0,0,0,0,1,1,0,0,0,0]
[1,0,0,0,0,0,0,0,0,1,1,0,0]
[1,0,0,0,0,0,0,0,0,0,0,1,1]
[0,1,0,1,0,1,0,0,0,0,0,0,0]
[0,1,0,0,1,0,1,0,0,0,0,0,0]
[0,1,0,0,0,0,0,1,0,1,0,0,0]
[0,1,0,0,0,0,0,0,1,0,0,1,0]
[0,1,0,0,0,0,0,0,0,0,1,0,1]
[0,0,1,1,0,0,0,1,0,0,0,0,0]
[0,0,1,0,1,0,0,0,1,0,0,0,0]
[0,0,1,0,0,1,0,0,0,1,0,0,0]
[0,0,1,0,0,0,1,0,0,0,0,0,1]
[0,0,1,0,0,0,0,0,0,0,1,1,0]
[0,0,0,1,0,0,1,0,0,0,1,0,0]
[0,0,0,1,0,0,0,0,1,0,0,0,1]
[0,0,0,1,0,0,0,0,0,1,0,1,0]
[0,0,0,0,1,1,0,0,0,0,0,1,0]
[0,0,0,0,1,0,0,1,0,0,1,0,0]
[0,0,0,0,1,0,0,0,0,1,0,0,1]
[0,0,0,0,0,1,0,1,0,0,0,0,1]
[0,0,0,0,0,1,0,0,1,0,1,0,0]
[0,0,0,0,0,0,1,1,0,0,0,1,0]
[0,0,0,0,0,0,1,0,1,1,0,0,0]

[6,6,6,6,6,6,6,6,6,6,6,6,6]
```

Figure 2: Computed result of our algorithm by haskell

References

Bolster, Paul J., Vahan Janjigian, and Emery A. Trahan (1995) "Determining investor suitability using the analytic hierarchy process," *Financial Analysts Journal*, pp. 63–75.

Cook, Wade D., Boaz Golany, Moshe Kress, Michal Penn, and Tal Raviv (2005) "Optimal allocation of proposals to reviewers to facilitate effective ranking," *Management Science*, Vol. 51, No. 4, pp. 655–661.

Cowan, Nelson (2001) "The magical number 4 in short-term memory: a reconsideration of mental storage capacity," *Behavioral and Brain Sciences*, Vol. 24, No. 1, pp. 87–114.

Durer, S., M. Ahlatcioglu, and F. Tiryaki (1994) "Ideal menkul kiymet portföyü olusturmada analitik hiyerarsi yaklasimi," in *Arastırma Sempozyumu'94, asım 1994, Devlet Istatistik Enstitüsü Konferans Salonu*, pp. 21–23, Ankara, Turkey.

Eason, George, Benjamin Noble, and I. N. Sneddon (1955) "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, Vol. 247, No. 935, pp. 529–551.

Fisher, Ronald Aylmer (1937) *The design of experiments*, Edinburgh; London: Oliver And Boyd.

Han, Sangheon (2014) "A practical approaches to decrease the consistency index in AHP," in *proceeding paper in SCIS & ISIS 2014*, Kitakyushu, Japan.

Ishii, Goro (1972) *Theory of Design of Experiments*, Tokyo: Baihukan.

Maxwell, James Clerk (1892) *A treatise on electricity and magnetism*, Vol. 2: Clarendon Press, 3rd edition.

Mazur, David R. (2010) *Combinatorics: a guided tour*: The Mathematical Association of America.

Miller, George A. (1956) "The magical number seven, plus or minus two: some limits on our capacity for processing information.," *Psychological Review*, Vol. 63, No. 2, p. 81.

Mirkin, Boris G. (1979) *Group Choice*, New York: John Wiley & Sons.

Oyatoye, E. O., G. U. Okpokpo, and G. A. Adekoya (2010) "An application of analytic hierarchy process (AHP) to investment portfolio selection in the banking sector of the Nigerian capital market," *Journal of Economics and International Finance*, Vol. 2, No. 12, pp. 321–335.

Rosa, Alexander (1991) "Combinational design theory — notes," `http://kam.mff.cuni.cz/~atcagc14/materialy/DesignTheoryNotes.pdf`.

Saaty, Thomas L. (1980) *The analytic hierarchy process*, New York: McGraw-Hill, Inc.

——— (1986) "Axiomatic foundation of the analytic hierarchy process," *Management Science*, Vol. 32, No. 7, pp. 841–855.

——— (1994) "How to make a decision: the analytic hierarchy process," *Interfaces*, Vol. 24, No. 6, pp. 19–43.

——— (2000) *Fundamentals of decision making and priority theory with the analytic hierarchy process*, Vol. 6: Rws Publications.

Saaty, Thomas L. and Luis G. Vargas (2012) *Models, methods, concepts & applications of the analytic hierarchy process*, Vol. 175: Springer Science & Business Media.

Thalheimer, Richard and Mukhtar M. Ali (1979) "Time series analysis and portfolio selection: an application to mutual savings bank," *Southern Economic Journal*, Vol. 45, No. 3, pp. 821–837.

Tiryaki, Fatma (2001) "The use of data envelopment analysis for stocks selection on Istanbul stock exchange," in *Management of Engineering and Technology, 2001. PICMET'01. Portland International Conference on*, Vol. 1, pp. 373–vol, IEEE.

Communicated by *Yoshio Tabata*