Granulation of Keywords into Sessions for Timetabling Conferences

Masahiro Tanaka†, Yasuyuki Mori* and Andrzej Bargiela‡

†Department of Information Science and Systems Engineering, Konan University
Kobe 658-8501, Japan.
m_tanaka@konan-u.ac.jp

* Department of Information Science and Systems Engineering, Konan University
Kobe 658-8501, Japan, and also at Takamatsu Junior College, Takamatsu, Japan.
ymori@takamatsu-u.ac.jp

‡Department of Computing and Mathematics, The Nottingham Trent University
Nottingham NG1 4BU, UK.
andre@doc.ntu.ac.uk

Abstract— **This paper proposes a timetabling method of conferences by using Kohonen's self organizing map (SOM). The timetabling problem is articulated as granulation of keywords into coherent sessions. The highly dimensional feature vector (representing keywords) is mapped with the aid of SOM onto a 2-dimensional grid that represents temporal and spatial dimension of individual sessions.**

Keywords— **Information granulation, Conference timetabling, Self Organizing Map, Automated timetabling, Soft computing.**

I. Introduction

Timetabling of a conference is a practical problem. Time is divided into several timeslots which are called "sessions". Each session must consist of several papers which deal with similar problems. The timetabling problem treated here is to group the papers to form a session, and allocate each session to a specified timeslots of a specified room.

This problem is widely known to all conferences' organizers, but very few number of papers dealing with this problem can be found even at major timetabling conferences ([3]-[5]). However, there are some related papers. For example, Cowling et al. [7] proposed a tool for personnel scheduling. Desrosiers et al. [8] dealt with a problem of designing MBA student teams. These problems have some sort of related issues.

Recently in many conferences, organized sessions are very popular. As is well known, an organized session is organized by a person who calls for papers and/or selects them for a session. Then the task of the conference organizer is to find session organizers and allocate the sessions on the timetable. The hierarchical structure of such an ogranizing method of a conference is useful in simplifying the procedure of timetabling, but there are various drawbacks, i.e. the problems of the quality of papers, the topics, etc. Even if the ogranizer is capable of collecting enough amount of good quality of papers, some authors who are competing in the field may not join the session. Considering this aspect, the amount of ogranized sessions in a general conference should be restricted. The importance of the ordinary sessions should be emphasized. The following discussion is focused on timetabling of non-organized sessions. Then the problem arises how to ogranize good sessions consisting of papers where each paper has attributes such as keywords, abstract, authors, and so on. Probably the common way is to divide the set of papers into several subsets based on the primary keyword or abstract and then sub-committees divide them into the set of several papers each of which are understood as a session by the attached information of the papers.

This is a safe method, however, the members of the committee have to meet at a certain place, which is very expensive with respect to time and money. Even if this is done, it seems impossible to avoid the cases that some papers are quite heterogeneous among other papers in same sessions. This could happen when the first partitioning of all the papers is errornous or some papers of minor fields are collected in sessions such as "miscellaneous session".

To avoid creation of miscellaneous sessions and to make each session filled with papers of similar interest, we treated this problem by using the grouping genetic algorithm [18]. The grouping genetic algorithm used there is based on the method proposed by Falkenauer [10], [11], [12], [13], [14] and adapted to our problem. The algorithm can yield a timetable of high-quality while satisfying the hard constraints. The major drawback of using this algorithm is that it implies heavy computational load. A program in Visual Basic, running on Pentium 4 PC, takes several hours to execute. Another drawback is that it intrinsically yields sessions of the same size, thus it is not a trivial task to meet the requirement of making a different size of sessions.

The method proposed in this paper is quite different from previously published methods in that it uses Kohonen's SOM [15], [16], [17]. SOM has several good features to be applied to the conference timetables, namely,

- The keywords can be used as the input vectors to the SOM.

- The generated 2-D map can be directly used as the timetable. Since the map preserves the topological relationship in the original data, the vectors mapped on the same unit are similar in nature, and the vectors mapped to the neighboring units have the similarity based on the distance upon the map.
- Large clusters are automatically spread over many units, and the numbers of data for the units tend to be uniform. This is suitable for session forming.

SOM was desined in early 80's by T. Kohonen [15], [16] and is a paradigm of the neural network of unsupervised learning. It is used to map a high-dimensional vectors onto a low-dimensional (typically one or two-dimensional) space so that humans can visually understand the high-dimensional data, which is otherwise hard to understand with regard to the similarity and the difference between the data points.

SOM is useful not only for visualization but also for granulation of the patterns. Bargiela and Pedrycz [1] used SOM for granulation of traffic data. Our problem here is also a kind of granulation of patterns. In this paper, we will show a preliminary experiment result of a new method of conference timetabling by using 2-dimensional (2D) SOM.

II. Problem Description

A. Conference Timetabling Problem

Let $\mathcal{P} = \{P_1, P_2, \cdots, P_n\}$, $i = 1, \cdots, n$ are accepted papers. Each paper P_i is supposed to have some predefined keywords

$$\mathcal{K}_i = \{k_i(1), \cdots, k_i(n_i)\} \subset \mathcal{K}$$

where $\mathcal{K}$ is the set of keywords related to the conference and n_i is not necessarily constant. Let $a(i)$ is the presenting author of the paper P_i which is represented as a unique number.

Table I is an example of the papers with attributes.

TABLE I
Papers and the attributes.

Paper Number	keywords k(1)	k(2)	k(3)	k(4)	k(5)	presenting author
32	12	11	28	65	–	102
73	3	11	17	46	37	15
⋮	⋮	⋮	⋮	⋮	⋮	

The Cartesian product of the rooms and the timeslots, i.e. $rooms \times timeslots$ are called "resources" $\mathcal{R}$.

Our problem is to allocate $\mathcal{P}$ to $\mathcal{R}$, or to find mapping:

$$\phi : \mathcal{P} \to \mathcal{R} \tag{1}$$

This problem has various kinds of constraints which can be classified into two categories: the hard constraints that must be kept strictly otherwise the generated timetable may contain severe problems, and the soft constraints by which the quality of the timetable may be affected.

The following constraints are probably the typical ones for timetabling of conferences.

Hard Constraints

H1 Each session has an upper limit of the number of papers.
H2 Any speaker cannot be a speaker in another session of the same timeslot.

Soft Constraints

S1 The papers of a same session are preferred to have as many as possible common keywords.
S2 The gap between the capacity and the number of actually allocated papers in each session should be as small as possible.
S3 The sessions in the same room are often preferred to have keywords in a same category.

B. Objective Function for Session Construction

Since we will use SOM, the objective function need not be defined strictly because the problem is not dealt with as an optimization problem. However, it may be useful if we define the objective function explicitly so that we can share the common awareness for the problem.

Probably the most important feature of the timetable is that the papers of a same session should have common keywords. Since the common keywords at each session cannot be predefined because SOM is unsupervised learning method, the fitness function must be defined by the keywords that actually appeared there. For this objective, we define the objective function based on the commonality among the papers in same sessions.

Let $k_1^{(i)}, \cdots, k_{m(i)}^{(i)}$ are the keywords appearing in session i, and the number of papers that have keyword $k_j^{(i)}$ is $N(k_j^{(i)})$. The objective function is the summation of the utility $u(N(k_j^{(i)}))$ over the timetable, i.e.

$$f = \sum_i \left(\sum_j u(N(k_j^{(i)})) \right)$$

where k is the number of papers that have this keyword in the current session.

The nonlinear function $u(\cdot)$ should be a nonlinear function which increases significantly as k becomes large.

C. The Objective Function for Session Allocation

In our approach using SOM, similar sessions come to neighbors. This is a welcome feature indeed for the neighboring time slots, but the similarity of the sessions is not needed in neighboring sessions of the same timeslots.

III. Solution Method

A. Paper Data for SOM

The input data must represent the feature of each paper. Thus we use keywords as the elements of the input vector.

Suppose there are distinct n-keywords in the papers. Then we define the paper k as a n-vector with elements

$$x_i(k) = \begin{cases} 1, & \text{if keyword } i \text{ is assigned for paper } k \\ 0, & \text{otherwise} \end{cases}$$

B. Structure and Training of SOM

As we have explained so far, we will use the 2D SOM.

The coordinate of the unit is expressed as (i, j), where $i \in \{1, 2, \cdots, p\}$ and $j \in \{1, 2, \cdots, q\}$. Now we define q as the number of parallel sessions to be held simultaneously, and p is the number of time slots during the conference days. Assume the data is n-dimensional. Each unit has a codebook vector $\boldsymbol{m}_{(i,j)} \in \mathbf{R}^n$. The codebook vector is a representative of the data, and each instance of data is mapped onto the unit whose codebook is nearest among the whole units. The training of SOM is done as the following procedure.

1. Initialize SOM. Randomly generate the codebook vectors $\boldsymbol{m}_{(i,j)}, i = 1, \cdots, p, j = 1, \cdots, q$.
2. Repeat the training by gradually making the neighbor size small.
   (a) Pick up the training datum $\boldsymbol{u}$ from the data set.
   (b) Find the unit $(i*, j*)$ whose codebook vector is nearest among all the codebook vectors.
   (c) Move the corresponding codebook vector slightly toward $\boldsymbol{u}$. That is

$$\boldsymbol{m}_{i*,j*} := \boldsymbol{m}_{i*,j*} + c\,(\boldsymbol{u} - \boldsymbol{m}_{i*,j*})$$

and also train the neighbor units in the same way, where c is a small positive parameter.

When the SOM is trained for sufficient large amount of time, the training data (papers) will be assigned to the units relatively scattered. However, we cannot expect that same number of papers are assigned to the units uniformly. Thus we need to modify the timetable.

C. Modification of Schedule

The method is heuristic that can be done automatically or manually. The authors think that it is not necessary to define this heuristic algorithm in a rigid way. Rather, this can be done arbitrary suitable to the conference.

An example is shown here. The heuristic is given as the following.

1. Find the largest session. Assign the papers to the empty sessions to make them full. When the session is full, it is removed from the table.
2. Do the above for the second largest, third largest,... sessions.
3. Assign the papers to its nearest free space. When the neighbor session is over-capacity, move the papers that were originally there to its neighbor.
4. Repeat the above procedure until every paper is assigned to a session.

IV. Experimental Result

In this section, we show a result using real data. The data set is the paper data presented at a domestic conference SCI'01 of the Institute of Systems, Control and Information Engineers in Japan.

The number of papers was 313. Originally the keywords were arbitrary given, but they were summarized into 86 keywords by us manually. The category keywords and the keywords were defined based on the covering fields of this institute described on the web page of this institute[1].

Table II shows the distribution of the keywords. First, 101 keywords were prepared, but some of them were not used in any papers.

TABLE II
Distribution of keywords.

Category	keywords	ocuurence times
1	1 ∼ 14	168
2	15 ∼ 32	175
3	34 ∼ 42	67
4	43 ∼ 60	98
5	61 ∼ 79	223
6	80 ∼ 101	148

Table III shows the keywords that appear in Table IV.

TABLE III
Category keywords and keywords (in part)

number	meaning
1	Structure analysis
3	Stability
5	Optimization
6	Multiobjective Optimization
15	System identification
16	Estimation
17	Linear and Nonlinear Control
18	Analysis of Stability
19	Digital Control
20	Optimal Control
21	Adaptive Control
22	Learning control
26	Robot Control
27	Process System
28	Stochastic Systems
30	Distributed, Time-Delay System
32	Others
33	Sensor
34	Actuator
37	Measuring method
38	Image processing applied measurement
39	Signal processing
40	Transmission
41	System instrumentation
42	Others
44	Software engineering
53	Virtual reality
60	Others
62	Inference
67	Image comprehension
70	Fuzzy Theory
71	Fuzzy inference
75	Genetic Algorithm
77	Distributed system
82	robot
89	Medicine
90	Energy
93	Traffic
99	Society

Table IV shows the number of papers first assigned by SOM.

[1]see Web page at http://www.iscie.or.jp/index-e.html

TABLE IV
Number of papers first assined by SOM.

	R1	R2	R3	R4	R5	R6	R7
T1	18	8	7	14	1	9	15
T2	3	1	2	1	0	1	18
T3	13	2	9	0	0	4	5
T4	4	0	8	0	13	6	5
T5	8	2	6	0	1	0	16
T6	9	3	27	5	5	0	7
T7	3	1	3	0	3	6	6
T8	13	8	5	11	3	6	16

The variance of the number of papers on the SOM is 34.62. The distribution of the mapped data on SOM is not so uniform as expected. This is because the elements of the vectors take either 0 or 1, and hence there are fairly a large amount of completely same vectors which cannot be separated on SOM by any method. In our paper using SOM [19], we created a new method of equalizing the number of papers on the SOM. The method is based on the adaptive weighting of the elements of the input vector. However it did not work well for this data. A notable feature of the data in this paper is that the vector is very high dimensional, and the elements are binary. These feature did not match well for the method in [19].

So, we cannot use this table as the schedule itself. However, we can see the quality of the classifier. For example, the paper information at unit $(3,3)$ are shown in Table V.

TABLE V
Papers at the unit (3,3).

paper number	keywords		
34	26	82	
39	62	67	82
41	77	82	
66	53	82	
120	26	35	82
205	26	82	
206	26		
225	6	82	99
276	33	82	89

We can see that 8 papers out of 9 have the keyword "82" which is "robot". Another keyword that is fairly common is "26" which is "robot control". So, we assign a session name "Robot control".

A neighbor unit $(4,3)$ has the following papers.

TABLE VI
Papers at the unit (4,3).

paper number	keywords		
56	33	37	39
270	33	37	41
271	33	37	
277	37	38	62
290	37		
291	33	38	
292	33	37	
309	18	37	93

We can see that and 7 papers out of 8 have "37" (measurement) in common, and 5 papers out of 8 have the common keyword "33"(sensor). So, this session can be named as "Measurement Sensors". Note that there is certain similarity between the contents of the above two units.

Now we check the unit far from these. As an example, let us check the unit $(2,5)$.

TABLE VII
Papers at the unit (2,5).

paper number	keywords			
86	3	17	18	90
110	3	17	44	
154	3	17	60	
163	3	17		
170	3	17	22	99
190	3	17	70	71

Here, 3 (stability) and 17 (linear and nonlinear control) are common to all the papers. Thus the session can be named as "Control and Stability".

V. Conclusions

In this paper, the problem of timetabling of conference programs was considered by applying SOM. We used 2D SOM, however, it may be interesting to use 3D SOM by taking the 3rd axis for the date. Since we defined the neighborhood in the usual way for SOM, the similarity held not only for the time-axis but also for the room number. However, it is not usually necessary for the neighboring rooms to provide similar kinds of sessions. They can be independent. Then the question may arise how to define the neighborhood. These topics will be the research themes for us in the future.

Acknowledgment

The authors express their gratitude to the Institute of Systems, Control and Information Engineers for providing us with the real data of the conference SCI'01.

This work was partly supported by the Hirao Taro Foundation of the Konan University Association for Academic Research. The first author was partly supported by Grant-in-Aid for Scientific Research (C) by Japan Society for the Promotion of Science (12680455).

References

[1] A. Bargiela and W. Pedrycz: Classification and clustering of granular data, Proc. of IFSA/NAFIPS'2001, Vancouver, July (2001)

[2] E. Burke, P. Cowling, P. De Causmacker and G. V. Berghe: A Memetic Approach to the Nurse Rostering Problem; *Applied Intelligence*, Vol.15, 199–214 (2001)

[3] E. K. Burke, Wilhelm Erben (Eds.): Practice and Theory of

TABLE VIII
Created Timetable (1).

Time	R1	R2	R3	R4
T1	Neural network	Knowledge Acquisition and neural nets	Learning	Bionics
T2	Neural network	Robot and image processing	Robot	Hybrid Control
T3	Image processing	Image processng	Robot control	Discrete system
T4	Multimedia	–	Measurement and sensor	–
T5	Image processing and virtual reality	Human interface and virtual reality	Image measurement and sensor	Genetic algorithm
T6	Human interface	Medicine and autonomous system	Finance	Energy and adapive control
T7	Medicine	Computer network	Robust control	Genetic algorithm
T8	Computer network	Knowledge acquisition and fuzzy theory	System identification	Modeling

TABLE IX
Created Timetable (2).

Time	R5	R6	R7
T1	Biotics	GA	GA and Neural networks
T2	Stability of systems	Control and stability	GA
T3	Optimization	GA applications	Optimization
T4	Structure analysis and modeling	Optimization of discrete systems	Optimization
T5	Discrete systems and vehicles	Neural networks	Optimization
T6	Discrete systems	Neural networks	Optimal control
T7	Distributed systems and stability	GA and autonomous systems	Control of linear and nonlinear systems
T8	Modeling	Control of linear and nonlinear systems	Control of linear and nonlinear systems

Automated Timetabling III, Third International Conference, PATAT 2000, Konstanz, Germany, August 16–18 (2000)

[4] E. K. Burke, Michael W. Carter (Eds.): Practice and Theory of Automated Timetabling II, Second International Conference, PATAT'97, Toronto, Canada, August 20–22 (1997)

[5] E. K. Burke, Wilhelm Erben (Eds.): Practice and Theory of Automated Timetabling III, Third International Conference, PATAT 2000, Konstanz, Germany, August 16–18 (2000)

[6] E. K. Burke and A. J. Smith: Hybrid Evolutionary Techniques for the Maintenance Scheduling Problem; *IEEE Transactions on Power Systems*, Vol.15, No.1, 122–128 (2000)

[7] P. Cowling, G. Kendall and E. Soubeiga: Hyperheuristics: A Tool for Rapid Prototyping in Scheduling and Optimisation. in *S. Cagnoni et al. (eds.): EvoWorkshops 2002*, LNCS 2279, Springer, 1–10 (2002)

[8] J. Desrosiers, N. Mladenovic and D. Villeneuve: Design of Balanced MBA Student Teams, MIC'2001 (4th Metaheuristics International Conference), Porto, Spain, July 18, 281–285 (2001)

[9] A. E. Eiben, J. K. Vander Hauw and J. I. Van Hemert: Graph Coloring with Adaptive Evolutionary Algorithms; *Journal of Heuristics*, Vol.4, 25–46 (1996)

[10] E. Falkenauer: A New Representation and Operators for Genetic Algorithms Applied to Grouping Problems; *Evolutionary Computation*, Vol.2, No.2, pp.123–144 (1994)

[11] E. Falkenauer: A Hybrid Grouping Genetic Algorithm for Bin Packing; *Journal of Heuristics*, Vol.2, 5–30 (1996)

[12] E. Falkenauer: *Genetic Algorithms and Grouping Problems*, John Wiley & Sons (1998)

[13] E. Falkenauer: The Lavish Ordering Genetic Algorithms; *Metaheuristics: advances and trends*, 249–256 (1999)

[14] P. De Lit, E. Falkenauer and A. Delchambre: Grouping Genetic algorithms: an efficient method to solve the cell formation problem; *Mathematics and Computers in Simulation*, Vol.51, 257–271 (2000)

[15] T. Kohonen, Analysis of a simple self-organizing process, Biological Cybernetics, **44** 135–140(1982)

[16] T. Kohonen, Self-organizing formation of topologically correct feature maps, Biological Cybernetics, **43** 59–69(1982)

[17] T. Kohonen, Self Organizing Maps, Springer-Verlag (1995)

[18] Y. Mori and M. Tanaka: A hybrid grouping genetic algorithm for timetabling of conference programs, PATAT2002, accepted.

[19] M. Tanaka and Y. Kajitani: Interactive Weighting SOM for Data Mining, in N.Baba, C.J. Jain and R.J. Howlett (Eds.) Knowledge-Based Intelligent Information Engineering Systems & Allied Technologies, IOS Press, 37–41 (2001)