Proceedings of the International MultiConference of Engineers and Computer Scientists 2010 Vol I,
IMECS 2010, March 17 - 19, 2010, Hong Kong

# Constructing Student Personal Course Scheduling Problem with Spreading Activation on a Course Network

Hori, Y. *, Nakayama, T., Imai, Y.

*Abstract*— **In general, university student have to decide own course schedule via course syllabus. However, at a large number of classes, it is very exhaustive to manually generate the course schedule for the students, due to various constraint condition or criteria. This paper has formalized this problem as a constraint satisfaction/optimization problem, developed an automated tool for a student personal course schedule, and evaluated the effectiveness of our system. We represent students' interest as a user profile model generated from syllabus browsing histories, keyword query and cluster of the curricum to find a suitable course schedule for a student.**

***Keywords: Constraint Satisfaction Problem, Clustering, Data-mining, Syllabus***

## 1 Introduction

University Students have to decide their own course schedule by themselves. In order to make a course schedule, it is necessary to satisfy the student's interest and to meet course credit restrictions. On the other hands, the university's curriculum changes dynamically to catch up with social needs, advanced science and technology. In the many university, there are on-line course support system that allows students view all subjects information is available[4]. However, it is not easy for students to generate manually a course schedule from a large number of combination of classes, due to various constraints and/or criteria, especially for the freshman in the university.

Our university's students must entry about 57 subjects from 521 subjects in the 4 years curriculum. The university only requires a few prerequisite courses, and their flexible system allows students to choose from a wide range of courses. However it is difficult to check carefully all course schedules without their learning plans. Therefore, the adviser-system to support students has been adopted in our university. The role of advisers is expected to respond to the students who have various problems on their curriculum, to give information on the course planning. This system is very ineffective from the viewpoint of cost and time.

Nowadays, most universities are introducing syllabus to support students' subjects registration. Syllabus is commonly a booklet that shows outlines of subjects. Students decide to take their courses by reading syllabus. However, the syllabus has some problems. One of the problems is that there are some difficult information to comprehend in syllabus. Web syllabus provides only a course information, therefore students cannot know overall structure of the curriculum. In the past studies, Although some applications provide the capability to support making students' own course schedule, there are a requirement to know their courses and the curriculum. It is not always the case that the students have much knowledge in the field of their curriculum. Therefore, there has been no effective way of planning the course schedule from these applications.

In the next section, we introduce some related works. In section 3, we define the feature space and then describe the method of making a course schedule. In section 4, we describe the results of the preliminary use test. We conclude this paper in section 5.

## 2 Related Works

Nozawa et. al. had been proposed a syllabus analysis system[2]. In designing of an original curriculum by a higher education institution, or in external evaluation of an institution's curriculum, comprehending the curriculum contents of many institutions in the same field is necessary. However, this is very hard to do even for education experts in that field. This system uses clustering view from each class syllabus description. The main aim of this system is not for students, but for teachers to make specialized curriculum. This system treats syllabus data which constitute some curriculum and are expressed in a common format, calculates the similarity between the syllabi based on the occurrence frequency of technical terms, clusters the syllabus, and thus helps to find distinguishing features of the curriculum by visualizing and comparing the assignments of the syllabus to the clusters along various classification axes. Visualizing of syllabus data is useful to make student's study strategy, but it is not always the case the students find suitable course plan because of their lack of desire. We have tried to make a model of the student's interests.

Ohiwa et. al. has developed idea generation tool with card-handling[6]. Card handling is one of the most useful methods for information representation and idea generation. A generated card can be picked and moved by a mouse. Cards may be grouped by enclosing them with a curve. Relationships of cards and groups can also be marked by special lines. This system can be used for

*Kagawa University Information Technology Center, Saiwai 2-1, Takamatsu, Japan 760-0016 Email: horiyuki@itc.kagawa-u.ac.jp

ISBN: 978-988-17012-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

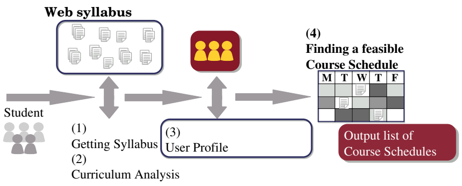


Figure 1: The configuration of Active Syllabus

the course planning. However, there are a lot of restrictions about course credit in the curriculum. It is difficult to operate card-handling due to various constrain condition or criteria. Thus, we have formalized this problem as a scheduling problem for a student personal course schedule[1].

The spreading-activation-based model has been proposed in previous studies investigating the syntactic priming (SP) effects in the document understanding [13][14][16]. Kojima has proposed a framework of document understanding using text information and context[17]. These past studies focused human's memory activation to find topics words. We applied this spreading-activation-based model to well-concerned course planning via all courses in curriculum.

The course planning is a multi-objective combinatorial optimization problem. In past study, some application system has been reported in a nurse work time scheduling[9] and a trainman allocation problem[10][11]. We have formalized students course planning as a constrain satisfaction/optimization problem. This approach is particularly effective where there are many constraints. An objective function is first defined to measure the degree of active value by a course plan. We set out to optimize a problem defined by this objective function and the block of constraints for one student. Using this approach, we optimized every student's course plan in a semester. This suggested the proposed system could be used for discovering useful course schedule. This shows the proposed system supports the student to easily make their own course schedule and to easily to know new interesting course.

# 3 Generating Course Schedules

This section describes about how to generate time schedules using spreading activation on a course network. We define the feature space represented as user's interest and learning strategy, formalize as a constraint satisfaction/optimization problem to generating time schedules.

Feature vectors in the field of text are based on a set of nouns extracted from syllabus text; examples include Web mining[19]. In this case, the differences among the user's interest/learning strategy and courses are well represented by term correspondences. We use the heuristic search technique to solve a constraint satisfaction/optimization problem. We show the system architecture of generating time schedules in Figure 1.

1. Getting syllabus text
   We consider the syllabus data discovery problem to find sets of Web page, each of which share some template, among many pages collected in the university web site by a Web crawler. A found set is a potential input for information extraction and wrapper generation algorithms. These results are restored into database and performing edit and browsing on our system.
2. Curriculum Analyzing
   We represent feature vector based on a set of nouns extracted from syllabus text. The differences among the courses are represented by the similarity between the syllabus based on the occurrence frequency of technical terms. Next, we use the Repeated bisection clustering method to analyze the curriculum's preference.
3. User Profile
   We represent student's interest and learning strategy as a set of nouns extracted from syllabus text. We has been developed 3 types of making user's profile method.
4. Finding a feasible course schedule
   The student's course schedule is built via to solve a constraint satisfaction/optimization problem from the user profile.

From the next section, details in each step and the system behavior are described.

## 3.1 Getting syllabus text data

Obtaining syllabus information on the syllabus web site is described here. In general, there are some fundamental information and contents of the subject in the description of the syllabus. Fundamental information contains subject name, target faculty/grade, credit information (core/required/elective), teacher's name, semester, lecture day and number of credits. The contents of the subject contains summary, goal, teaching method, course content. Our system extracts all of these as a subject data and define $c_i$ vector. The feature vector of each subject was constructed as follows. The weight of term $w_{ij}$ was defined using tf-idf.

$$c_i = (w_{i1}, w_{i2}, \ldots, w_{in})$$
$$w_{i,t_j} = \mathrm{tf}(t_j, c_i) \cdot \mathrm{idf}(t_j)$$
$$\mathrm{idf}(t_j) = \log(N/\mathrm{df}(t_j))$$

where $\mathrm{tf}(i,j)$ is the occurrence of term $t_j$ in a subject $j$, $N$ is the number of total subjects, and $\mathrm{df}(t_j)$ is the number of subject term $t_j$ appeared once or more. We use mecab[1] to extract noun from syllabus data. The followings were excluded as stop words.

1. Words of one letter hiragana and katakana in Japanese.

[1] http://mecab.sourceforge.net/

ISBN: 978-988-17012-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

2. Low frequency terms: The terms under 70% of frequency were excluded.

### 3.2 Curriculum Analyzing

To facilitate comprehension of the curriculum's features, our system is utilizing document-clustering of syllabus data. The repeated bisection clustering is used in order to get student's intended course plan.

$$\begin{aligned} L_{ij} &= \frac{c_i c_j}{||c_i|| \, ||c_j||} \\ &= \frac{\sum_{k=1}^{n} w_{ik} w_{jk}}{\sqrt{\sum_{k=1}^{n} w_{ik}^2} \sqrt{\sum_{k=1}^{n} w_{jk}^2}} \end{aligned} \quad (1)$$

Our system makes syllabus clusters using similarity between the syllabus based on the occurrence frequency of technical terms. The clusters help to find distinguishing features of the curriculum. Our system uses ward-method to make clusters. Moreover, because the size of the cluster is different in the subject and each curriculum, the faculty staff has adjusted the clustering results.

### 3.3 User Profile

There are many combinations of the course schedules in the term, and the student has an original intended plan. Our system makes the student profile as follow.

$$U = \{u_1, u_2, \cdots, u_m\}_{u_i \in S}$$

Here, $u_i$ is a word described in the syllabus, $S$ is all syllabus set. However, it is not good way to manually choose various words via syllabus to make a student profile. Our system has 3 types of strategies to make a student profile. The student can use these strategies by combining.

- Manual select
  The students select words from syllabus by manual operations.
- Cluster select
  By clustering the syllabus, classified fields of subject is generated. The student selects a cluster and it includes various keywords.
- Browsing behavior
  Students repeat browsing which follows links from one page to another, and examine the contents of the pages they got by link selection. Students expect to acquire useful information by following syllabus. In a word, a student's syllabus browsing behavior could be utilized as an intended query. Our system extracts, keywords from the student's syllabus browsing history.

### 3.4 Course scheduling

As discussed previously, making a course schedule is formalized as a scheduling problem. We use the courses contain the keywords in the user profile for an initial solutions of the scheduling problem, solve the scheduling problem using iterative scheme with best-first search. To solve a scheduling problem, There are five conditions as the follows.

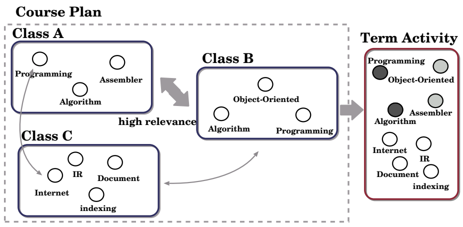


Figure 2: The spreading activation model for a subject network

- An output must consider about 1 semester's schedule.
- An output contain classes from the class in each semester and grade.
- An output must consider the schedule within the restriction and number of credits of each semester.
- An output must include the required classes.
- An output must consider the student's interest.
  Out system sets the objective function and chooses the one with the maximum value for the objective function.
- An output must consider a student's personal schedule.

We formalized this as a constraint satisfaction optimization problem.

Output:
  Course Schedule $C = \{c_1, ..., c_o\}$
Input:
  User Profile $U = \{u_1, ..., u_m\} u_i \in S$
Maximize: $T$
  $T = \sum_i a_i \cdot O, \; a_i = \sum_j a_j L_{ij} (i, j = 1, 2, \cdots, o)$
(2)
Subject to
  $min <$ credit$(C) < max$
  $M\{m | m$ is a required course$\} \subset C$
  $P\{p | p$ is a prohibited course pattern $\} \not\subset C$
  $D\{d | d$ is a weight value which represents student's requests $\}$

Here, credit$(C)$ is number of credit in the course schedule $C$, $m$ is a required subject in the semester, $p$ is a prohibited course pattern, $d$ is a weight value which represents student's requests.

The objective function $T$ in (2) is sum of $c_i$ in the course schedule $C$ with $L_{ij}$ represents similarity between each classes. This function represents the network with the vector and similarities of classes in the course schedule.

ISBN: 978-988-17012-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

$a_i$ is active value of class node $c_i$. Therefore, $T$ means active value of the course network from the view point of student's interest.

The spreading activation based model has been proposed in previous studies investigating the syntactic priming (SP) effects in head-initial languages (e.g., English). The cohesiveness is calculated by spreading activation on a semantic network constructed systematically from word network. The spreading activation based model is defined as follows.

$$A(t) = C(t) + ((1-\gamma)I + \alpha L)A(t-1) \qquad (3)$$

where $\gamma$ is attenuation, $\alpha$ is a propagation parameter, $A(t)$ is a vector of active value when the firing count is $t$, $C(t)$ is a vector of active value from syllabus text, $I$ is identity matrix and $L$ is a propagation matrix and it's element $L_{ij}$ represents a similarity metric between node $i$ and $j$ such as cosine similarity. In equation (2), $a_i = \sum_j a_j \cdot L_{ij}$ is an element of activities, a vector of active value presents $A = (a_1, a_2, \cdots, a_o)^t$ if propagation count $t$ is left out. This means $C(t) = 0$, $\gamma = 1$, $\alpha = 1$ in equation (3). Therefore equation (3) becomes as follows.

$$A(t) = LA(t-1)$$

Here, our system uses only one time propagation to the course network. Sum of each element $A(t)$ is activation-based value in entire course network. $O$ is a projection operator from syllabus feature vector to student interest vector, then activation-based value of entire course network represents $T$ in equation (2). This projection means activation-based value via student's interest on a course network. We use $T$ and $d$ represents student's requests as the objective function.

The solution of the objective function is larger when a course schedule has similar courses. Therefore, as can be seen from Fig 2, we can represent the word network model which has an ability to optimize both connection weights and the activation functions. For the first trial of making a course schedule, a course schedule matched by a student profile is used. Our system chooses the one with the maximum value for the objective function. This becomes the new trial solution for the next iteration, and repeat this iterative process until a satisfactory and feasible schedule is obtained. This optimization is as follows.

1. Make an initial course schedule matched by a student profile.
2. Choose one subject from current course schedule, and replace with the most similarity other subjects.
3. If objective function's value is updated, this trial solution is replaced to current one.
4. repeat 2. until the whole of subjects is checked or replacing.

### 3.5 Example of Output

Fig 3(upper-side) shows an example of course schedule by our system. Fig 3(down-side) shows an example of course schedule making by a real student. * mark shows a required subject in the semester.

A course schedule in Fig 3(upper) is made by a profile includes words related computer science. This course schedule has more classes in the field of computer science and mathematics than student's one.

## 4 Experiment

We have developed our system as Web system and CLI system, applied to our university's curriculum, and evaluated the effectiveness of our system. The test involved a total of 6 student ranging in grade from 3 to 4. We requested each student to describe the relevance ratio in the result course schedule.

### 4.1 Outline of algorithm evaluation test

We first describe the curriculum for this experiment. Table 1 shows a credit requirements in our university. The number in () shows registrable course number in the course division. Table 1 In this curriculum consist of two basic elements. First one is public lecture course to acquire broader knowledge and understanding, the other courses are professional education to expertise acquisition in a particular area.

The number of courses, keywords, and the target grade are shown as Table 2. As there is a lot of variation in the 4th grade student's course schedule, it have been eliminated from the experiment.

Table 2: Experimental data

| | |
|---|---|
| Target curriculum: | Dept of Information Systems Engineering, Kagawa University |
| Number of courses: | 513 |
| Number of keywords: | 7,819 |
| Target grade: | 1 $\sim$ 3 1st/2nd semester |

First, We requested each student to describe their own course schedule. Next we requested each student to select course schedule via our system. The number of courses are same as the courses in their own course schedule. We requested each student to test $p$ courses of goodness of fit in the course schedule which has $n$ courses. However they didn't know if the course schedule is made by our system or human. We define the average of precision $p/n$ of the course schedule by our system and human. We selected the student's profile as fit each course schedule by human.

We evaluate how well combine the course schedule suited the student by comparing the average of precision.

### 4.2 Results

Table 3 shows a comparison of the precision.

Our system is not shown high score, but shown higher score than human's one in all grades. This is because the course schedule by our system has a lot of related courses, the students evaluate that is good to learn. On the other hands, the course schedule by human has one

ISBN: 978-988-17012-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

| | Mon | Tue | Wed | Thu | Fri |
|---|---|---|---|---|---|
| | 1st grade time schedule by active syllabus(upper 1st semeter/bottom 2nd semester) | | | | |
| 1 | Fundamental Mathematics | Introduction to Statistics | Computer System* | | Modern History |
| 2 | General Biology C | Introduction to Economics | | | Introduction to Mechanical Engineering |
| 3 | | Introduction to Mathematics | Artificial Intelligence | | |
| 4 | | | Programming I * | Differential and Integral Calculus I | |
| 5 | | | Programming I * | Mathematical Science I * | |
| 1 | Fundamental Informatics | Introductory Career-design I | Probability and Statistics* | | |
| 2 | | Community and Sociology | Mathematics Science II* | | |
| 3 | | Fundamental Life Science | Logical Circuit* | | |
| 4 | | | Programming II* | Differential and Integral Calculus | |
| 5 | | | Programming II* | | Basic Physics |
| | 1st grade time schedule by student (upper 1st semester/bottom 2nd semester) | | | | |
| | Mon | Tue | Wed | Thu | Fri |
| 1 | Algebra | Basic Physics | Computer System * | German I | Biology |
| 2 | English I | Life and Human | Gender and Sexuality | | German-Nordic Legend and Opera |
| 3 | German I | | | | |
| 4 | | | Programming I* | Differential and Integral Calculus I | |
| 5 | | | Programming I* | Mathematical Science I* | English Communication |
| 1 | | Life and Human | Probability and Statistics * | Introduction to Psychology C | Human Communication |
| 2 | English II | German II | Mathematical Science II* | Sociology D | |
| 3 | | | Logical Circuit * | German II | |
| 4 | | | Programming II* | Differential and Integral Calculus | Cross-cultural Communications |
| 5 | | | Programming II* | | |

Figure 3: An example course schedule generated by Active Syllabus and student

Table 1: Experimental curriculum data

| Category | | | Required credits for graduate | |
|---|---|---|---|---|
| General Education Subjects | Theme subject (53) | | 8 credits | |
| | General Education Seminar [elective] (52) | | 2 credits | |
| | Common Subjects (70) | | 8 credits | |
| | Health · Sports [elective] (44) | | 2 credits | |
| | General Education for upper-grade [elective] (6) | | 4 credits | |
| | Foreign Language | Second Foreign Language (145) | 4 credits | 24 credits |
| | | First Foreign Language (71) | | 6 credits |
| | Subtotal | | | 30 credits |
| Faculty of Engineering Classes | General Engineering Subject | Analytical Thinking Subject (9) | | 8 credits |
| | | Communication Skill (5) | | 6 credits |
| | | Fundamental Mathematics/Science Skill (8) | | 10 credits |
| | Specialized Subject | Fundamental Specialized Subject (32) | | 30 credits |
| | | Advanced Specialized Subject (32) | | 32 credits |
| | | Graduation Research | | 6 credits |
| | Elective courses | | | 6 credits |
| | Subtotal | | | 98 credits |
| | Total | | | 128 credits |

or two unrelated courses. They selected such courses for the credit.

### 4.3 Comparison of the user profile

Next, we evaluate how much keywords in the student profile was corresponding to the result of course schedule. In this test, we calculate F-value of the keywords in the result course schedule in the student profile. We use three methods to make a profile for this test, (R) Select keywords from target courses in random order, (F) Select keywords according to how frequently it appeared from target courses and (C) Select keywords from one of the course classification result.

We considered that (R) is implemented for students who have no knowledge in the their major field, (F) is for students who depend entirely on the others' behavior, (C) is for students who want to know one field on the curriculum. The history of the number of register are used in the past 5 years for (F). In this test, we use curriculum data in Table 2, set the number of courses in the result is 8 for the reason of our university's credit limitation.

Fig 4 shows the results of the first degree's course schedule. The histogram vertical axis shows average F-value of 100 times of the result, and horizontal axis shows the

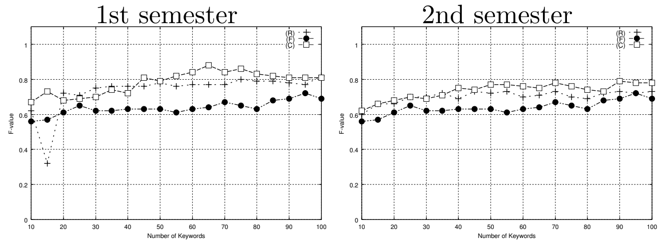


Figure 4: The results of the objective evaluation of first grade's course schedules

number of keywords in the student profile. There is the case of F-value shows under 0.6, however we got over 0.6 for F-value for the this test. Fig 5 shows average of F-value and distribution of activity value.

It can read that there is a trade-off relation between activity values and F-values from Fig 5. This is for the reason that the objective function is used activity value of the course network with the maximum. Active Syllabus search a subject that has more similarity other subjects than the matching rate of the student profile. F-values of (R) and (C) are same, however activity value of (R) is lowest. It is thought that there is less related subjects in


ISBN: 978-988-17012-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Table 3: The results of the subjective evaluation

| Grade/Semester | 1/1 | 1/2 | 2/1 | 2/2 | 3/1 | 3/2 | average |
|---|---|---|---|---|---|---|---|
| Manual | 0.78 | 0.80 | 0.79 | 0.91 | 0.94 | 0.90 | 0.85 |
| Active Syllabus | 0.85 | 0.86 | 0.87 | 0.91 | 0.94 | 0.94 | 0.89 |

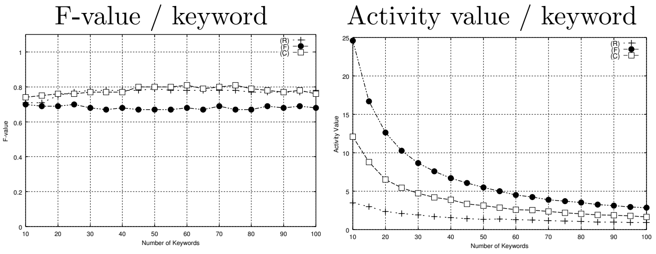


Figure 5: The distribution of F-values and activity values for the number of keywords

the (R)'s course schedule. (C) and (F) shows high score of activity value. It is though that the cause of there is much related subjects in the the student profile of (C) and (F). As previously experiment result, students decide that there is much related subjects in the course schedule is preferred.

## 5 Conclusions and Future Work

In this paper, we have proposed a plan support system Active Syllabus aimed improving the efficiency of course schedules. This system made the course schedule using the spreading activation model in consideration of student's interest and the relation between subjects. The result of the experiment, F-value was about 0.7. Our system should be doing more tests to confirm these preliminary results with in real use case.

Using activation spreading model into the syllabus description, Our system can provide not only existing classes matched by a student's interest, but also their relevant classes. This is differ from a lot of past research that merely provide show the class information. Promising direction for future works are the optimization of period to makes a student profile, the examination of syllabus descriptions and the similarity between classes. These problems should be considered for wide coverage.

## References

[1] Yukio Hori, Masashi Takimoto, Yoshiro Imai: A support system for course schedule design based on syllabus description, 8th International Conference on Information Technology Based Higher Education and Training (ITHET),pp.58-62, (2007).

[2] T. Nozawa, M.Ida, et. al.: Construction of Curriculum Analyzing System Based on Document Clustering of Syllabus Data(Education), Transactions of Information Processing Society of Japan Vol.46, No.1, pp. 289-300, (2005).

[3] K. Miyazaki, M. Ida, F. Yoshikane, T. Nozawa, H. Kita: Development of a Course Classification Support System for the Awarding of Degrees using Syllabus Data, IPSJ Journal, Vol.46, No.3, pp.782-791, (2005).

[4] S. Yamada, Y. Matsunaga, E. Itoh, S. Hirokawa: A Study of Design for Intellingent Web Syllabus Crawling Agent, The transactions of the Institute of Electronics, Information and Communication Engineers, D-I, Vol. J86-D-I, No.8, pp.566-574 (2003)

[5] Y. Chubachi, Y. Ishii, H. Ohiwa: enTrance System Fundamental Concept and Value for a University EnvironmentSoftware Engineers AssociationSoftware Symposiumpp.181-184 (2001).

[6] H. Ohiwa, N. Takeda, K.Kawai, A. Shimomi: KJ editor: a card-handling tool for creative work support, Knowledge-Based Systems, vol.10, pp.43-50 (1997).

[7] Y. Ohmi, K. Kawai, N. Takeda, H. Ohiwa: Pointing Operations in Cooperative Works using Card-handling Tool "KJ-Editor", Transactions of Information Processing Society of Japan, Vol.36, No. 11, pp.2720-2727 (1995).

[8] A. Ikegami, A. Niwa, M. Ookura: Nurse Scheduling Problem in Japan, Operations research as a management science research ,Vol.41, No.8, pp. 436-442 (1996).

[9] Aickelin, U. and Dowsland, K. A.: An indirect genetic algorithm for a nurse-scheduling problem. Comput. Oper. Res. 31, 5, 761-778 (2004)

[10] A. Caprara, M. Fischetti, P. Toth: A heuristic method for the set covering problem, Operations Research, Vol.47, pp.730-743 (1999)

[11] Easton, K., Nemhauser, G., and Trick, M.: The traveling tournament problem description and benchmarks, in Proceedings of Seventh International Confer- ence on Principles and Practice of Constraint Prodram- ming (CP 99), pp. 580-584 (1999)

[12] M. Horio, A. Suzuki: Development of a General-purpose Scheduling Solver under the Framework of RCPSP/.TAU. with Time Constraints. Journal of Japan Industrial Management Association, Vol.54, No.3, pp.203-213, (2003).

[13] J.R. Anderson: A spreading activation theory of memory, Journal of Verbal Learning and Verbal Behavior, 22, pp.261-295 (1983)

[14] A.M. Collins and E.F. Loftus: A Spreading Activation Theory of Semantic Processing, Psychological Review, 82, pp.407-428 (1975)

[15] R.F. Lorch: Priming and Searching Processes in Semantic Memory: A test of three models of spreading activation, Journal of Verbal Learning and Verbal Behavior, Vol.21, 468-492 (1982)

[16] D.L. Waltz, J.B. Pollack: Massively parallel parsing, A Strongly Interactive Model of Natural Language Interpretation, Cognitive Science, 9, pp.51-74 (1985)

[17] H. Kozima, A. Ito: Context-sensitive word distance by adaptive scaling of a semantic space, R. Mitkov, N. Nicolov (eds.), Recent Advances in Natural Language Processing, Contemporary Issues in Linguistic Theory 136, Amsterdam: John Benjamins, pp.111-124 (1997)

[18] B.A.Huberman and T.Hogg: Phase Transition in Artificial Intelligence Systems, Artificial Intelligence, Vol.33, pp.155-171 (1987).

[19] Perkowitz, M. and Etzioni, O.: Towards adaptive Web site: Conceptual framework and case study, Artificial Intelligence, Vol. 118, pp. 245-275, (2000).