PERGAMON Computers and Mathematics with Applications 46 (2003) 831–848
An International Journal computers & mathematics with applications
www.elsevier.com/locate/camwa

Acquisition of Compound Skills and Learning Costs for Expanding Competence Sets

YI-CHUNG HU
Department of Business Administration
Chung Yuan Christian University
Chungli 320, Taiwan, R.O.C.

RUEY-SHUN CHEN
Institute of Information Management
National Chiao Tung University
Hsinchu 300, Taiwan, R.O.C.

GWO-HSHIUNG TZENG* AND YU-JING CHIU
Institute of Management of Technology
National Chiao Tung University
Hsinchu 300, Taiwan, R.O.C.
ghtzeng@cc.nctu.edu.tw

(Received March 2002; revised and accepted June 2002)

Abstract—For each decision problem, there is a competence set consisting of ideas, knowledge, information, and skills for solving that problem. When decision makers have not acquired the competence set, it is more difficult for them to make decisions. In order to effectively acquire a needed competence set to cope with the problem they face, finding an appropriate learning sequence for acquiring needed single skills for decision makers, the so-called competence set expansion, is very necessary. A compound skill represents a collection of single skills that might be acquired, and some useful compound skills can be added to the needed competence set to help acquire some single skills. To effectively expand the competence set, effective acquisitions of compound skills and learning costs are both necessary. This paper thus proposes a data mining technique to extract potentially useful compound skills from single skills. Subsequently, an effective method is proposed to obtain the learning cost between any two skills. A computer simulation is employed to further show that it is feasible to use those potentially useful compound skills to facilitate the acquisition of single skills through a known integer programming method for expanding the competence set. © 2003 Elsevier Ltd. All rights reserved.

Keywords—Competence set, Data mining, Neural networks, Fuzzy sets, Integer programming.

1. INTRODUCTION

With competence sets as proposed by Yu and Zhang [1], for each decision problem (e.g., promoting products or improving services for a business) there exists a competence set consisting of ideas, knowledge, information, and skills for solving that problem. When decision makers

*Author to whom all correspondence should be addressed.

0898-1221/03/$ - see front matter © 2003 Elsevier Ltd. All rights reserved. Typeset by $\mathcal{A}\mathcal{M}\mathcal{S}$-TEX
doi: 10.1016/S0898-1221(03)00289-X

have acquired the needed competence set and are proficient in it, they will be comfortable and confident in making decisions [2,3]. Otherwise, they must acquire the needed competence set to solve the problem. In order to acquire a needed competence set to cope with a decision they face, finding appropriate learning sequences of acquiring needed single skills, the so-called competence set expansion, is very necessary. For example, the courses "Introduction to Computer Science", "Data Structures", and "Algorithms" are single skills in the needed competence set for obtaining the Bachelor's degree of Computer Sciences. A more appropriate learning sequence can be arranged as: learning "Introduction to Computers" is learned first, "Data Structures" is learned subsequently, and then "Algorithms" is learned last.

It is known that learning directly from one skill to another skill requires learning cost, which can be measured by time or money. For example, a student may spend one year to acquire "Data Structures" after he had acquired "Introduction to Computer Science". Usually, an appropriate learning sequence is a learning sequence with minimum learning costs. Moreover, a compound skill represents a collection of single skills that might be acquired by decision makers. Some useful compound skills can be added to the needed competence set for helping to acquire some single skills. For example, it seems to be easier for us to acquire "Algorithms" after both "Introduction to Computer Science" and "Data Structures" have already been acquired in comparison with the situation when only "Introduction to Computer Science" has been acquired. However, it seems that for some problems those useful compound skills are not completely known in advance. Actually, effective acquisitions of compound skills and learning costs between any two skills in the needed competence set are actually two significant tasks before expanding the competence set.

In fact, several well-known models regarding competence set expansion have focused on the development of effective methods for generating learning sequences with minimum learning cost. For example, the deduction graph with an integer programming method [4,5], the minimum spanning table method [6], the tree expansion process with an integer programming method [7], and the stage expansion method [3] have been proposed for expanding the competence set. However, compound skills and learning costs were hypothesized to be known in all these models.

Therefore, there are two purposes for this paper. The first is to propose a data mining technique to discover potentially useful compound skills from single skills by viewing each skill as a linguistic value, and the second is to obtain the learning cost between any two skills. From the viewpoint of data mining, which extracts implicit, previously unknown, and potentially useful knowledge from data [8], potentially useful compound skills extracted from single skills should be added to the needed competence set including all single skills. Furthermore, we consider that it seems to be impossible to exactly measure learning costs by either money or time since how much money or time will be spent should be determined by decision makers. Since there actually exist relationships between any two skills [5], the learning cost must reflect those relationships. Generally speaking, the larger the relationship that exists between two skills, the smaller is learning cost between these two skills. We thus propose a relationship-based method to determine learning costs.

For this, we employ a computer simulation to further demonstrate that it is possible to use those potentially useful compound skills to facilitate the acquisition of single skills through an integer programming method for expanding the competence set proposed by Li [5]. That is, when a user-specified threshold, which can reflect the preference of decision makers, of the proposed data mining technique is given for determining which compound skills are potentially useful, it is feasible to obtain learning sequences with smaller minimum learning cost by using single skills and potentially useful compound skills instead of using only single skills.

In the following sections, the notations used in this paper are first stated in Section 2. In Section 3, we introduce the concepts of the competence set and its expansion. The proposed data mining technique used for finding potentially useful compound skills is described in Section 4. The concepts of linguistic values used for describing the competence set are also presented in detail. Subsequently, the relationship-based method used for determining the learning costs is described in Section 5. Li's method is also briefly described in Section 6. A computer simulation for demonstrating the feasibility of facilitating the acquisition of six single skills is demonstrated in Section 7. Discussions and conclusions are presented in Sections 8 and 9, respectively.

2. NOTATIONS

Notations used in following sections are as follows:

K	Number of single skills	f_p	p^{th} single skill, where $1 \le p \le K$
n	Number of criteria	$\lvert s_u \rvert$	Length of the u^{th} potentially useful compound skill denoted by s_u, $2 \le \lvert s_u \rvert \le K$
c_i	i^{th} criterion for evaluating each skill, where $1 \le i \le n$		
w_i	Weight of c_i, where $1 \le i \le n$	$c(f_i, f_j)$	Learning cost for directly learning f_j from f_i, where $1 \le i, j \le K$

3. COMPETENCE SET

For each decision problem E, there is a competence set, denoted by CS(E), consisting of ideas, knowledge, information, and skills for successfully solving the problem. In addition, there exists a skill set denoted by Sk(E) that has been acquired by decision makers, and a truly needed competence set denoted by Tr(E). Roughly speaking, CS(E) is the union of Sk(E) and Tr(E). However, it may be impossible to resolve E using only Sk(E) since Sk(E) may cover partial Tr(E). That is, decision makers must acquire Tr(E) $\setminus$ Sk(E) from the existing competence set (i.e., Sk(E)) through the competence set expansion. A competence set expansion represents the way to find an effective way to generate learning sequences of acquiring the truly needed skills so that Tr(E) $\setminus$ Sk(E) can be obtained [6]. We depict the concept of the competence set expansion in Figure 1, where the shaded area is exactly Tr(E) $\setminus$ Sk(E). In this paper, for simplicity we assume that Tr(E) $\setminus$ Sk(E) only contains all single skills and potentially useful compound skills, and Sk(E) is an empty set.

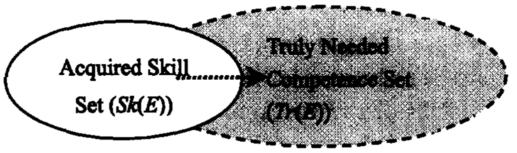

Figure 1. Competence set expansion.

As we have mentioned above, the skills in the competence set are interrelated [5]. We consider that if the relationship that exists between two single skills, say f_1 and f_3, is much larger than another two skills, say f_2 and f_3, then it is more practical to acquire f_3 from f_1 instead of from f_2. That is, the learning costs $c(f_1, f_3)$ and $c(f_2, f_3)$ must reflect individual grades of relationships such that $c(f_1, f_3) < c(f_2, f_3)$. We further interpret $c(f_i, f_j)$ to be unhelpful grades for acquiring f_j from f_i.

Sometimes, a compound skill can facilitate the acquisition of other single skills. If $c(f_1 \wedge f_2, f_3) < c(f_1, f_3) < c(f_2, f_3)$ holds, then $f_1 \wedge f_2$ can facilitate the acquisition of f_3, where $f_1 \wedge f_2$ is a compound skill and represents both f_1 and f_2 have been already acquired. In comparison with f_1 and f_2, the grade of the relationship between $f_1 \wedge f_2$ and f_3 is largest. However, if $c(f_1, f_3) < c(f_1 \wedge f_2, f_3) < c(f_2, f_3)$ or $c(f_1, f_3) < c(f_2, f_3) < c(f_1 \wedge f_2, f_3)$ holds, then we can say that $f_1 \wedge f_2$ is not helpful for acquiring f_3.

4. FIND POTENTIALLY USEFUL COMPOUND SKILLS

The proposed data mining technique for finding potentially useful compound skills is based on the well-known *a priori* property [9] introduced in the *a priori* algorithm proposed by Agrawal *et al.* [10]. In Section 4.1, we introduce concepts of data mining and the *a priori* property. The discovery of potentially useful compound skills from single skills is presented in Section 4.2. Subsequently, the proposed algorithm is completely demonstrated in Section 4.3.

4.1. Data Mining

Data mining is the exploration and analysis of large quantities of data in order to discover meaningful patterns [11]. It extracts implicit, previously unknown, and potentially useful patterns from data [8]. Recently, association rule mining has become an important research topic, and association rules have been applied to analyze market baskets to help managers determine which items are frequently purchased at the same time by customers [9,11].

The *a priori* algorithm is an influential algorithm that can be used to find association rules. In this algorithm, a candidate k-itemset ($k \geq 1$), containing k items, is considered frequent (i.e., frequent k-itemset) if its support is larger than or equal to a user-specified minimum support, which can reflect the preference of decision makers. For example, any set of fruits, say {Apple, Orange}, sold in one supermarket is a candidate two-itemset. If the purchase frequency (i.e., divide the number of transactions in which both apples and oranges were purchased by the total number of transactions) of {Apple, Orange} is larger than or equal to a user-specified minimum support, then {Apple, Orange} is a frequent two-itemset. Subsequently, we can use frequent itemsets to generate association rules. In principle, the larger the support of one frequent itemset is, the more practical is further processing (i.e., rule generation).

In addition, the *a priori* property employed in association rule mining shows that any subset of a frequent itemset must also be frequent [9]. In other words, any superset of an infrequent k-itemset cannot be frequent. For example, if {Apple, Orange, Banana} is frequent, then all its subsets (i.e., {Apple}, {Orange}, {Banana}, {Apple, Orange}, {Apple, Banana}, {Orange, Banana}) are also frequent. It is clear that the frequent itemsets are potentially useful for mining association rules. However, not all frequent itemsets can appear in the association rules consisting of antecedences and consequences. If the relationships that hold between one frequent itemset and the other frequent itemsets are not sufficiently significant, then this itemset cannot appear in any association rule.

In a similar manner, the proposed algorithm tries to find all potentially useful compound skills for expanding the competence set. Except for single skills in the needed competence set, those potentially useful compound skills discovered from the single skills by the proposed data mining technique can be added to the needed competence set. Indeed, the potentially useful compound skills are potentially useful for generating learning sequences. Actually, not all potentially useful compound skills can appear in the generated learning sequence.

4.2. Evaluation of Each Skill

Without losing generality, each element in the competence set can be viewed as a single skill. Let $f_p = (f_{p_1}, f_{p_2}, \dots, f_{p_n})$ where f_{p_i} $(0 \leq f_{p_i} \leq 1)$ is the part-worth of the i^{th} criterion with respect to f_p. That is, we assume that each single skill can be evaluated by various criteria. Furthermore, if "the skill which we desire to acquire" is a name that is specified to a linguistic variable, as originally proposed by Zadeh [12], in the universe of discourse $U = \{c_1, c_2, \dots, c_n\}$, then one possible set of names of linguistic values is $\{f_1, f_2, \dots, f_K\}$. Formally, a linguistic variable is characterized by a quintuple [13,14] denoted by $(x, T(x), U, G, M)$, in which x is the name of the variable; $T(x)$ denotes the set of names of linguistic values or terms, which are linguistic words or sentences in a natural language [15], of x; U denotes a universe of discourse; G is a syntactic rule for generating values of x; and M is a semantic rule for associating a linguistic value with a meaning (i.e., a fuzzy set). Therefore, f_p can be described as

$$f_p = \sum_{i=1}^{n} \frac{\mu_{f_p}(c_i)}{c_i} = \sum_{i=1}^{n} \frac{f_{p_i}}{c_i}, \tag{1}$$

where $\mu_{f_p}(c_i)$ is the degree to which c_i belongs to f_p. We use the well-known simple additive weighting (SAW) method, which can yield extremely close approximations to "true" value functions [16], in order to summarize the part-worths of criteria with individual weights to obtain the evaluation of f_p. That is,

$$e(f_p) = \sum_{i=1}^{n} w_i f_{p_i}, \tag{2}$$

where

$$\sum_{i=1}^{n} w_i = 1. \tag{3}$$

If a compound skill s_u consists of j single skills, then we define that the length of s_u denoted by $|s_u|$ is j. For example, $|f_1 \wedge f_2| = 2$ holds. Of course, a single skill whose length is just one can be viewed as a compound skill. From equation (1), the linguistic value $f_i \wedge f_{i+1} \wedge \cdots \wedge f_{j-1} \wedge f_j$ with length, denoted by $|f_i \wedge f_{i+1} \wedge \cdots \wedge f_{j-1} \wedge f_j|$, $j - i + 1$ can be represented as

$$f_i \wedge f_{i+1} \wedge \cdots \wedge f_{j-1} \wedge f_j = \sum_{i=1}^{n} \frac{\mu_{f_i \times f_{i+1} \times \cdots \times f_{j-1} \times f_j}(c_i)}{c_i}, \tag{4}$$

where $\mu_{f_i \times f_{i+1} \times \cdots \times f_{j-1} \times f_j}(c_i)$ is the degree to which c_i belongs to $f_i \wedge f_{i+1} \wedge \cdots \wedge f_{j-1} \wedge f_j$ and is equal to $\mu_{f_i}(c_i) \cdot \mu_{f_{i+1}}(c_i) \cdots \mu_{f_{j-1}}(c_i) \cdot \mu_{f_j}(c_i)$ [17]. Based on equation (4), the evaluation of $f_i \wedge f_{i+1} \wedge \cdots \wedge f_{j-1} \wedge f_j$ is computed as

$$e(f_i \wedge f_{i+1} \wedge \cdots \wedge f_{j-1} \wedge f_j) = \sum_{i=1}^{n} w_i \mu_{f_i \times f_{i+1} \times \cdots \times f_{j-1} \times f_j}(c_i). \tag{5}$$

We say that $f_i \wedge f_{i+1} \wedge \cdots \wedge f_{j-1} \wedge f_j$ is a "candidate" compound skill and contains $j - i + 1$ single skills (i.e., $f_i, f_{i+1}, \ldots, f_{j-1}, f_j$). Significantly, we say that $f_i \wedge f_{i+1} \wedge \cdots \wedge f_{j-1} \wedge f_j$ is "potentially useful" if $e(f_i \wedge f_{i+1} \wedge \cdots \wedge f_{j-1} \wedge f_j)$ is larger than or equal to the user-specified minimum threshold. That is, $f_i \wedge f_{i+1} \wedge \cdots \wedge f_{j-1} \wedge f_j$ can be added to the needed competence set. The threshold can indicate the tolerant or acceptable lower bound value of evaluations. Also, it can indicate the subjectivity or past experiences of decision makers. Since all single skills must be acquired, the user-specified minimum threshold must be smaller than or equal to $\min\{e(f_p) \mid 1 \leq p \leq K\}$. In fact, the terms "candidate" and "potentially useful" are derived from the concepts of "candidate" and "frequent", respectively, used in the *a priori* algorithm. We also emphasize that since each single skill is evaluated by various criteria, the important degree for evaluating each skill is defined. Moreover, the potentially useful compound skills can be further discovered.

For any two potentially useful compound skills, say $f_i \wedge f_{i+1} \wedge \cdots \wedge f_{j-1} \wedge f_j$ and $f_i \wedge f_{i+1} \wedge \cdots \wedge f_{j-1} \wedge f_j \wedge f_{j+1}$, since $\mu_{f_i \times f_{i+1} \times \cdots \times f_j \times f_{j+1}}(c_i) \leq \mu_{f_i \times f_{i+1} \times \cdots \times f_{j-1} \times f_j}(c_i)$ from (4), $f_i \wedge f_{i+1} \wedge \cdots \wedge f_{j-1} \wedge f_j \wedge f_{j+1} \subseteq f_i \wedge f_{i+1} \wedge \cdots \wedge f_{j-1} \wedge f_j$ thus holds. We may view it as a special property for mining potentially useful compound skills. It is clear that any subset of a potentially useful compound skill must also be potentially useful. It should be noted that if there exist K single skills, then the maximum length of a compound skill is at most K.

4.3. The Proposed Algorithm

In the *a priori* algorithm, two frequent $(k-1)$-itemsets are actually considered to be joined to be a candidate k-itemset, and these two frequent itemsets share $(k-2)$ items for satisfying the *a priori* property. For example, if {Apple, Orange} and {Apple, Banana} are two frequent itemsets and they share one item (i.e., Apple), then a candidate three-itemset {Apple, Orange, Banana} is generated. Of course, {Apple, Orange, Banana} can also be derived by joining {Apple, Orange} with {Orange, Banana}.

In a similar manner, a candidate compound skill with length k ($2 \leq k \leq K$) can be derived by joining two useful compound skills with length $(k-1)$, and these two useful compound skills share $(k-2)$ single skills. For example, if $f_1 \wedge f_2$ and $f_1 \wedge f_3$ are two potentially useful compound skills and they share one single skill (i.e., f_1), then a candidate compound skill $f_1 \wedge f_2 \wedge f_3$ with length 3 is generated. We describe the proposed data mining technique as follows.

```
ALGORITHM: ALGORITHM FOR FINDING POTENTIALLY USEFUL COMPOUND SKILLS.
Input:
  (a) A set of single skills
  (b) User-specified minimum threshold
Output: Potentially useful compound skills
Method:
Step 1. Initialization
  Set $k = 1$.
Step 2. Generate potentially useful compound skills with length $k$
  Step 2-1: Set $k + 1$ to $k$.
  Step 2-2:
    For any two potentially useful compound skills $s_u$ and $s_v$ ($u \neq v$) with length $(k-1)$ do
      If the following conditions are satisfied, then $s_u$ and $s_v$ can be joined to be a potentially
      useful compound skill $s_u \wedge s_v$:
      (1) $s_u$ and $s_v$ share $(k-2)$ single skills;
      (2) $s_u \wedge s_v$ has not been generated;
      (3) $e(s_u \wedge s_v)$ is larger than or equal to the minimum threshold.
    End
Step 3. Termination test
    If a potentially useful compound skill with length $k$ is generated, then return to Step 2.
    Otherwise, stop the algorithm.
```

Subsequently, we must determine the learning cost between any two skills in the needed competence set. A relationship-based method is proposed in the following section.

5. FIND LEARNING COSTS

As we have mentioned above, decision makers must determine how much money or time will be spent during the learning process. For example, one student may spend one year to acquire "Data Structures" after he or she had acquired "Introduction to Computer Science", whereas another student may spend only six months. Since skills in the competence set are strongly interrelated [5], there exist relationships between any two skills. That is, it is necessary to find the learning costs that can reflect those relationships. Intuitively, if the relationship that exists between two skills is much larger, then it is much easier to acquire one skill after the other skill has been acquired.

There exist distinct relationships between any two subsystems in the real world [18], although we do not know exactly what these relationships are. Grey theory, as proposed by Deng [18], can perform grey relational analysis for these subsystems by dealing with finite and incomplete output data series obtained from these subsystems [19]. We can view each needed skill f_p as a subsystem, and its finite output data series is just $(f_{p_1}, f_{p_2}, \ldots, f_{p_n})$. Then, the grey relations can be employed to find the learning cost between any two skills. Generally speaking, the larger the grey relation that exists between two skills, say f_p and f_i, the smaller is learning cost between f_p and f_i.

In this section, we introduce concepts of grey relations in Section 5.1. In Section 5.2, a relationship-based method is proposed for finding learning costs.

5.1. Grey Relations

Grey relational analysis is a method that can find the relationships between one major sequence and the other sequences in a given system [20]. Given one reference sequence, say f_p $(1 \leq p \leq K)$, and some comparative sequences, say f_i $(1 \leq i \leq K)$, we can easily obtain the grey relation between f_p and f_i by viewing f_p as a desired goal [21]. Formally, given the reference sequence f_p and the comparative sequences f_i with the normalized form, the grey relational coefficient (GRC) $\xi(f_{p_j}, f_{i_j})$ between f_{p_j} and f_{i_j} $(1 \leq j \leq n)$ can be computed as [19,20]

$$\xi(f_{i_j}, f_{p_j}) = \frac{\Delta_{\min} + \rho\Delta_{\max}}{\Delta_{ij} + \rho\Delta_{\max}}, \tag{6}$$

where ρ is the discriminative coefficient $(0 \leq \rho \leq 1)$, and usually $\rho = 0.5$ [20]. It should be noted that the appropriate value of ρ is dependent on requirements of individual applications. Moreover,

$$\Delta_{\min} = \min_i \min_j \left|f_{p_j} - f_{i_j}\right|, \qquad 1 \leq i \leq K, \quad 1 \leq j \leq n, \tag{7}$$

$$\Delta_{\max} = \max_i \max_j \left|f_{p_j} - f_{i_j}\right|, \qquad 1 \leq i \leq K, \quad 1 \leq j \leq n, \tag{8}$$

$$\Delta_{ij} = \left|f_{p_j} - f_{i_j}\right|, \tag{9}$$

where $|\cdot|$ denotes the absolute value. Clearly, $\xi(f_{i_j}, f_{p_j})$ is between zero and one. Then, the grey relational grade (GRG) denoted by $\Upsilon(f_i, f_p)$ can be computed as

$$\Upsilon(f_i, f_p) = \frac{1}{n}\sum_{j=1}^{n} \xi\left(f_{i_j}, f_{p_j}\right). \tag{10}$$

$0 \leq \Upsilon(f_i, f_p) \leq 1$ thus holds. The larger the value of $\Upsilon(f_i, f_p)$, the closer the relationship is between f_p and f_i. The main difference between correlation analysis and grey relational analysis is that correlation analysis measures the relationship between any two random variables, and grey relational analysis tries to find the relationships between one reference sequence and other comparative sequences by viewing the reference sequence as a desired goal that each comparative sequence expects to attain [21].

5.2. The Relationship-Based Method

First, $c(f_i, f_p)$ is heuristically computed as

$$c(f_i, f_p) = 1 - \Upsilon(f_i, f_p), \qquad 1 \leq i, p \leq K. \tag{11}$$

It is clear that $0 \leq c(f_i, f_p) \leq 1$ also holds. Basically, (11) indicates the relationship between the learning cost (i.e., $c(f_i, f_p)$) and the grade of relationship (i.e., $\Upsilon(f_i, f_p)$). An initial learning cost table can be thus built. An example of an initial learning cost table is shown as Table 1, in which we can see that there are six single skills. By using the initial learning costs table, we can subsequently compute $c(s_u, f_p)$ $(1 \leq p \leq K)$, where s_u is discovered by the proposed data mining technique.

Table 1. Initial learning cost table with six single skills.

Compound Skill	Single Skill					
	f_1	f_2	f_3	f_4	f_5	f_6
f_1	$c(f_1, f_1)$	$c(f_1, f_2)$	$c(f_1, f_3)$	$c(f_1, f_4)$	$c(f_1, f_5)$	$c(f_1, f_6)$
f_2	$c(f_2, f_1)$	$c(f_2, f_2)$	$c(f_2, f_3)$	$c(f_2, f_4)$	$c(f_2, f_5)$	$c(f_2, f_6)$
f_3	$c(f_3, f_1)$	$c(f_3, f_2)$	$c(f_3, f_3)$	$c(f_3, f_4)$	$c(f_3, f_5)$	$c(f_3, f_6)$
f_4	$c(f_4, f_1)$	$c(f_4, f_2)$	$c(f_4, f_3)$	$c(f_4, f_4)$	$c(f_4, f_5)$	$c(f_4, f_6)$
f_5	$c(f_5, f_1)$	$c(f_5, f_2)$	$c(f_5, f_3)$	$c(f_5, f_4)$	$c(f_5, f_5)$	$c(f_5, f_6)$
f_6	$c(f_6, f_1)$	$c(f_6, f_2)$	$c(f_6, f_3)$	$c(f_6, f_4)$	$c(f_6, f_5)$	$c(f_6, f_6)$

From Table 1, we can observe that if there are K single skills, then there are K^2 various learning costs in the initial learning costs table. In addition, there implicitly exists a function that can map from $(f_{i_1}, \dots, f_{i_n}, e(f_i), f_{p_1}, \dots, f_{p_n}, e(f_p))$ to $c(f_i, f_p)$. A three-layer neural network trained by the backpropagation algorithm can serve as a tool of the approximation of functions like regression [22]. We illustrate the architecture of the three-layer network employed in this paper in Figure 2. Here, we can see that there are $2n+3$ input nodes (i.e., $n+1$ nodes for f_i, $n+1$ nodes for f_p, and one bias node with input 1) and one output node. It should be noted that it has been proved that a single hidden layer is sufficient to approximate any continuous function [23]. However, how many hidden nodes are necessary is generally not known [24].

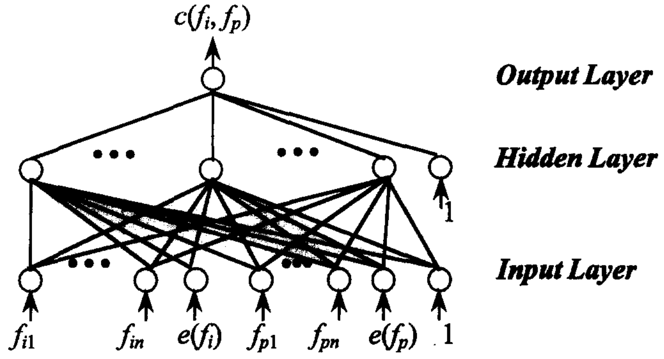

Figure 2. A three-layer neural network.

The three-layer network can be trained by using given K^2 training data from the initial learning costs table, and the optimal weights can be further determined when a termination condition is reached. Usually, we can stop the training when the mean squared error E_{ave}, shown as (12), reaches below the pregiven tolerant error

$$E_{\text{ave}} = \frac{1}{2N} \sum_{j=1}^{N} (d_j - o_j)^2, \tag{12}$$

where d_j and o_j are the actual output and the desired output of the j^{th} input training data, respectively, and N (e.g., $N = K^2$) is the number of training data. For example, if the input training data is $(f_{i_1}, \dots, f_{i_n}, e(f_i), f_{p_1}, \dots, f_{p_n}, e(f_p))$, then its desired output is $c(f_i, f_p)$. Subsequently, $c(s_u, f_p)$ $(1 \le p \le K)$ of acquiring f_p from s_u can be estimated by feeding $(s_{u_1}, \dots, s_{u_n}, e(s_u), f_{p_1}, \dots, f_{p_n}, e(f_p))$ to the trained network. But, due to the restrictions on the representation of compound skills that are described in the following section, it is not necessary to compute $c(s_u, f_p)$.

A relationship-based method proposed for obtaining learning costs by grey relations and a three-layer neural network is described as follows.

ALGORITHM: RELATIONSHIP-BASED METHOD FOR GENERATING THE LEARNING COSTS.

```
Input:
  (1) Single skills $f_i$ $(1 \le i \le K)$ and potentially useful compound skills discovered by the
      proposed data mining technique.
  (2) The number of hidden nodes used in the three-layer neural network.
Output: Learning costs table.
Method
Step 1. Generate initial learning costs table
  1.1. Calculate $\Upsilon(f_i, f_p)$ by equation (10), $1 \le i, p \le K$.
  1.2. Calculate $c(f_i, f_p)$ by equation (11), $1 \le i, p \le K$.
```

Step 2. Train the three-layer neural network

Use K^2 training data obtained from the initial learning costs table to train the three-layer neural network by the backpropagation algorithm.

Step 3. Generate complete learning costs table

Generate $c(s_u, f_p)$ by feeding $(s_{u_1}, \dots, s_{u_n}, e(s_u), f_{p_1}, \dots, f_{p_n}, e(f_p))$ to the trained neural network, $1 \le p \le K$

After the learning costs are obtained by the above-mentioned method, we must select one effective method to expand the competence set. An integer programming method proposed by Li [5] is determined to be appropriate and is introduced in the following section.

6. GENERATE LEARNING SEQUENCES

If a learning cost exists between any two skills in a competence set, then the expansion of this competence set can be categorized as a cyclic expansion problem [5]. To resolve this problem, below, we briefly describe an integer programming method proposed by Li [5], which is used to generate learning sequences with minimum learning cost in the subsequent section. For problem E, we also make several revisions to Li's method for coping with $\text{Sk}(E) = \emptyset$, since in Li's method $\text{Sk}(E)$ was assumed to be not empty.

Actually, single skills and compound skills construct a digraph G consisting of subgraphs S and T, and each skill corresponds to a node. S consists of skills in $\text{Sk}(E)$, and T consists of skills in $\text{Tr}(E) \setminus \text{Sk}(E)$. The learning sequences are generated for $\text{Tr}(E) \setminus \text{Sk}(E)$ by starting from $\text{Sk}(E)$. A node representing a compound skill, say $f_i \wedge f_{i+1} \wedge \cdots \wedge f_{j-1} \wedge f_j$, is called a compound node and is depicted as Figure 3, for example. In Figure 3, several nodes, referred to as decomposed nodes (i.e., $f_i, f_{i+1}, \dots, f_{j-1}$ and f_j), are directly linked through dotted arcs to the node representing $f_i \wedge f_{i+1} \wedge \cdots \wedge f_{j-1} \wedge f_j$. However, there are two restrictions for representing a compound node in G: the first restriction is that there is no arc between the decomposed nodes of a compound node; and the second restriction is that no nodes can be directly linked to a compound node except its decomposed nodes. For example, the node α representing f_1 cannot be directly linked to the node β representing $f_2 \wedge f_3$. In G, the other arcs are directed except dotted arcs. For example, the directed arc denoted by $r(\beta, \alpha)$ connecting the node α from the node β with cost $c(f_2 \wedge f_3, f_1)$ represents that to learn skill f_1 from $f_2 \wedge f_3$ requires $c(f_2 \wedge f_3, f_1)$ cost units.

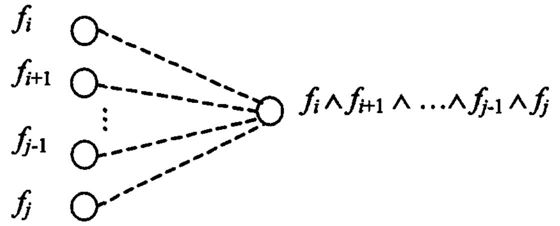

Figure 3. Compound node.

Significantly, a learning sequence can be represented by a directed path [5]. On the other hand, it is impossible to learn a single skill, say f_1, from $f_1 \wedge f_2 \wedge f_3$ since if $f_1 \wedge f_2 \wedge f_3$ is acquired, then it indicates that decision makers had already acquired f_1, f_2, and f_3. And, there also does not exist a learning cost for each dotted arc between a compound node and its decomposed nodes. For example, we can set a null value (not zero) to $c(f_1 \wedge f_2, f_1)$. Furthermore, $c(f_p, f_p)$ $(1 \le p \le K)$ also does not exist. In principle, all nodes representing single skills must take part in the generation of the final learning sequence. But only partial nodes representing potentially useful compound skills can appear in the final learning sequence since not all of them can facilitate the acquisition of single skills.

For G, several definitions must be given. First, define $|A(i)|$ and $|B(i)|$ as the numbers of nodes immediately before node i and immediately after node i, respectively. In addition, define u_i for node i as $u_i = 1$ if node i takes part in the generation of the learning sequence; otherwise $u_i = 0$. Also define $v(i,j)$ for the arc connecting node i to node j as $v(i,j) = 1$ if $v(i,j)$ is one path of the learning sequence; otherwise $v(i,j) = 0$. Let $V(S)$ and $V(T)$ be sets of nodes of S and T, respectively. Then both u_i and $v(i,j)$ are 0-1 variables and satisfy the following properties:

$$\text{(a) } u_i = 1, \text{ for each } i \in V(S) \text{ or } i \text{ corresponding to a single skill in } T; \tag{13}$$

$$\text{(b) } |A(i)|u_i \geq \sum_{j \in A(i)} v(i,j), \text{ if } i \in V(S); \tag{14}$$

$$\text{(c) } u_i \leq \sum_{j \in B(i)} v(j,i), \text{ if } i \in V(T) \text{ and } i \text{ is not a compound node}; \tag{15}$$

$$\text{(d) } (|A(i)| + |B(i)|)u_i \geq \sum_{j \in B(i)} v(j,i) + \sum_{j \in A(i)} v(i,j), \text{ if } i \in V(T). \tag{16}$$

In addition, let λ_i be the sequence number of node i in the learning sequence; then the following relations hold:

$$\text{(a) } \lambda_i = 0, \text{ if } i \in V(S), \tag{17}$$

$$\text{(b) } \lambda_i - \lambda_j + Kv(i,j) \leq K - 1, \tag{18}$$

$$\text{(c) } u_i \leq \lambda_i \leq Ku_i, \tag{19}$$

where λ_i is an integer variable. For each compound node i, if its decomposed nodes are nodes $i_1, i_2, \ldots, i_l$, then the following relations also hold:

$$\text{(a) } l\,u_i \leq \sum_{j=1}^{l} u_{i_j}; \tag{20}$$

$$\text{(b) } \lambda_i \geq \lambda_{i_j}, \text{ for } 1 \leq j \leq l. \tag{21}$$

Generally speaking, if λ_i and λ_j are the same and $\lambda_i - \lambda_p = 1$, then f_i and f_j can be learned simultaneously when f_p has been already acquired. Basically, (13)–(16) find those arcs which can be contained in the learning sequence. Equations (17)–(19) find the sequence number of each node in the learning sequence. Equations (20),(21) control the sequence of a compound node and its decomposed nodes; that is, the sequence number of a compound node is not larger than that of each of its decomposed nodes. We omit the detailed discussions of the above-mentioned properties, which can be found in [5].

Since in this paper Sk(E) is assumed to be an empty set, several revisions of Li's method must be made. Let S consist of a virtual node labeled by 0 (i.e., $V(S) = \{0\}$). Furthermore, the virtual node (i.e., node 0) is directly linked to each node i by a directed arc with learning cost being equal to zero, where node $i \in T$ and node i is not a compound node. Our purpose is to find the starting node labeled by $n_0^{(T)}$ (i.e., $n_0^{(T)} \in V(T)$) immediately after node 0 in the learning sequence. That is, $n_0^{(T)}$ corresponds to a single skill which we suggest decision makers to learn first. If nodes $i_1, i_2, \ldots, i_K$ represent single skills $f_1, f_2, \ldots, f_K$, respectively, then the following relation holds:

$$\sum_{j=1}^{K} v(0, i_j) = 1. \tag{22}$$

An integer program can be further formulated by combining (12)–(21) and giving an object function as follows:

$$\text{minimize } \operatorname{cost}(G) = \sum_{r(i,j) \in E} c(i,j) \cdot v(i,j), \tag{23}$$

where E is the set of arcs of G. In the subsequent section, we use a computer simulation to demonstrate the effectiveness of the potentially useful compound skills through Li's method.

7. SIMULATIONS

We assume that, for one decision problem E, six single skills (i.e., f_1, f_2, f_3, f_4, f_5, and f_6) must be acquired, as shown in Table 2, where we can see that each skill is evaluated by ten criteria. For simplicity, all criteria have equal weights (i.e., $w_i = 0.1$, $1 \leq i \leq 10$). The evaluation of each single skill can be obtained by (2). In real application, the part-worths of criteria with respect to each single skill and the weights of criteria may be obtained by questionnaire. In addition, decision makers can determine their own preferable minimum threshold.

Table 2. Six single skills.

Skill	Criterion										Evaluation
	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}	
f_1	0.8571	0.7143	0.8571	0.3657	0.8857	0.9429	0.7714	0.6857	0.4086	0.7231	0.723
f_2	0.8286	0.7143	0.7143	0.7786	0.8286	0.8571	0.7143	0.9529	0.6157	0.7804	0.780
f_3	0.7714	0.8286	0.7714	0.7786	0.7143	0.8286	0.6571	0.7500	0.7143	0.7529	0.753
f_4	0.8571	0.6857	0.8571	0.5814	0.8857	0.8857	0.6786	0.6786	0.7500	0.7653	0.765
f_5	0.9143	0.7429	0.8571	0.5914	0.8571	0.7143	0.8571	0.7857	0.6071	0.7557	0.756
f_6	0.9714	0.9143	0.8571	0.6529	0.7143	0.6857	0.8214	0.6857	0.9057	0.7894	0.789

Train a three-layer neural network for estimating the learning costs

The relationship-based method described in Section 5 is used to obtain learning costs. A three-layer network can be trained by using training data from the initial learning costs table. To obtain a sufficiently large size of training data, we heuristically add another two virtually single skills to play an auxiliary role. That is, the part-worth of each criterion is zero for one skill, and it is one for the other skill. To obtain the initial learning cost table, these two skills serve as comparative sequences for other six single skills (i.e., f_1, f_2, f_3, f_4, f_5, and f_6). The initial learning cost table can be thus obtained by (9) and (10) and is shown as Table 3, although those two virtual skills are omitted in this table. It is clear that there are 48 training data (i.e., $6^2 + 2 \times 6$) used for training the neural network with 23 input nodes (i.e., $2 \times 10 + 3$) and ten hidden nodes. In addition, when E_{ave} is smaller than 0.0005, we terminate the progress of training. The trained network is subsequently used for estimating the learning costs $c(s_u, f_p)$ $(1 \leq p \leq 6)$ if all s_u have been obtained by the proposed learning algorithm described in Section 4.3.

Table 3. Initial learning cost table.

Computer Skill	Single Skill					
	f_1	f_2	f_3	f_4	f_5	f_6
f_1	0.000	0.189	0.241	0.120	0.179	0.217
f_2	0.190	0.000	0.155	0.173	0.188	0.224
f_3	0.220	0.139	0.000	0.172	0.200	0.173
f_4	0.112	0.160	0.177	0.000	0.155	0.186
f_5	0.174	0.182	0.216	0.163	0.000	0.148
f_6	0.221	0.228	0.196	0.204	0.155	0.000

Generate a learning sequence with minimum cost for only single skills

We now obtain the potentially useful compound skills. At first, the user-specified minimum threshold is set to 0.65. This is a valid value since $\min\{e(f_p) \mid 1 \leq p \leq 6\} = e(f_1) = 0.723$. We find that no potentially useful compound skills can be found from the six single skills. Let f_i be the i^{th} node $(1 \leq i \leq 6)$ in G; a mathematical integer program of finding a learning sequence with minimum learning cost is thus formulated below.

$$\begin{aligned}\text{Minimize } \operatorname{cost}(G) = {} & 0.189v(1,2) + 0.241v(1,3) + 0.120v(1,4) + 0.179v(1,5) + 0.217v(1,6)\\ & + 0.190v(2,1) + 0.155v(2,3) + 0.173v(2,4) + 0.188v(2,5) + 0.224v(2,6)\\ & + 0.220v(3,1) + 0.139v(3,2) + 0.172v(3,4) + 0.200v(3,5) + 0.173v(3,6)\\ & + 0.112v(4,1) + 0.160v(4,2) + 0.177v(4,3) + 0.155v(4,5) + 0.186v(4,6)\\ & + 0.174v(5,1) + 0.182v(5,2) + 0.216v(5,3) + 0.163v(5,4) + 0.148v(5,6)\\ & + 0.221v(6,1) + 0.228v(6,2) + 0.196v(6,3) + 0.204v(6,4) + 0.155v(6,5),\end{aligned}$$

$$\text{s.t. } u_i = 1, \quad 0 \le i \le 6,$$

$$6u_0 \ge \sum_{i=1}^{6} v(0,i),$$

$$\sum_{i=1}^{6} v(0,i) = 1,$$

for $1 \le i \le 6$

$$\text{(a)} \quad \sum_{j=0}^{6} v(j,i) \ge 1, \qquad i \ne j;$$

$$\text{(b)} \quad \sum_{j=0}^{6} v(j,i) + \sum_{j=1}^{6} v(i,j) \le 11, \qquad i \ne j.$$

$$\lambda_0 = 0,$$

$$\lambda_i - \lambda_j + 6v(i,j) \le 5, \qquad 0 \le i \le 6, \quad 1 \le j \le 6 \quad \text{and} \quad i \ne j,$$

$$1 \le \lambda_i \le 6, \quad 1 \le i \le 6.$$

Solve the above program by the LINGO package to obtain the optimal solution as

$$v(0,3) = v(3,2) = v(3,4) = v(4,1) = v(4,5) = v(5,6) = 1,$$

$$\lambda_3 = 1, \quad \lambda_2 = 2, \quad \lambda_4 = 2, \quad \lambda_1 = 3, \quad \lambda_5 = 3, \quad \lambda_6 = 4.$$

Since $\lambda_3 = 1$, therefore $n_0^{(T)} = 3$ holds. That is, f_3 is suggested for decision makers to learn first among the six single skills. The learning sequence with minimum learning cost of 0.726 is depicted in Figure 4. We can see that f_2 and f_4 can be learned simultaneously when f_3 has already been acquired. We can also observe that although $\max\{e(f_p) \mid 1 \le p \le 6\} = e(f_6) = 0.789$, f_6 is not suggested to be learned first. It is noted that the minimum learning cost can be regarded as the minimum degree of the difficulty for acquiring all single skills.

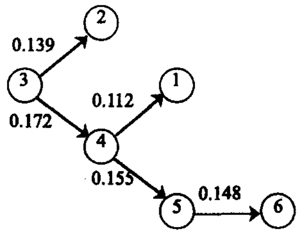

Figure 4. Learning sequence for minimum threshold 0.65.

Find potentially useful compound skills

We now try to set a smaller value, 0.60, to the user-specified minimum threshold. Three potentially useful compound skills, $f_2 \wedge f_6$ with $e(f_2 \wedge f_6) = 0.614$, $f_4 \wedge f_6$ with $e(f_4 \wedge f_6) = 0.606$, and $f_5 \wedge f_6$ with $e(f_5 \wedge f_6) = 0.601$ can be further discovered from the six single skills. For example, since $f_2 \wedge f_6 = 0.8049/c_1 + 0.6531/c_2 + 0.6122/c_3 + 0.5083/c_4 + 0.5918/c_5 + 0.5878/c_6 + 0.5867/c_7 + 0.6534/c_8 + 0.4222/c_9 + 0.7246/c_{10}$, therefore $e(f_2 \wedge f_6) = 0.10 \cdot (0.8049 + 0.6531 + 0.6122 + 0.5083 + 0.5918 + 0.5878 + 0.5867 + 0.6534 + 0.4222 + 0.7246) = 0.614 \geq 0.60$ holds. We then employ the trained neural network to estimate the learning cost, with the results as shown in Table 4. For example, the estimation of $c(f_2 \wedge f_6, f_1) = 0.171$ can be obtained by feeding (0.8049, 0.6531, 0.6122, 0.5083, 0.5918, 0.5878, 0.5867, 0.6534, 0.4222, 0.7246, 0.614, 0.8571, 0.7143, 0.8571, 0.3657, 0.8857, 0.9429, 0.7714, 0.6857, 0.4086, 0.7231, 0.723) to the trained network. By displaying nothing in the corresponding location, some null values are given.

Table 4. Learning cost table for potentially useful compound skills.

Compound Skill	Single Skill					
	f_1	f_2	f_3	f_4	f_5	f_6
$f_2 \wedge f_6$	0.171		0.210	0.173	0.163	
$f_4 \wedge f_6$	0.327	0.327	0.358		0.281	
$f_5 \wedge f_6$	0.324	0.396	0.357	0.357		

Generate learning sequences by potentially useful compound skills

Let $f_2 \wedge f_6$, $f_4 \wedge f_6$, and $f_5 \wedge f_6$ be the seventh, eighth, and ninth nodes in G, respectively; a mathematical integer program of finding a learning sequence with minimum learning cost is thus formulated below.

$$\begin{aligned}
\text{Minimize } \operatorname{cost}(G) = {} & 0.189v(1,2) + 0.241v(1,3) + 0.120v(1,4) + 0.179v(1,5) + 0.217v(1,6), \\
& + 0.190v(2,1) + 0.155v(2,3) + 0.173v(2,4) + 0.188v(2,5) + 0.224v(2,6) \\
& + 0.220v(3,1) + 0.139v(3,2) + 0.172v(3,4) + 0.200v(3,5) + 0.173v(3,6) \\
& + 0.112v(4,1) + 0.160v(4,2) + 0.177v(4,3) + 0.155v(4,5) + 0.186v(4,6) \\
& + 0.174v(5,1) + 0.182v(5,2) + 0.216v(5,3) + 0.163v(5,4) + 0.148v(5,6) \\
& + 0.221v(6,1) + 0.228v(6,2) + 0.196v(6,3) + 0.204v(6,4) + 0.155v(6,5) \\
& + 0.171v(7,1) + 0.210v(7,3) + 0.173v(7,4) + 0.163v(7,5) \\
& + 0.327v(8,1) + 0.327v(8,2) + 0.358v(8,3) + 0.281v(8,5) \\
& + 0.324v(9,1) + 0.396v(9,2) + 0.357v(9,3) + 0.357v(9,4),
\end{aligned}$$

$$\text{s.t. } u_i = 1, \qquad 0 \leq i \leq 6,$$

$$6u_0 \geq \sum_{i=1}^{6} v(0,i),$$

$$\sum_{i=1}^{6} v(0,i) = 1,$$

for $i = 1$,

$$\sum_{j=0}^{9} v(j,i) \geq 1, \qquad j \neq 1,$$

$$\sum_{j=0}^{9} v(j,i) + \sum_{j=1}^{6} v(i,j) \leq 14, \qquad j \neq 1;$$

for $i = 2$,

$$\sum_{j=0}^{6} v(j,i) + v(8,2) + v(9,2) \geq 1, \qquad j \neq 2,$$

$$\sum_{j=0}^{6} v(j,i) + v(8,2) + v(9,2) + \sum_{j=1}^{6} v(i,j) \leq 13, \qquad j \neq 2;$$

for $i = 3$,

$$\sum_{j=0}^{9} v(j,i) \geq 1, \qquad j \neq 3,$$

$$\sum_{j=0}^{9} v(j,i) + \sum_{j=1}^{6} v(i,j) \leq 14, \qquad j \neq 3;$$

for $i = 4$,

$$\sum_{j=0}^{6} v(j,i) + v(7,4) + v(9,4) \geq 1, \qquad j \neq 4,$$

$$\sum_{j=0}^{6} v(j,i) + v(7,4) + v(9,4) + \sum_{j=1}^{6} v(i,j) \leq 13, \qquad j \neq 4;$$

for $i = 5$,

$$\sum_{j=0}^{6} v(j,i) + v(7,5) + v(8,5) \geq 1, \qquad j \neq 5,$$

$$\sum_{j=0}^{6} v(j,i) + v(7,5) + v(8,5) + \sum_{j=1}^{6} v(i,j) \leq 13, \qquad j \neq 5;$$

for $i = 6$,

$$\sum_{j=0}^{6} v(j,i) \geq 1, \qquad j \neq 6,$$

$$\sum_{j=0}^{6} v(j,i) + \sum_{j=1}^{6} v(i,j) \leq 11, \qquad j \neq 6.$$

$$\lambda_0 = 0,$$
$$\lambda_i - \lambda_j + 6v(i,j) \leq 5, \qquad 0 \leq i \leq 6, \quad 1 \leq j \leq 6, \quad \text{and} \quad i \neq j,$$
$$1 \leq \lambda_i \leq 6, \qquad 1 \leq i \leq 6,$$

$\lambda_7 - \lambda_1 + 6v(7,1) \leq 5$; $\lambda_7 - \lambda_3 + 6v(7,3) \leq 5$; $\lambda_7 - \lambda_4 + 6v(7,4) \leq 5$; $\lambda_7 - \lambda_5 + 6v(7,5) \leq 5$;
$\lambda_8 - \lambda_1 + 6v(8,1) \leq 5$; $\lambda_8 - \lambda_2 + 6v(8,2) \leq 5$; $\lambda_8 - \lambda_3 + 6v(8,3) \leq 5$; $\lambda_8 - \lambda_5 + 6v(8,5) \leq 5$;
$\lambda_9 - \lambda_1 + 6v(9,1) \leq 5$; $\lambda_9 - \lambda_2 + 6v(9,2) \leq 5$; $\lambda_9 - \lambda_3 + 6v(9,3) \leq 5$; $\lambda_9 - \lambda_4 + 6v(9,4) \leq 5$.

$$\begin{array}{lll} 2u_7 \le u_2 + u_6; & \lambda_7 \ge \lambda_1; & \lambda_7 \ge \lambda_6, \\ 2u_8 \le u_4 + u_6; & \lambda_8 \ge \lambda_4; & \lambda_8 \ge \lambda_6; \\ 2u_9 \le u_5 + u_6; & \lambda_9 \ge \lambda_5; & \lambda_9 \ge \lambda_6. \end{array}$$

However, the optimal solution of this program is the same as the case that sets the minimum threshold to 0.65. This result indicates that the current potentially useful compound skills (i.e., $f_2 \wedge f_6$, $f_4 \wedge f_6$, and $f_5 \wedge f_6$) are not sufficient to reduce the minimum learning cost obtained by the case that sets the minimum threshold to 0.65. That is, it seems that the minimum threshold may be too large to find more potentially useful compound skills. In the following, we thus pay attention to how the minimum threshold can affect the obtainable minimum learning costs.

Obtain minimum learning costs for various minimum thresholds

Among all potentially useful compound skills, we can intuitively observe that $f_1 \wedge f_2 \wedge f_3 \wedge f_4 \wedge f_5 \wedge f_6$ is not useful for facilitating the acquisition of single skills, as we have mentioned above, since if $f_1 \wedge f_2 \wedge f_3 \wedge f_4 \wedge f_5 \wedge f_6$ is acquired, then it indicates that decision makers had already acquired all six single skills. To show various minimum thresholds could affect the minimum learning costs, the minimum threshold is arranged at intervals of 0.05 from 0.55 to 0.25, and the simulation results are shown in Table 5. From this table, we can see that nearly all (i.e., 62) potentially useful compound skills can be discovered when the user-specified minimum threshold is 0.25. It seems that the minimum learning cost, obtained by the case sets the minimum threshold to 0.65, can be decreased as the minimum threshold gradually decreases. However, the obtainable minimum learning cost can stay at a stable point when the minimum threshold is smaller than or equal to one real value.

Table 5. Learning cost table for potentially useful compound skills.

Minimum Threshold	Number	Minimum Learning Cost
0.65	6	0.726
0.60	9	0.726
0.55	20	0.480
0.50	21	0.480
0.45	32	0.474
0.40	41	0.414
0.35	49	0.414
0.30	56	0.414
0.25	62	0.414

However, decision makers must have their own preferable threshold in real applications. When the minimum threshold is set to a preferable value, say 0.50, a learning sequence depicted in Figure 5 with minimum learning cost 0.480 can be obtained. The learning sequence with minimum learning cost 0.414, when the threshold is set to 0.35, is also depicted in Figure 6. Further issues are discussed below.

8. DISCUSSION

The main contribution of this paper is to develop a practical model to generate learning sequences of acquiring needed single skills by potentially useful compound skills discovered by the proposed data mining technique to support the decision making. In addition, we provide a more reasonable method to measure the learning cost between any two skills by grey relational analysis.

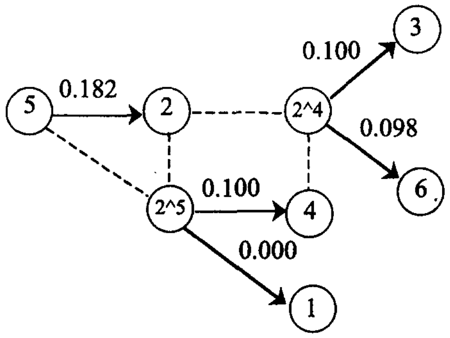

Figure 5. Learning sequence for minimum threshold 0.50.

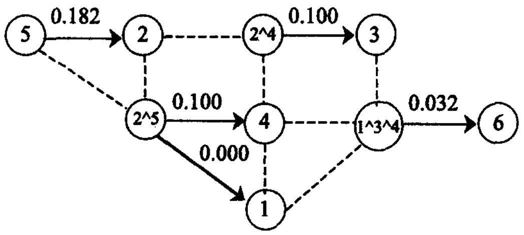

Figure 6. Learning sequence for minimum threshold 0.35.

Significantly, this is a starting point for integrating data mining techniques with the expansion of the competence set.

The generated learning sequence corresponding to a user-specified threshold is a sequence that should be more appropriate for decision makers to progressively acquire required skills. From the simulation results, we can see that it is possible to obtain smaller minimum learning cost (e.g., 0.414) by using those potentially useful compound skills in comparison with the case that uses only single skills (i.e., 0.726), when a user-specified minimum threshold is given (e.g., 0.35). If decision makers only pay attention to the issue of obtaining a learning sequence with a cost as small as possible, then the minimum threshold should be set to a smaller value (e.g, 0.10).

Indeed, there exists a trade-off between the required efficiency and the expected optimality. When the number of single skills is too large and the minimum threshold is too small, the processing time may suffer from performing the proposed data mining technique and Li's method. In addition, the generated learning sequence may be inappropriate for decision makers. If almost all compound skills are potentially useful, then only a multilayer neural network must be trained to obtain learning costs by giving an initial learning cost table. In real applications, decision makers should give a reasonable minimum threshold, which can indicate the tolerant or acceptable lower bound value of skills' evaluations. That is, they should determine their own preferable threshold depending on their past experiences or subjectivity.

For evaluating each skill, SAW, which assumes that any two criteria are independent, is one of the possible methods. However, it seems that any two criteria are interrelated. Therefore, the methods other than SAW which admit existing the dependent criteria, such as the fuzzy integral approach [25], can be considered to evaluate each skill in the proposed data mining technique.

On the other hand, it seems that our methods for acquiring compound skills and learning costs could also be widely applied in various problems. For example, if there are 20 projects that must be executed to attain one goal, then we can view these projects as single skills. Then, a generated learning sequence with minimum learning cost can be further viewed as the working sequence. It seems to be inappropriate to arrange a sequence only according to the evaluations, since the evaluations represent the individual significance of skills rather than the relative working order in the sequence.

9. CONCLUSIONS

When decision makers have already acquired the competence set to solve a decision problem they face, they will be confident in making decisions. Otherwise, they must expand the needed competence set from the acquired competence set to resolve the problem.

As we have mentioned above, effective acquisitions of potentially useful compound skills and learning costs are two significant tasks before expanding the competence set. However, in previous models, compound skills and learning costs were hypothesized to be known. We thus propose a data mining technique based on the well-known *a priori* property and a relationship-based method based on grey relations to discover compound skills and determine learning costs, respectively. In addition, the relationship-based method provides a reasonable way to give the learning costs by measuring the grade of the relationship between any two skills rather than by money or time.

Then, we employ a known integer programming method proposed by Li [5] to generate learning sequences. From the simulation results, we can see that it is more feasible to obtain a learning sequence with smaller minimum learning cost by using those potentially useful compound skills instead of using only single skills, when a threshold of the proposed data mining technique is given by decision makers for determining which compound skills are potentially useful.

In the discussions, we also suggest several approaches to improve the proposed approach, such as the evaluation of one skill. In addition, it seems that our methods could also be applied in various problems.

REFERENCES

1. P.L. Yu and D. Zhang, A foundation for competence set analysis, *Mathematical Social Sciences* **20** (3), 251–299, (1990).
2. M.J. Hwang, C.I. Chiang, I.C. Chiu and G.H. Tzeng, Multi-stages optimal expansion of competence sets in fuzzy environment, *International Journal of Fuzzy Systems* **3** (3), 486–492, (2001).
3. J.M. Li, C.I. Chiang and P.L. Yu, Optimal multiple stage expansion of competence set, *European Journal of Operational Research* **120** (3), 511–524, (2000).
4. H.L. Li and P.L. Yu, Optimal competence set expansion using deduction graph, *Journal of Optimization Theory and Application* **80** (1), 75–91, (1994).
5. H.L. Li, Incorporating competence sets of decision makers by deduction graphs, *Operations Research* **47** (2), 209–220, (1999).
6. J.W. Feng and P.L. Yu, Minimum spanning table and optimal expansion of competence set, *Journal of Optimization Theory and Application* **99** (3), 655–679, (1998).
7. D.S. Shi and P.L. Yu, Optimal expansion and design of competence sets with asymmetric acquiring costs, *Journal of Optimization Theory and Application* **88** (3), 643–658, (1996).
8. P. Adriaans and D. Zantinge, *Data Mining*, Addison-Wesley, Harlow, (1996).
9. J.W. Han and M. Kamber, *Data Mining: Concepts and Techniques*, Morgan Kaufmann, San Francisco, CA, (2001).
10. R. Agrawal, H. Mannila, R. Srikant, H. Toivonen and A.I. Verkamo, Fast discovery of association rules, In *Advances in Knowledge Discovery and Data Mining*, (Edited by U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth and R. Uthurusamy), AAAI Press, Menlo Park, CA, (1996).
11. M. Berry and G. Linoff, *Data Mining Techniques: For Marketing, Sales, and Customer Support*, John Wiley & Sons, New York, (1997).
12. L.A. Zadeh, The concept of a linguistic variable and its application to approximate reasoning, *Information Science* (Part 1) **8** (3), 199–249, (1975); (Part 2) **8** (4), 301–357, (1975).
13. H.J. Zimmermann, *Fuzzy Set Theory and Its Applications*, Kluwer, Boston, MA, (1991).
14. W. Pedrycz and F. Gomide, *An Introduction to Fuzzy Sets: Analysis and Design*, MIT Press, Cambridge, MA, (1998).
15. S.M. Chen and W.T. Jong, Fuzzy query translation for relational database systems, *IEEE Transactions on Systems, Man, and Cybernetics* **27** (4), 714–721, (1997).
16. K.P. Yoon and C.L. Hwang, *Multiple Attribute Decision Making: An Introduction*, Sage Publications, London, (1995).
17. Y.C. Hu, R.S. Chen and G.H. Tzeng, Generating learning sequences for decision makers through data mining and competence set expansion, *IEEE Transactions on Systems, Man, and Cybernetics (Part B)* **32** (5), 679–686, (2002).
18. J.L. Deng, Control problems of grey systems, *Systems and Control Letters* **1** (5), 288–294, (1982).
19. Y.P. Huang and C.H. Huang, Real-valued genetic algorithms for fuzzy grey prediction system, *Fuzzy Sets and Systems* **87** (3), 265–276, (1997).

20. Y.T. Hsu and C.M. Chen, A novel fuzzy logic system based on N-version programming, *IEEE Transactions on Fuzzy Systems* **8** (2), 155–170, (2000).
21. Y.C. Hu, R.S. Chen, Y.T. Hsu and G.H. Tzeng, Grey self-organizing feature maps, *Neurocomputing* **48** (1), 863–877, (2002).
22. K.A. Smith and J.N.D. Gupta, Neural networks in business: Techniques and applications for the operations researcher, *Computers and Operations Research* **27** (11/12), 1023–1044, (2000).
23. G. Cybenko, Approximation by superpositions of a sigmoidal function, *Mathematics of Control, Signals, and Systems* **2**, 303–314, (1989).
24. J. Hertz, A. Krogh and R.G. Palmer, *Introduction to the Theory of Neural Computation*, Addison-Wesley, Redwood City, CA, (1991).
25. T. Murofushi and M. Sugeno, Some quantities represented by the Choquet integral, *Fuzzy Sets and Systems* **56**, 229–235, (1993).