Adv. Studies Theor. Phys., Vol. 5, 2011, no. 15, 725 - 732

Optimal Competence Set Expansion

Anurag Sharma

Dept. of Mathematics, R.B.S. College, Agra, U.P., India
dranurag_2007@rediffmail.com

Rajeev Pandey

Dept. of Computer Science, R. G. P. Vishwavidalaya
Bhopal M.P. India
rajeev98iet@gmail.com

Abstract. A competence set is a collection of skills used to solve a problem. Based on deduction graph concepts, this paper proposes a method of finding an optimal process so as to expand a decision maker's competence set to enable him to solve his problem confidently. Using the concept of minimum spanning tree, Yu and Zhang addressed the problem of the optimal expansion of competence sets. In contrast, the method proposed here enjoys the following advantages: it can deal with more general problems involving intermediate skills and compound skills; it can find the optimal solution by utilizing a 0-1 integer program; and it can be directly extended to treat multilevel competence set problems, and thus is more practically useful.

Keywords: Branch-and-bound methods, competence sets, deduction graphs.

1. Introduction

For each significant decision problem, there is a competence set consisting of ideas, knowledge, information, and skills for its satisfactory solution. When the decision maker (DM) thinks that he has acquired and mastered the competence set as perceived, he will be confident and quick in making the decision. Otherwise, the DM may want to expand his competence set. One important function of decision aid is to identify the true competence set and the DM's already acquired competence set, and then help the DM to effectively expand the true competence set from the acquired competence set, thus allowing him to confidently make a good decision[3]. This concept of expanding a competence set can be incorporated into computer expert systems when the true competence set is finite and discrete [2].

In order to make our presentation precise, let *Tr* be the true competence set for a particular problem *E* (think of a student who needs to successfully take a number of courses and trainings in order to get a degree or certificate); let *Sk* be the DM's already acquired skills or competence set (think of those classes and trainings that have been successfully taken by the student); and let *HD* (habitual domains) be the set of skills related to solving the problem *E* including *Tr* and *Sk* (those courses and trainings that could possibly be relevant to the acquirement of the degree or certificate). How do we help the DM to effectively reach *Tr* based on *Sk*? When *HD* = *Tr*, and the cost of acquiring new skill x is determined by

$$c(Sk, x) = \min\{c(a, x) \mid a \in Sk\} \quad ,$$

where $c(a, x)$ is the cost of acquiring x directly from a, an efficient minimum spanning tree method can be used to identify the optimal expansion processes in the sense of minimizing the total cost of expansion and in the sense of lexicographical ordering[1-2].

However, in many applications, it may occur that $HD \supset Tr$ and $HD \neq Tr$. That is, there are intermediate skills or courses which are not needed in *Tr*, but could help the DM to speed up the learning (for instance, calculus may not be required for an Art degree but it may speed up the learning of other courses). In addition, due to the fact that a new concept may be more easily acquired through some subsets of *Sk*, the cost of acquiring new skill *x*, while depending on the current *Sk*, may not follow the rule of minimization. These observations make the applications of the Yu-Zhang method somehow limited.

In this article, we shall remove the above two restrictive conditions and show how the optimal expansion problem can be formulated and solved by 0-1 integer program using the concept of deduction graph discussed in [4-5]. More specifically, in the next section we shall discuss how the expansion processes can be represented by deduction graphs. We shall show that the minimum cost expansion process is the one corresponding to the minimum cost deduction graph, which leads to a 0-1 linear integer program as discussed in section 3.

2. Expansion Processes and Deduction Graphs

In order to facilitate our discussion, let us consider the following example.

Example 2.1. In order to become a certified regional sales representative of a medical instrument company, a trainee needs to have seven basic skills, $Tr = \{a, b, c, d, f, g, h\}$, with a representing product functioning, b pricing, c cost, d service/warranty, f salesmanship, g negotiation skills, and h closing skills, respectively. In addition, finance skills, designated by e, although not required for certification, can speed up the learning of other skills such as negotiation skills g and closing skills h. Assume that the trainee has already acquired the skills of

Sk = {a, b, c}, and that the time needed in various sequences of learning and acquiring the other skills is given by table 1.

In table 1, assume that the unit of time is a month. The rows are the given skills and the columns are those to be acquired. Thus, given skill a, it takes 1, 3, 4 months to learn d, f, g, respectively. The empty cells indicate that it is practically impossible (either because it is too expensive or in the wrong sequence) to acquire the column skill from corresponding row skills. Thus, it is practically impossible to learn e immediately after a or b is acquired. Also, the column for skill f indicates that, to learn f, it takes 3 time units and 1.8 time units immediately after a and d, respectively, but it takes 2.5 time units if a and b are already acquired; and that it is unpractical to acquire f from other combinations of skills listed in the rows. Observe that $e \notin Tr$, but e can facilitate the learning programs; Table 1 shows that learning *g or h* from *e* and *f* is faster than learning that from *f* directly. Thus, $e \in HD$.

	d	***e***	***f***	***g***	***h***
a	1		3	4	
b	2			4	
c	2	3			
a^b	1		2.5	3.5	
d		1	1.8		
e					4
f				2	2
e^f				1.5	1.5

Table 1. Time requirement for new skills.

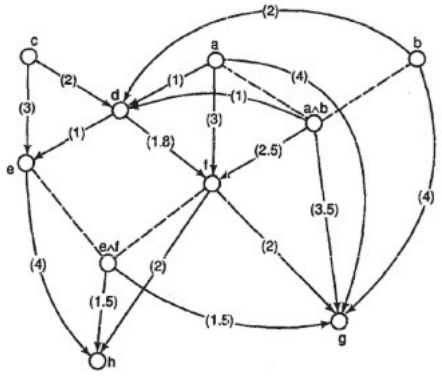

Figure : 1. Learning orders for Example 2.1.

The above information can be more vividly expressed as in figure 1. Note that the number on each directed arc represents the time units required to learn the new skill. For instance, 3 on the arc connecting f from a means that it requires 3 time units to learn f from *a*. There is no arc connecting g directly from c, meaning that it is practically impossible to learn *g* from c directly. A directed path represents

a learning sequence. For instance, the path $c \to d \to f \to g$ represents the learning sequence of learning *c*, then *d*, then *f*, and then *g*. Observe that nodes such as $a \wedge b$ and $e \wedge f$ are later to be defined as compound nodes, which can be activated if and only if all of its precedent nodes (*a* and *b* for $a \wedge b$; *e* and *f* for $e \wedge f$) are activated. In figure 1, the compound nodes are connected with their precedent nodes by dotted lines to specifically mean this effect.

Given a node i in a graph, define $B(i)$ as the set of nodes immediately before i in the graph and $A(i)$ as the set of nodes immediately after i in the graph. For instance in *figure 1*,

$$B(d) = \{a, b, c, a \wedge b\} \quad , \quad A(d) = \{e, f\}$$

The arc connecting j from i will be denoted by $r(i, j)$, and the corresponding cost of reaching j from i will be denoted by $c(i, j)$. For instance, in *figure 1*,

$$c(a, f) = 3$$

For simplicity, from now on, we shall denote *Sk* and *Tr\Sk* simply by *S* (source) and *T* (target), respectively. Recall that *HD* is the collection of all relevant skills including *Sk* and *Tr* (simply called the discussion domain).

Observe that all possible learning connections, such as those of *figure 1*, can be represented by a collection of arcs which connect a subset of *HD* from another subset of *HD*, meaning acquisition of a subset of skills from another subset of skills. For convenience, we shall use $G_\infty(S,T)$ to denote the original graph, which is the collection of all learning connections among the subsets of *HD*. We shall assume that $G_\infty(S,T)$ contains no cycle.

Definition 2.1. *A node i_0 in $G_\infty(S,T)$ is a compound node if i_0 can be decomposed into { i_1 , , . . . , i_k } and $B(i_0)$ = {i_1,,...,..i_k }. Furthermore, each $i \in B(i_0)$ and i_0 is connected by $r(i_t, i_0)$. Denote by $i_0 = i_1 \wedge ..... \wedge i_k$, and $r(i_t, i_0)$ is represented by a dotted line; see figure 2.*

Definition 2.2. *A subgraph of $G_\infty(S,T)$ is legitimate if it satisfies the condition that, if the subgraph contains any compound node i_0 of $G_\infty(S,T)$, then it also contains all nodes of $B(i_0)$ and all arcs $r(i_t, i_0)$, $i_t \in B(i_0)$.*

Definition 2.3. *A deduction graph from S to T, denoted by DG(S, T), is a legitimate subgraph of $G_\infty(S,T)$ such that each node of T is contained in DG(S, T). Furthermore, each node other than in S is connected by a directed path in DG(S, T) from some node of S.*

Note that, by definition, as each node other than those of S is connected by a directed path from some node of *S*, the corresponding skill of the node is then sequentially acquired along the path. Thus, a deduction graph from *S* to *T* represents a feasible plan to acquire all skills of *T* from the initial skill set *S*. Conversely, any such feasible plan can be represented by a deduction graph *DG(S, T)*.

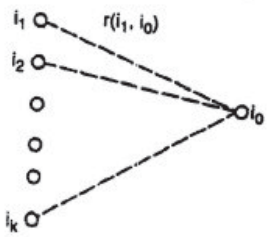

Figure : 2. Compound node.

Our original problem of finding the most effective plan to acquire T from S can be restated as follows: find the deduction graph from the subgraphs of $G_\infty(S, T)$ which minimizes the total cost or time.

Remark 2.1. In order to make our problem meaningful, from now on let us assume that, for each node $i \in S$, there is a directed path in $G_\infty(S,T)$ connecting i from some node of S. This assumption means that each i is attainable from S. Observe that this assumption means that $G_\infty(S, T)$ itself is a deduction graph from S to T and $B(i) \neq \varnothing$ if $i \notin S$.

3. Mathematical Programs for Solving the Problems

To facilitate our discussion, let N be the collection of all nodes, and let A be the collection of all arcs in $G_\infty(S, T)$.

Definition 3.1. *Given any subgraph G of $j \in A(i)$, define for each $i \in N$,*

$$x_i = \begin{cases} 1, & \text{if } i \in G, \\ 0, & \text{if } i \notin G, \end{cases}$$

and for each arc $r(i, j) \in A$, define

$$y(i, j) = \begin{cases} 1, & \text{if } r(i, j) \in G, \\ 0, & \text{if } r(i, j) \notin G, \end{cases}$$

Remark 3.1. As the directed arc $r(i, j)$ means connecting j from i, when $r(i, j)$ is in G, so are i and j. Thus, if $y(i, j) = 1$, then $x_i = x_j = 1$; and if $x_i = 0$, then $y(j, i) = 0$ for all $j \in B(i)$ and $y(i, j) = 0$ for all $j \in A(i)$. This condition is fulfilled by imposing condition (iii)(b) of the following lemma.

Lemma 3.1. *A subgraph G of $G_\infty(S, T)$ is a deduction graph from S to T if and only if its corresponding variables $\{ x_i \}$ and $\{y(i, j)\}$ assume values of 0 or 1 and satisfy the following conditions:*

(i) for all $i \in T$, $x_i = 1$;

(ii) for each compound node i with $B(i) = \{i_1,,..., i_k\}$, if $x_i = 1$ then all $x_{i_t} = 1$, $i_t \in B(i)$;

(iii) for each node $i \in N$, the following relations hold:

$$(a) \quad x_i \le \sum_{j \in B(i)} y(j,i), \text{ if } i \notin S;$$

$$(b) \quad \left(\left[A(i)\right]+\left[B(i)\right]\right) x_i \ge \sum_{j \in B(i)} y(j,i) + \sum_{j \in A(i)} y(i,j) \ ,$$

where [a(i)] and [B (i)] are the numbers of elements in A(i) and B(i), respectively, and the corresponding summations in (iii) are zeros if A(i) = ∅ or B(i) = ∅.

Proof. **Necessity**: From definitions 2.2-2.3, we see that (i) and (ii) must hold. Now, we show that (iii) also holds.

Given any node *i* in G (i.e., x_i = 1) and $i \notin S$, by the definition of deduction graphs there is a directed path in G connecting i from some node s_0 of S. Thus, $B(i) \neq \varnothing$, and (iii)(a), (iii)(b) hold. If $i \in S$, (iii)(b) holds obviously. For any node *z* not in *G* (i.e., $x_i = 0$), we have

$$y(j,i) = 0 \quad , \quad \text{for all } j \in B(i)$$

$$y(i,j) = 0 \quad , \quad \text{for all } j \in A(i)$$

which means that (iii)(a) and (iii)(b) are satisfied. This completes the necessity proof.

Sufficiency. Any subgraph *G* satisfying (ii) is legitimate (see definition 2.2). From (i), we know that each node of *T* is in G. It remains to show that, for each node $i \notin S$ in *G* (i.e., x_i = 1), there is a directed path connecting i from some node of *S*. To see this, as $i \notin S$, $B(i) \neq \varnothing$; see remark 2.1. From (iii)(a), there must be node $j_1 \in B(i)$ so that $y(j_1, i) = 1$. By (iii) (b) $x_{j_1} = 1$ Thus, arc $r(j_1, i)$ is in G. If $j_1 \in S$, we finish the proof, because $j_1 \to i$ is the needed path; otherwise, we repeat the process: by (iii)(a), there is $j_2 \in B(j_1)$ such that $y(j_2, j_1) = 1$; and by (iii)(b), $x_{j_2} = 1$. Thus, arc $r(j_2, j_i)$ is in *G*. *If* $j_2 \in S$, we finish; otherwise, again by (iii)(a) and (iii)(b), there is $j_3 \in B(j_2)$ such that arc $r(j_3, j_2)$ is in *G*. This process continues until we find a directed path $j_r \to j_{r-1} \to \ldots\ldots \to j_2 \to j_1 \to i$ with $j_r \in S$. Note that, as $G_\infty(S, T)$ is finite and contains no cycle, the above process should terminate in finite steps.

Lemma 3.2. *For subgraph G condition (ii) of Lemma 3.1 is satisfied if and only if, for each compound node i in $G_\infty(S, T)$, the following holds:*

$$\left[B(i)\right] x_i \le \sum_{j \in B(i)} x_{i_t} \ .$$

Recall that *c(i, j)* is the cost or time needed to acquire *j* from *i*. Given a deduction subgraph *DG(S, T)* of $G_\infty(S, T)$, to carry out the plan of *DG*(*S, T*) will cost a total of $\sum_{r(i,j) \in \mathcal{A}} c(i,j).\, y(i,j)$, where *A* is the set of all arcs $DG_\infty(S, T)$. This observation and lemmas 3.1-3.2 provide the following theorem.

Theorem 3.1. *Given $G_\infty(S, T)$, the subgraph obtained by solving the following program is the minimum cost deduction graph:*

$$\min \quad Z = \sum_{r(i,j) \in \mathcal{A}} c(i,j).\, y(i,j)$$

$$s.t. \quad (i) \qquad x_i = 1, \text{ for each } i \in T;$$

(ii) $\left[B(i)\right]x_i \le \sum_{j\in B(i)} x_{i_t}$, *for each compound node i;*

(iii) $x_i \le \sum_{j\in B(i)} y(j,i)$ *for each* $i \notin S$;

(iv) $\left(\left[A(i)\right]+\left[B(i)\right]\right)x_i \ge \sum_{j\in B(i)} y(j,i)+\sum_{j\in A(i)} y(i,j)$ *for each* $i \in N$;

(v) *all* x_i *and y(i,j) are 0 or 1.*

Proof. From lemmas 3.1-3.2, we know that a subgraph is a deduction graph from *S* to *T* iff the corresponding variables satisfy the constraints (i)-(v). The conclusion is then obvious.

Example 3.1. Now, we apply theorem 3.1 to solve example 2.1, which is equivalent to solving the following 0-1 linear integer problem. Referring to *figure 1*, we have the following program:

min OBJ = $y(a, d) + 3y(a,f) + 4y(a, g) + 2y(b, d) + 4y(b, g) + 2y(c, d) + 3y(c, e)$
$+ y(a \wedge b, d) + 2.5y(a \wedge b, f) + 3.5y(a \wedge b, g) + y(d, e) + 1.8y(d, f)$
$+4y(e, h) + 2y(f, g) + 2y(f, h) + 1.5y(e \wedge f, g) + 1.5(e \wedge f, h).$

s.t. (i) $x_d = x_f = x_g = x_h = 1;$

(ii) $2x_{a\wedge b} \le x_a + x_b,\ 2x_{e\wedge f} \le x_e + x_f;$

(iii) $x_d \le y(a, d) + y(b, d) + y(c, d) + y(a \wedge b, d),$
$x_e \le y(c, e) + y(d, e),$
$x_f \le y(a, f) + y(a \wedge b, f) + y(d, f),$
$x_g \le y(a, g) + y(b, g) + y(a \wedge b, g) + y(f, g) + y(e \wedge f, g),$
$x_h \le y(e, h) + y(f, h) + y(e \wedge f, h);$

(iv) $3x_a \ge y(a, d) + y(a, f) + y(a, g),$
$2x_b \ge y(b, d) + y(b, g),$
$2x_c \ge y(c, d) + y(c, e),$
$3x_{a\wedge b} \ge y(a \wedge b, d) + y(a \wedge b, f) + y(a \wedge b, g)$
$(4 + 2)x_d \ge [y(a, d) + y(b, d) + y(c, d) + y(a \wedge b, d)] + [y(d, e) + y(d, f)],$
$(2 + 1)x_e \ge [y(c, e) + y(d, e)] + y(e, h),$
$(3 + 2)x_f \ge [y(a, f) + y(a \wedge b, f) + y(d, f)] + [y(f, g) + y(f, h)],$
$5x_g \ge y(a, g) + y(b, g) + y(a \wedge b, g) + y(f, g) + y(e \wedge f, g),$
$(3 + 0)x_h \ge [y(e, h) + y(f, h) + y(e \wedge f, h)],$
$2x_{e\wedge f} \ge y(e \wedge f, g) + y(e \wedge f, h);$

(v) all $y(i, j)$ and x_i $(i = a, b, c, d, a \wedge b, e \wedge g)$ are 0-1 integers.

Using a 0-1 integer programming package[6], we obtain the solution as
$y(a \wedge b, d) = y(d, e) = y(d, f) = y(e \wedge f, g) = y(e \wedge f, h) = 1.$

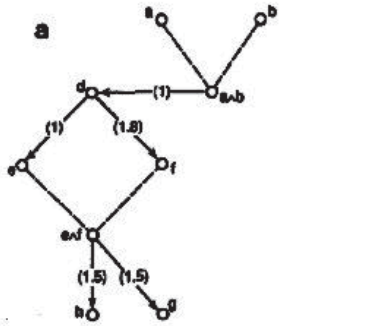

Figure : 3a Graph DG(5, *T)*

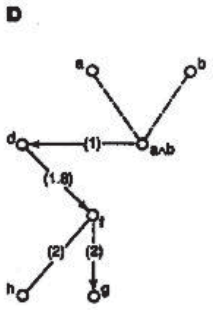

Figure : 3b Another graph DG(5, *T)*

The resulting competence graph represented as *DG*(*S(E), T(E)*) is shown in *figure 3a*. The total cost is 1 + 1 + 1.8 + 1.5 + 1.5 = 6.8. Note that there is an alternative optimal solution, for instance, $y(a \wedge b, d) = y(d, f) = y(f, g) = y(f, h) = 1$, all other $y(i, j) = 0$, where the total cost is also 6.8 (*Figure 3b*).

4. Conclusion

We have described a new method for effectively helping the decision maker to expand his competence sets based on the concept of deduction graphs. An example has been given for better explanation of the newly described method. Many research problems are open. For instance, in a multilevel case, how do we effectively solve the corresponding multiple criteria problems? Is there any optimal stopping rule so that we could solve the problem sequentially (instead of solving the problem in one step)? If the need for the skills in the competence set is a random process, how do we formulate the problem of expanding and/or acquiring new skills?

References

[1]. L. SCHRAGE, Linear, Integer, and Quadratic Programming with LINDO, Scientific Press, Palo Alto, California, 1984, 212-234.

[2]. R.T. VAN and L. WOLSERY, Solving Mixed 0-1 Programs by Automatic Reformation, Operations Research, 35(1987), 45-57.

[3]. P.L Yu and D. Zhang, A Foundation for Competence Set Analysis, Mathematical Social Sciences, 20(1990), 251-299.

[4]. P.L. Yu and D. Zhang, Optimal Expansion of Competence Sets and Decision Support, Information Systems and Operational Research, 30(1992), 68-84.

[5]. C.C. YANG and H.L. Li, Abductive Reasoning by Constructing Probabilistic Deduction Graphs for Solving the Diagnosis Problem, D Systems, 7(1991), 121-131.

[6]. P.L. Yu, Multiple-Criteria Decision Making: Concepts, Techniques, and Extensions, Plenum press, New York, 1985, 125-201.

Received: June, 2011