Scheduling Meetings in Distance Learning*

Jian Wang, Changyong Niu, and Ruimin Shen

Department of Computer Science and Engineering
Shanghai Jiaotong University, Shanghai 200030, China
{jwang,cyniu,rmshen}@sjtu.edu.cn

Abstract. Peer-to-peer technique becomes mature gradually, and multiple domain-involved applications emerge, such as IPTV, distance learning, chatting network. In the context of distance learning, small scale of teachers would serve large scale of geographically located students. Normally, knowledge points are associated with different difficulty levels. And students usually are interested in varied subsets of knowledge points. Also teachers are capable of serving knowledge point subsets. The objective to schedule meeting among students and teachers according to their respective interests and capabilities is to reduce total learning duration. After formulating meeting schedule as Integer Programming problem, this paper proposes three heuristic algorithms to approximate the optimal solution. To the best of our knowledge, such problem is firstly investigated in distance learning context. Performance evaluation demonstrates their behaviors and *PKPA* algorithm excels two others substantially.

Keywords: Scheduling, Timetable, Integer Programming, Heuristics, Distance Learning.

1 Introduction

Scheduling is of common combinatorial optimization and planning problem, and finds usage in almost every resource-constrained scenarios. The timetable scheduling, one of important scheduling problems, exists in realistic life, such as lecture, transportation, examination, meeting. Due to heavy problem complexity and involved domain context, such problem is widely solved with heuristics.

Peer-to-Peer technique has been thoroughly studied in the past decade. And multiple domain-involved applications emerge such as IPTV, distance learning, chatting network, content-based service. The Peer-to-Peer overlay distributes learning content, consisting of audio, video, and handwriting, to large population of consumers efficiently. Consequently, distance learning achieves big progress in terms of scale of service capability. Among involved students and teachers in distance learning, there exists scheduling issue for efficient resource usage.

Knowledge points in distance learning are normally associated with difficulty levels. Students request different subset of them due to interest or difficulty

* This work is supported by the NFSC under Grant No.60672066, China.

M. Xu et al. (Eds.): APPT 2007, LNCS 4847, pp. 580–589, 2007.

level. In addition, the teachers do also specialize in subset of knowledge points whose size is often large. The objective of scheduling meetings among involved students and teachers according to their respective interests and capabilities is to reduce total learning duration. Of course, such scheduling strategy can apply into virtual lab or virtual collaboration on the Internet, too.

To the best of our knowledge, scheduling meetings in context of distance learning is firstly investigated. The remainder of this paper is organized as follows. Section 2 reviews representative works on timetable scheduling. Section 3 formally formulates scheduling meeting problem and three heuristic algorithms are proposed consequently in section 4. The performance of algorithms are numerically evaluated in section 5. The paper concludes in section 6 with future work.

2 Related Work

As scheduling is concerned, the most relevant works are timetable ones. Tabu [1] deals with problem of assigning teachers to courses in a secondary school. Such timetable is to be built when teaching assignments are not fixed. It considers the characteristics of the school week, finite teachers and rooms, individual subject requirement, prerequisites, as well as characteristics and regulations of country-specific education system into account and finds a schedule for a set of meetings between teachers and groups of students over a set of time periods using tabu searching algorithm. Timetable is so difficult to solve due to large search space and highly constrained requirements. Thus, works [2] and [3] strive to approximate the optimal solution for timetable problem by genetic algorithms.

TGA [4] presents two-phase genetic algorithm to solve timetable problem for universities, and it uses two populations for class scheduling and room allocation, respectively. Consequently, it achieves better performance than simple genetic algorithm. Work [5] utilizes Particle Swarm Optimization technique to solve the discrete problem of timetable scheduling. And PSO performs well in discrete problem. Furthermore, the timetable problem is solved with efficient heuristic numerical algorithm in [6]. The main objective of timetable is to find feasible time slots with respect to multiple constraints.

TCDMP [7] is a Timetable-Constrained Distance Minimization Problem, which is a sports scheduling problem applied for tournaments and the total travel distance on individual teams must be minimized. MICSP [8] tends to address curriculum planning problem, which is defined as constructing a set of courses for each semester - over a sequence of semesters - in order to satisfy the academic requirements such as for undergraduate university degree. Obviously, both have different objectives to be optimized compared to timetable works. Interestingly, scheduling meetings in distance learning also differs from existing works due to intrinsic objective. As students interested in same knowledge point can be served by one teacher, where such sharing is easy in peer-to-peer overlay, the objective in this paper is to minimize total learning duration with relatively few constraints. The total learning duration denotes the time period from the instant the learning begins to the time all students complete learning.

3 Problem Formulation

In distance learning, activities are often conducted in *group* form. The group is formed by students with common interest, as well as supervised teachers. Given a large set of knowledge points, students involved with common knowledge point form a group. In collaborated virtual experiment scenario, specific experiment is correlated with student subsets as well as teachers. Thus, group is common form in distance education, especially when number of overall students is large. Due to finite number of teachers, it is imperative to schedule learning groups in a sequence time-efficiently.

3.1 Assumptions and Constraints

For formulating scheduling meeting problem easily, the notations and assumptions are introduced as following.

- ◇ P: set of knowledge points, and $P = \{p_1, p_2, \cdots, p_l\}$
- ◇ T: set of teachers, and $T = \{t_1, t_2, \cdots, t_m\}$
- ◇ S: set of students, and $S = \{s_1, s_2, \cdots, s_n\}$
- ◇ X_{t_i}: subset of knowledge points that can be served by teacher t_i, and $X_{t_i} \subseteq P, 1 \leq i \leq m$
- ◇ Y_{s_j}: subset of knowledge points that is requested by student s_j, and $Y_{s_j} \subseteq P, 1 \leq j \leq n$
- ◇ Each knowledge point $p_k, 1 \leq k \leq l$ incurs same unit learning time

Assuming student in group interacts with supervised teacher during learning session, it is useless to prerecord lectures of knowledge point set such that student requests content of his favorites on demand, consequently. Thus, the constraints of scheduling meetings are given.

- ◇ Each teacher t_i can serve at most one knowledge point at any time.
- ◇ Each student s_j can enjoy at most one knowledge point at any time.

The problem to investigate is: Given knowledge point set P, teacher set T and their corresponding serving capabilities $\{X_{t_i}, 1 \leq i \leq m\}$, student set S and their corresponding knowledge requests $\{Y_{s_j}, 1 \leq j \leq n\}$, how does the scheduling strategy arrange students and teachers to study knowledge point together (i.e. meeting) in minimum total learning time.

3.2 Formulation

A decision variable z_{ijk}^r is introduced to denote whether student s_j can enjoy knowledge point p_k served by teacher t_i at time round r

$$z_{ijk}^r = \begin{cases} 1 & \text{student } s_j \text{ enjoys } p_k \text{ served by teacher } t_i \text{ at time round } r \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

Consequently, the main objective is to minimize number of rounds for scheduling all involved teachers and students to complete all learning requests.

$$\text{minimize } R \tag{2}$$

Subject to

$$\sum_{j=1}^{n}\sum_{k=1}^{l} z_{ijk}^{r} \leq 1, \forall i \in \{1, 2, \cdots, m\}, \forall r \in \{1, 2, \cdots, R\} \tag{3}$$

$$\sum_{i=1}^{m}\sum_{k=1}^{l} z_{ijk}^{r} \leq 1, \forall j \in \{1, 2, \cdots, n\}, \forall r \in \{1, 2, \cdots, R\} \tag{4}$$

$$\bigcup_{i=1}^{m}\bigcup_{k=1}^{l}\bigcup_{r=1}^{R}\{p_k | z_{ijk}^{r} = 1\} = Y_{s_j}, \forall j \in \{1, 2, \cdots, n\} \tag{5}$$

$$\bigcup_{j=1}^{n}\bigcup_{k=1}^{l}\bigcup_{r=1}^{R}\{p_k | z_{ijk}^{r} = 1\} \subseteq X_{t_i}, \forall i \in \{1, 2, \cdots, m\} \tag{6}$$

$$|\bigcup_{i=1}^{m}\bigcup_{j=1}^{n}\bigcup_{r=1}^{R}\{r | z_{ijk}^{r} = 1\}| \leq \tau, \forall k \in \{1, 2, \cdots, l\} \tag{7}$$

$$\begin{aligned} z_{ijk}^{r} \in \{0, 1\}, \forall r \in \{1, 2, \cdots, R\}, \forall i \in \{1, 2, \cdots, m\}, \\ \forall j \in \{1, 2, \cdots, n\}, \forall k \in \{1, 2, \cdots, l\} \end{aligned} \tag{8}$$

The formulation is a comprehensive integer linear programming. Equation (3) reflects the constraint that each teacher can serve at most one knowledge point at any time, and that of student in Equation (4). Equation (5) declares that individual learning request set Y_{s_j} must be satisfied. In addition, Equation (6) shows that teacher serves knowledge set no more than his capability X_{t_i}. Equation (7) enforces cost efficiency as the one knowledge point can at most be taught τ times. At last, Equation (8) indicates that the formulation is an integer programming. Since such problem is intrinsically complex, this paper resorts to heuristic algorithms to approximate the optimal solution.

4 Heuristic Algorithms

A simple example is given for above formulation before proposing three heuristic algorithms. In Figure 1, there are four students $\{s_1, s_2, s_3, s_4\}$, three knowledge points $\{p_1, p_2, p_3\}$, as well as three teachers $\{t_1, t_2, t_3\}$. Student requests and teacher capabilities are reflected by those edges in graph. For instance, student s_3 requests knowledge point set $\{p_1, p_2\}$, while teacher t_2 could serve knowledge point set $\{p_2, p_3\}$.

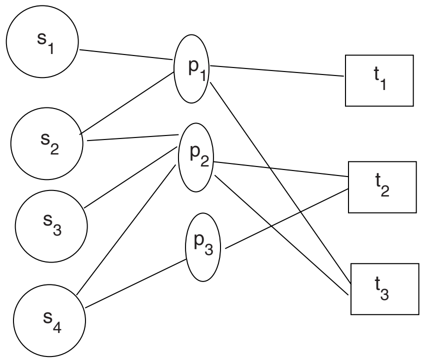

Fig. 1. An example relation graph for knowledge point learning

As scheduling meeting is concerned, there requires 2 rounds to complete knowledge learning process. In round 1, teacher t_2 serves the knowledge point p_3 that is interested by students s_4, while teacher t_1 serves p_1 that is enjoyed by s_1, s_3. Then in round 2, teacher t_2 serves p_2 that is interested by s_2, s_3, s_4. Obviously, there may exist multiple scheduling arrangements corresponding to same number of rounds. Such phenomenon contributes finding scheduling arrangement of minimum time quickly.

As size of students and knowledge points becomes larger, it is impracticable to find the optimal solution for the problem due to large computation load. Thus, three heuristic algorithms are proposed to approximate optimality quickly. They are feeded with same input and derive round number R individually. For clarity, the input is described as: knowledge point set P, student set S and corresponding request $\{Y_{s_j}\}$, teacher set T and corresponding capabilities $\{X_{t_i}\}$. For convenience, the edge between student s_j and his knowledge request $\{Y_{s_j}\}$ is uniquely labeled by each element in $\{Y_{s_j}\}$. Similarly, such labeling strategy applies to edges between teacher t_i and capability $\{X_{t_i}\}$.

In brief, three following algorithms are greedy-based. The τ-constraint defined in Equation (7) is relaxed in algorithms. Each iteration attempts to maximize number of students that can be served, although PSA, PTA, PKPA start iteration at position of student, teacher and knowledge point, respectively. In addition, each algorithm contains two layers of loop. The outer loop guarantees each individual student to be satisfied. And inner loop attempts to maximize served students in one round.

4.1 Preferential-Student Algorithm (PSA)

The idea is that in each iteration, first find student s_j with maximum knowledge points pending for studying. Then among those points, choose the one p_k that can be served by maximum available teachers. Finally, select the teacher t_i with minimum capability among corresponding teachers.

```
1: E = ∪_{j=1}^{n} Y_{s_j}, round = 0
2: WHILE E ≠ ∅
3:    D = {s|s ∈ S, Y_s = ∅}, S = S − D
4:    A = ∅, B = ∅, C = ∅, W = ∅
5:    WHILE S − A − B ≠ ∅
6:       s = argmax_{x∈S−A−B} |Y_x|
7:       p = argmax_{x∈Y_s−W} |Ω(x)|
8:       IF Ω(p) == ∅
9:          B = B ∪ {s}
10:         CONTINUE
11:      END
12:      t = argmin_{z∈Ω(p)} |X_z|
13:      C = C ∪ {t}, V = {s|s ∈ S − A − B, p ∈ X_s}
14:      A = A ∪ V, W = W ∪ {p}
15:      Y_z = Y_z − {p}, ∀z ∈ V
16:   END
17:   E = ∪_{j=1}^{n} Y_{s_j}
18:   round = round + 1
19: END
20: RETURN round
```

Fig. 2. Preferential-Student Algorithm

Definition 1. *Let $\Omega(p) = \{t|p \in X_t, t \in T, t \text{ is available}\}$ denote the teacher subset, where each is available and is capable to serve knowledge point p.*

In Figure 2, E represents collection of pending student requests and S denotes students that still have requests. The S will be divided into two set A and B. A contains the students that can learn in current round, while B contains those not. Step 6 selects the student s with largest size of request set. Step 7 chooses the knowledge point p of s that has maximum available teachers. If such p does not exist, the student s joins into set B. Otherwise, the teacher of minimum capability t corresponding to p is chosen. Consequently, the unavailable teacher set C, knowledge point served W, the learning student set A as well as knowledge request of necessary students Y_z are updated in sequence. Whenever $S - A - B == \emptyset$ (i.e. the current students are divided into groups), the inner loop stops. Thus, E is updated and round is increased by one before next loop.

4.2 Preferential-Teacher Algorithm (PTA)

This algorithm takes inverse design direction compared to PSA. The idea is that in each iteration, first find maximum capability teacher t_i, then select knowledge point $p_k \in X_{t_i}$ with maximum interested students.

In Figure 3, E represents the pending knowledge requests and S denotes those students that have knowledge points to learn. The teacher set will also be divided into two groups B and C. B contains those teachers not interested by current student S. In addition, A denotes allocated students while W denotes

1: $E = \cup_{j=1}^{n} Y_{s_j}, round = 0$
2: WHILE $E \neq \emptyset$
3: $D = \{s|s \in S, Y_s = \emptyset\}, S = S - D$
4: $A = \emptyset, B = \emptyset, C = \emptyset, W = \emptyset$
5: WHILE $T - B - C \neq \emptyset$
6: $t = argmax_{z \in T-B-C} |X_z|$
7: $p = argmax_{z \in X_t - W} |\{s|s \in S - A, z \in Y_s\}|$
8: IF $p == \emptyset$
9: $B = B \cup \{t\}$
10: ELSE
11: $V = \{s|s \in S - A, p \in Y_s\}$
12: $A = A \cup V, C = C \cup \{t\}, W = W \cup \{p\}$
13: $Y_z = Y_z - \{p\}, \forall z \in V$
14: END
15: END
16: $E = \cup_{j=1}^{n} Y_{s_j}$
17: $round = round + 1$
18: END
19: RETURN $round$

Fig. 3. Preferential-Teacher Algorithm

allocated knowledge points. Step 6 selects maximum capable teacher t. Step 7 selects knowledge point p of t interested by maximum students in $S - A$. If such p is not found, the teacher t joins into B. Otherwise, allocated students A, allocated teachers C, allocated knowledge points W, and knowledge point request Y_z are updated consequently. The inner loop stops when all teachers are grouped. Of course, E and $round$ updated in order for next round of the inner loop.

4.3 Preferential-Knowledge Point Algorithm (PKPA)

The idea of third algorithm is in each iteration, first find knowledge point p_k with maximum unallocated interested students. Then select available teacher t_i that can serve p_k and has minimum corresponding capability.

In Figure 4, pending knowledge point set E and pending students S are initialized. The allocated students A, allocated teachers C, and allocated knowledge points W are set to empty. Step 6 selects the knowledge point p interested by maximum unallocated students. If p is not interested, that means no more knowledge points can be allocated. Otherwise, minimum capable teacher for knowledge point p is selected. If no available teacher exists, p joins into W. Otherwise, A, C, W, Y_z are modified accordingly. The inner loop stops when all knowledge points are inspected or no unallocated student exists. Within outer loop, $E, round$ are updated conveniently.

```
E = ∪_{j=1}^{n} Y_{s_j}, round = 0
WHILE E ≠ ∅
    D = {s|s ∈ S, Y_s = ∅}, S = S − D
    A = ∅, C = ∅, W = ∅
    WHILE P − W ≠ ∅
        p = argmax_{z∈P−W} |{s|s ∈ S − A, z ∈ Y_s}|
        V = {s|s ∈ S − A, p ∈ Y_s}
        IF V == ∅
            BREAK
        END
        t = argmin_{z∈T−C, p∈X_z} |X_z|
        IF t == ∅
            W = W ∪ {p}
        ELSE
            A = A ∪ V, C = C ∪ {t}, W = W ∪ {p}
            Y_z = Y_z − {p}, ∀z ∈ V
        END
    END
    E = ∪_{j=1}^{n} Y_{s_j}
    round = round + 1
END
RETURN round
```

Fig. 4. Preferential-Knowledge Point Algorithm

5 Performance Evaluation

We evaluate three heuristic algorithms running on a computer with 1.5 GHz CPU and 512 MB memory. The algorithms are implemented upon the *Matlab*. Scheduling meeting problem has six configuring parameters: l:number of knowledge point, m: number of teachers, n: number of students, f: size of $X_{t_i}, 1 \leq i \leq m$, g: size of $Y_{s_j}, 1 \leq j \leq n$, τ: cost tradeoff parameter. We relax τ in three proposed algorithms. In addition, each result is averaged over 10 runs through random sampling. We vary one configuring parameter while fixing others in order to reveal individual impact on algorithm performance. Each X_{t_i} and Y_{s_j} are randomly sampled on knowledge point set P. In addition, $\left(\bigcup_{j=1}^{n} Y_{s_j}\right) \subseteq (\bigcup_{i=1}^{m} X_{t_i})$ is ensured such that feasible solution exists.

l:number of knowledge point. In Figure 5, $m = 6, n = 40, f = 6, g = 5$. As number of knowledge point increases, the groups become less overlapping. Three algorithms need more rounds to complete the arrangement. Note that PKPA performs better than two others.

m:number of teachers. In Figure 6, $l = 20, n = 40, f = 6, g = 5$. Obviously, as the number of teachers grows up, more probably concurrent groups appear. Of course, the rounds derived by three algorithm decrease, although PKPA does the best.

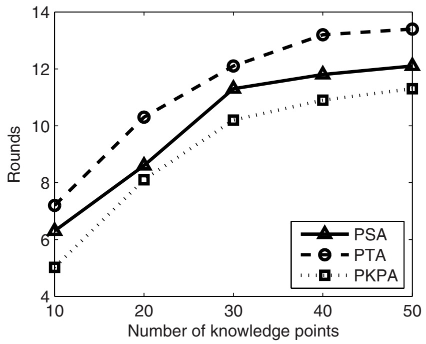

Fig. 5. Rounds vs. number of knowledge point

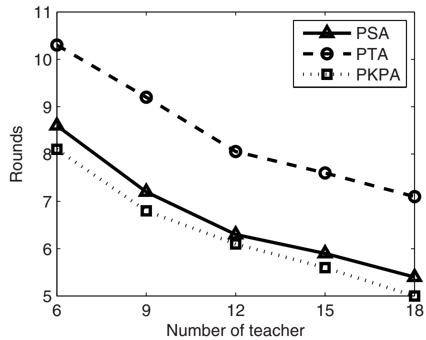

Fig. 6. Rounds vs. number of teacher

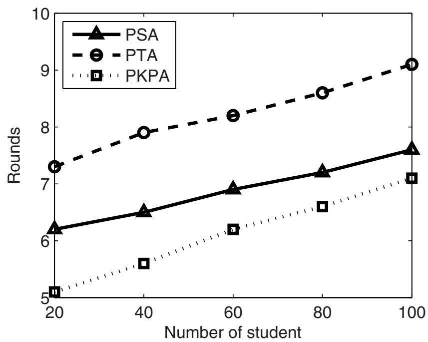

Fig. 7. Rounds vs. number of student

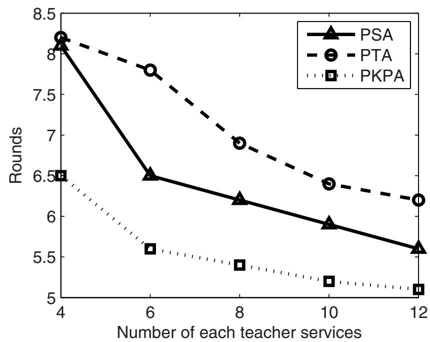

Fig. 8. Rounds vs. number of each teacher services

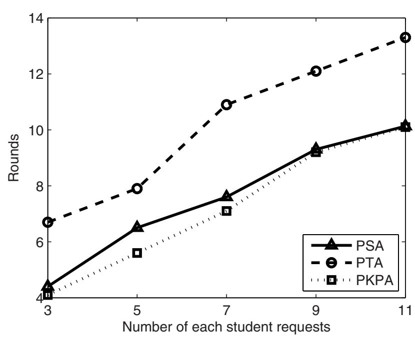

Fig. 9. Rounds vs. number of each student requests

n:number of students. In Figure 7, $l = 20, m = 11, f = 6, g = 5$. Increasing number of students means more requests need to be handled by teacher set. So the rounds increase definitely.

f:number of each teacher services. In Figure 8, $l = 20, m = 11, n = 40, g = 5$. Normally, teacher with larger service capability would decrease corresponding rounds since more probably the request can be satisfied.

g:number of each student requests. Figure 9 is similar to Figure 7. Here $l = 20, m = 11, n = 40, f = 12$. The number of the whole requests increases as g becomes large. Thus, rounds increases for three algorithms.

Summary: In five figures corresponding to each configuring parameter except for τ, the PKPA algorithm always outperforms two others. Potential reasons are: 1) selecting maximum student in each iteration. 2) choosing teacher with minimum capability would let other teacher can still contribute in following selections within same iteration.

6 Conclusion

This paper formulates scheduling meetings problem in distance learning scenario. Then three heuristic algorithms are proposed to quickly approximate the optimal solution. And the evaluation reveals that PKPA performs the best. In future work, one is to derive mathematical analysis of such formulation further, and the other is to incorporate τ into heuristic algorithm design.

References

1. Alvarez-Valdes, R., Parreno, F., Tamarit, J.M.: A Tabu Search Algorithm for Assigning Teachers to Courses. Top 10(2), 239–259 (2002)
2. Burke, E., Newall, J-P.: Multi-Stage Evolutionary algorithm for timetable problem. IEEE Transaction of Evolutionary Computer 3(1), 63–74 (1999)
3. Li, J.-n.: An Evolutionary algorithm for Solving Curriculum Schedule Problem of University. Microcomputer Development 13(10), 96–98 (2003)
4. Ueda, H., Ouchi, D., Takahashi, K., Miyahara, T.: A Co-evolving Timeslot/Room Assignment Genetic Algorithm Technique for University Timetabling. In: Proceedings of the 3rd International conference on Practice and Theory of Automated Timetabling, Konstanz, Germany, August,16-18, 2000, pp. 48–63 (2000)
5. Chu, S.-C., Chen, Y.-T., Ho, J.-H.: Timetable Scheduling Using Particle Swarm Optimization. In: Proceedings of the First International Conference on Innovative Computing, Information and Control, Beijing, China, August 30 - September 01, 2006, pp. 324–327 (2006)
6. Xie, F.-r.: A heuristic Numerical Algorithm for Solving the Problem of Curriculum Scheduling. Operations Research and Management Science 14(5), 36–40 (2005)
7. Rasmussen, R.V., Trick, M.A.: The Timetable Constrained Distance Minimization Problem. In: Proceedings of third International conference on CPAIOR, Cork, Ireland, May 31-June 2, 2006, pp. 167–181 (2006)
8. Wu, K., Havens, W.S.: Modeling an Academic Curriculum Plan as a Mixed-Initiative Constraint Satisfaction Problem. In: Proceedings of 18th Canadian Conference on Artificial Intelligence, Victoria, Canada, May 9-11, 2005, pp. 79–90 (2005)