II International Conference on Industrial Engineering and Industrial Management
XII Congreso de Ingeniería de Organización
September 3-5, 2008, Burgos, Spain

Formalisation of the assignment of a set of tasks when work performance depends on the time devoted previously to the tasks

Albert Corominas Subias[1], Jordi Olivella Nadal[1] , Rafael Pastor Moreno[1]

[1] IOC Research Institute and Management Department, Universitat Politècnica de Catalunya

Keywords: Task assignment, Learning curves

1. Introduction

The influence of experience on performance is present in any policy of scheduling of work, but usually the experience obtained during the scheduling horizon is not taken into account. In effect, a necessary element for task scheduling is a relation of the tasks that a worker is able to do, and this is, at least in part, a consequence of experience. Moreover, the training plans can include forced assignment of tasks to obtain future capacities. A static point of view neglects the effect of the experience obtained during the planned horizon on the performance of the same horizon period. This effect is sometimes really negligible – when we are planning a period too short to generate signifying experience on the tasks involved – but often it is not. Thus, when scheduling a set of tasks the experience acquired in the first stages can influence the capacity of the worker to do other tasks of the same set later. In spite of this, the consideration of the influence on the performance in a task of the experience on other tasks has not been treated in the literature, probably due to the hard computational problem that this generates.

The relationship between experience and performance has been widely studied and this has led to the emergence of the concept of the learning curve, an equation that shows the relationship between experience and performance. It is based on the premise that the performance of a task by an organization or person improves with experience. The concept has been extensively applied to organizational processes, and specific aspects of it have also been analysed. Nembhard and Uzumeri (2000) test a total of 11 alternative learning curve models that predict individual performance of a task according to the number of previous repetitions of the task. The predictive capacity of the models is proved and a three-parameter hyperbolic model is found to give the best approximation.

In addition to experience in the task to be done, other elements reflecting the current capacity of an individual have been considered. The forgetting factor has been included in the learning curve by Shafer *et al.* (2001). An indicator of individual potential or general cognitive ability is used by Fowleret *et al.* (2007). The influence of the experience acquired in one task on the performance of another was presented by Olivella (2007).

The assignment of a set of tasks has been dealt with in the literature (Bailey *et al.*, 1995; Alfares and Bailey, 1997; Pastor and Corominas, 2007), but few works have addressed aspects of learning. Hanne and Nickel (2005) consider a learning curve for determining the personal skills of programmers who are involved in a software project, but only when they use simulation to analyse the problem and not when they are searching for an optimal solution. In Fowler *et al.* (2007) cross-training activities must be performed in a specific way that depends on the worker's previous knowledge and the task that he or she is learning. As far as the authors are aware, the assignment of a set of tasks taking into account the influence of experience acquired in one task on other tasks has not been discussed in the literature. Here we develop the case where the experience is measured as the time devoted previously to the tasks. In the next section the model proposed is described, while in Section 3 we expose the a set of alternative formulations and a numerical experience to compare them and Section 4 is dedicated to the conclusions and the future work.

2. Characteristics of the model

The problem to solve is task assignment when work performance depends on experience of the task and on experience of other tasks, with experience measured in periods of time devoted to a task. As performance depends on experience of a set of tasks exact information on this experience in each moment and for each worker is necessary. The model has the characteristics that follow:

a) The time is divided into periods of the same length. In each period a worker is assigned to one task or to no one; changes of assignment during a period are not possible,

b) In a period, a task can only be assigned to one worker,

c) There are precedence relations between pairs of tasks,

d) Each task cannot start before the beginning of a certain period (ready time),

e) The volume of work necessary to complete each task is expressed in units of work – we can though these units to be, for example, hours of work when working at a certain performance rate,

f) To reflect the influence of experience in a given task A on the performance of another task B, we consider a certain proportion of experience in task A as equivalent to experience in task B. The higher this proportion is, the more the performance of task B will increase due to the experience in task A. For example, one period of experience of task A could be equivalent to an experience of 0.01 periods of task B in order to forecast performance of task A, for example, if the relation between both tasks is very little, or could be equivalent to an experience of 0.8 of task B, for example, it they are very similar tasks. The equivalent experience in carrying out a task that is acquired by carrying out other tasks is bounded, which prevents workers from being considered fully experienced in a task because of their experience of another task but with no experience of the task to be performed.

g) The performance on a task –or learning curve– is function of the sum of the experience of this task and the equivalent experience obtained by doing other tasks.

Two objective functions are considered: (1) the completion time of the last task to be completed (the makespan), (2) the makespan as a primary objective and the sum of the completion times of all the tasks as a secondary one.

3. Formulation and numerical results

3.1. Formulation

Data

J	Number of tasks (j=1 ..J)
W	Number of workers (i=1 ..W)
T	Number of periods in the planning horizon (t=1 ..T)
r_j	Ready time of task j (the task cannot start before the period r_j)
v_j	Volume of work to be done in task j (number of units of work to be dedicated to task j to complete it)
e^0_{ji}	Initial experience of worker (i) of task (j), measured as periods devoted to the tasks
$q_{jj'}$	Proportion of the experience of task j' that is equivalent to the experience of task j when doing task j
b_j	Upper bound of equivalent experience of task j obtained by doing other tasks
P	Group of couples (a,b) where a is an immediate predecessor of b
m_j	Lower bound of the number of periods required to complete task j
n_j	Upper bound of the volume of work for task j done in a given period
ε	Small positive number

Variables

C_j	Completion time for task j
C_{max}	Completion time for the last task to be finished
x_{jit}	Binary variable that indicates whether task j is done by worker i in a period of time t ($i=1..I, j=1..J, t=r_j..T$)
e_{jit}	Number of periods devoted to task j by worker i before the beginning of a period of time t ($i=1..I, j=1..J, t=r_j+1..T$)
e'_{jit}	Equivalent experience of task j of worker i before the beginning of a period of time t (obtained by doing other tasks) ($i=1..I, j=1..J, t=r_j..T$)
s_{jt}	Volume of work for task j that has been completed at the end of period t ($j=1..J, t=r_j..T$)
$\delta_{j\tau_j}$	Binary variable that is 1 if task j has been completed in period τ_j and 0 otherwise ($j=1..J, t=r_j..T$)

Performance function

$\varphi_{jt}(e_{1t},..,e_{Jt})$	Number of units of work for task j that worker i is able to do in period t when the worker has been working a number $e_{1t},..,e_{Jt}$ of periods on tasks 1..J. This function reflects the fact that the performance depends on experience of the task and on experience of other tasks.

Constraints

Constraint (11) means that a worker cannot simultaneously carry out more than one task during period t, (12) means that a task cannot be simultaneously carried out by two workers or more, (13) forces C_j to be equal to or greater than the last period of activity for each task and (14) imposes that C_{max} is not less than the completion time of any task.

$$\sum_{j=1..J,t\geq r_j} x_{jit} \leq 1, \quad i=1..W,\ t=1..T \tag{11}$$

$$\sum_{i=1}^{W} x_{jit} \leq 1, \quad j=1..J,\ t=r_j..T \tag{12}$$

$$\sum_{i=1}^{W} t \cdot x_{jit} \leq C_j, \quad j=1..J,\ t=r_j..T \tag{13}$$

$$C_j \leq C_{\max}, \quad j=1..J \tag{14}$$

Constraint (15) initializes accumulative variables e_{ijt} (experience), while (16) makes e_{jit} the time devoted to task j up to period t.

$$e_{ji(r_j+1)} = e^0_{ji} + x_{jir_j}, \quad j=1..J, i=1..W \tag{15}$$

$$e_{jit} = e_{ji(t-1)} + x_{ji(t-1)}, \quad j=1..J, i=1..W, t=r_j+2..T \tag{16}$$

Constraints (17) and (18) make variables e'_{ijt} equal to the experience in a task acquired by carrying out other tasks. Equivalent experiences are obtained by linear combination of experience in other tasks (17) and are limited by the bounds of equivalent experience b (18).

$$e'_{jit} \leq \sum_{j'\in J|j'\neq j\wedge r_j<t} q_{jj'} \cdot e_{j'it}, \quad j=1..J, i=1..W, t=r_j+1..T \tag{17}$$

$$e'_{jit} \leq b_j \quad j=1..J, i=1..W, t=r_j..T \tag{18}$$

Constraint (19) makes s_{jt} the accumulated volume of work done at the end of each period. This constraint is not linear for two reasons: it includes a product between a variable and a function and this function is not necessarily linear. The linearization of the model is described later.

$$\sum_{i=1}^{W} \sum_{t=r_j}^{\tau_j} x_{jit} \cdot \varphi_j(e_{jit} + e'_{jit}) = s_{j\tau_j}, \quad j=1..J, \tau_j = r_j..T \tag{19}$$

Constraint (20) implies that the tasks are completely finished. Constraint (21) implies that δ_{jt} is equal to 1 when task j is completed at the end of period t, and constraint (22) guarantees that work is only assigned to non-completed tasks. For constraint (22) to be effective δ_{jt} must be equal to 1 when task j is completed at the end of the period t—as imposed by constraint (21)—and 0 beforehand. This last condition is implicit in the formulation: if δ_{jt} is 1, no more assignments to task j in the periods from t to T are possible; thus, if δ_{jt} was 1 when task j had not been completed in period t, task j would not be completed.

$$s_{jT} \geq v_j, \quad j=1..J \tag{20}$$

$$s_{j\tau_j} \leq v_j - \varepsilon + n_j \cdot \delta_{j\tau_j}, \quad j=1..J, \tau_j = r_j..T \tag{21}$$

$$\sum_{i=1}^{W}\sum_{t=\tau_j+1}^{T} x_{jit} \leq v_j \cdot (1-\delta_{j\tau_j}), \quad j=1..J, \tau_j = r_j + m_j..T \tag{22}$$

Constraint (24) guarantees that the precedence between tasks is respected $-a$ is an immediate predecessor of b.

$$x_{bi\tau} = 0, \quad (a,b) \in P, i=1..W, \tau = (r_b..r_{a+1}) \tag{23}$$

$$x_{bi\tau} \leq \delta_{a(\tau-1)}, \quad (a,b) \in P, i=1..W, \tau = (\max(r_a+1, r_b)..T) \tag{24}$$

Variables C_j can be eliminated and constraints (13) and (14) replaced by (25). This change is considered optional. Introducing the redundant constraints (26) and (27) can make the calculations easier and is also optional.

$$T+1-\sum_{t=r_j}^{T}\delta_{jt} \leq C_{\max}, \quad j=1..J \tag{25}$$

$$\delta_{jt} \leq \delta_{j(t+1)}, \quad j \in J, t = r_j..T-2 \tag{26}$$

$$s_{jt} \geq \delta_{j(t+1)} \cdot v_j, \quad j \in J, t = r_j..T-1 \tag{27}$$

Objective function

We consider two possible objective functions to minimize, (28) and (29), which lead to Formulations 1 and 2 respectively.

$$C_{\max} \tag{28}$$

$$C_{\max} + \frac{1}{T \cdot J} \cdot \sum_{j=1..J} C_j \tag{29}$$

Linearization of the model

To approximate the solution of the defined mathematical program, constraint (19) has to be also approximated by a linear expression. By assuming that function φ_j is concave, we consider a piecewise linear approximation. We consider the data, variables and constraints that follow:

Data

L	Number of intervals in the linear approximations of φ_j *(l=1..L$_j$)*
br_{jl}	Length of intervals in the linear approximations of φ_j *(j=1..J,l=1..L$_j$)*
α_j	Ordinate intercept of the linear approximation of φ_j corresponding to task j

β_{jl} — Slope of the l interval in the linear approximation of φ_j corresponding to task j

M_j — Supremum of the values that can take φ_j

Variables

y_{jitl} — Variables in the linear approximation of function φ_j corresponding to task j, worker i, period t and interval l

u_{jit} — Number of units of work for task j that worker i does in the period t

Constraints

$$y_{jitl} \le br_{jl}, \quad l = 1..L_j \tag{30}$$

$$y_{jitl} \ge 0, \quad l = 1..L_j \tag{31}$$

$$e_{jit} + e'_{jit} = \sum_{l=1}^{L_j} y_{jitl}, \quad j = 1..J, i = 1..W, t = r_j..T \tag{32}$$

$$u_{jit} \le \alpha_j + \sum_{l=1}^{L_j} \beta_{jl} \cdot y_{jitl}, \quad j = 1..J, i = 1..W, t = r_j..T \tag{33}$$

To linearize the constraint (19) the factor $x_{jit} \cdot \varphi_j$ has to be to be replaced by a linear approximation. To do this u_{jit} has to be the approximation of φ_j when x_{jit} is 1 and 0 otherwise. The constraint (34) imposes this condition.

$$u_{jit} \le M_j \cdot x_{jit}, \quad j = 1..J, i = 1..W, t = r_j..T \tag{34}$$

Finally, the factor $x_{jit} \cdot \varphi_j$ is replaced by u_{jit}, giving place to constraint (35), that replaces constraint (19).

$$\sum_{i=1}^{W} \sum_{t=r_j}^{\tau_j} u_{jit} = s_{j\tau_j}, \quad j = 1..J, \tau_j = r_j..T \tag{35}$$

3.2. Numerical results

The options presented give rise to a total of 16 alternative formulations, obtained by combining four options: substitution of constraints (13) and (14) by constraint (25), introduction of the redundant constraint (26), introduction of the redundant constraint (27) and what objective function is selected. The formulations with the objective function (28) provide the minimum value of *Cmax*, while the solutions with the objective function (29) minimize not only *Cmax* but also the sum of the completion times of each task.

A first test of all the formulations, with the linearization presented before, was carried out. The six formulations with the lowest computation times and at least two formulation of each objective function were selected to do the numerical experience. Table 4 shows the six corresponding linear problems. Formulations 2 and 4 include the redundant constraint (26), while 1 and 3 do not. Moreover, Formulations 3 and 4 include the redundant constraint (27), while 1 and 2 do not. The difference between Formulations 5 and 6 and Formulations 3 and 4 is only the objective function.

Linear Program	*Constraints*	*Objective function*
1	(11)-(18),(20)-(24),(30)-(35)	*Cmax*
2	(11)-(18),(20)-(24),(26),(30)-(35)	*Cmax*
3	(11)-(18),(20)-(24),(27),(30)-(35)	*Cmax*
4	(11)-(18),(20)-(24),(26),(27),(30)-(35)	*Cmax*
5	(11)-(18),(20)-(24),(27),(30)-(35)	*Cmax+ΣCj/TJ*
6	(11)-(18),(20)-(24),(26),(27),(30)-(35)	*Cmax+ΣCj/TJ*

Table 4. Linear programs to test

For each linear program it is attempted to solve a total of 36 instances of 3 tasks and 3 workers, 36 of 4 tasks and 4 workers and 36 of 5 tasks and 4 workers—the instances are detailed in Appendix 1 and the results are summarized in Table 5. With regard to linear programs 1 to 4, none of them seems to have a definitive advantage, though 4 —the one with the two redundant constraints— gives the shortest computation times while 1 —the one without redundant constraints—is the only one that solves all the instances. Linear programs 5 and 5 have a more complicated objective function and, as expected, show longer computation times.

	3 tasks, 3 workers				4 tasks, 4 workers				5 tasks, 4 workers			
Lm[1]	Min[2]	Max[3]	Ave[4]	Un[5]	Min[2]	Max[3]	Ave[4]	Un[5]	Min[2]	Max[3]	Ave[4]	Un[5]
1	0.6	21.5	2.2	0	0.8	659.8	26.2	0	2.1	8263.0	391.3	0
2	0.6	21.5	2.4	0	1.0	659.9	30.0	0	2.0	8083.1	361.7	1
3	0.7	55.2	3.3	0	1.6	423.6	26.0	0	2.7	6078.5	306.9	2
4	0.7	352.2	11.7	0	1.5	93.5	17.2	0	4.2	420.3	59.6	2
5	0.8	384.9	41.8	0	1.4	3247.6	218.9	1	4.3	9249.5	1146.5	6
6	3.8	88.6	16.0	0	1.1	7663.5	868.6	0	86.6	9017.6	2366.4	9
[1] Linear model; [2] Minimum solution time; [3] Maximum solution time; [4] Average solution time; [5]Number of instances without an optimal solution in 10200 seconds or failed (from 36 instances).												

Table 5. Summary of the solution times according to the linear program and the instance size (in sec.)

4. Conclusions and future work

The contribution of the paper is in modelling the assignment of the tasks of a project with work performance depending on experience of the tasks and of the experience of other tasks.

The influence of the experience of a task in the performance of another is supposed to be a very spread phenomena, and thus including it in the assignment of work methodologies would generate appreciable performance improvements. Further research to verify this hypothesis is proposed. Research prospects of the authors include the consideration, as constraints or as objectives, of knowledge goals, dealing with non-concave learning curves, and considering the case were performance depends on the work done, in instead of the time devoted to the tasks. Furthermore, the case where experience is measured as work done will also be developed.

References

Alfares, H. K. and Bailey, J. E. (1997). "Integrated project task and manpower scheduling." *IIE Transactions*, 29(9):711-717.

Bailey, J., Alfares, H. and Lin, W.Y. (1995). “Optimization and heuristic models to integrate project task and manpower scheduling.” *Computers & Industrial Engineering,* 29(1-4):473-476.

Fowler, J. W., Wirojanagud, P. and Gel, E. (2007). "Heuristics for workforce planning with worker differences." *European Journal of Operational Research* In Press, Corrected Proof. doi:10.1016/j.ejor.2007.06.038

Hanne, T. and Nickel, S. (2005). "A multiobjective evolutionary algorithm for scheduling and inspection planning in software development projects." *European Journal of Operational Research*, 167(3):663-678.

Nembhard, D.A. and Uzumeri, M.V. (2000). "An individual-based description of learning within an organization." *IEEE Transactions on Engineering Management*, 47(3):370-378.

Olivella, J. (2007). “An experiment on task performance forecasting based on the experience of different tasks”. *I-KNOW'07*, Graz: 305-312.

Pastor, R. and Corominas, A. (2007). “Job assignment”. In Nembhard, D. A. (ed.), *Workforce Cross Training*. Taylor & Francis Ltd., Boca Raton, Fla.

Shafer, S. M., Nembhard, D. A. and Uzumeri, M.V. (2001). "The effects of worker learning, forgetting, and heterogeneity on assembly line productivity." *Management Science*, 47(12):1639-1653.

Appendix 1. Instances

Cases with 3 tasks and 3 workers

T=20;
L=2;
b_j = [3,3,3];
br_{jl}= [[2,9],[2,9],[2,9]];
β_{jl}= [[.1,.1],[.1,.1],[.5,.1]];
r_j= [1,5,9], [1,1,9] or [1,5,5];
v_j = [5,15,15] or [10,15,10];
$q_{jj'}$= [[0, .0, .0],[.2, .0, .0],[.2, .2, .0],],
[[0, .0, .0],[.5, .0, .0],[.0, .0, .0],] or

[[0, .2, .3],[0, 0, .3],[0, 0, 0],];
Precedence: 1 is predecessor of 3 or none.

Cases with 4 tasks and 4 workers
T=20;
L=2;
b_j = [3,3,3,3];
br_{jl}= [[2,9],[2,9],[2,9],[2,9]];
β_{jl}= [[.1,.1],[.1,.1],[.5,.1],[.5,.1]];
r_j= [1,3,6,9], [1,1,6,6] or [1,3,3,9];
v_j = [5,15,15,15] or [10,10,15,15];
$q_{jj'}$= [[0, 0, 0, 0],[.2, .0, 0, 0],[.2, .2, .0, 0],[.2, .2, .0, 0]],
[[0, .0, .0, .0],[.5, .0, .0, .0],[.0, .0, .0, .5],[.0, .0, 0, 0]] or
[[0, .2, .2, .3],[0, 0, .2, .3],[.0, 0, .0, .0],[.0, 0, .0, .0]];
Precedence: 1 is predecessor of 3 or none.

Cases with 5 tasks and 4 workers
T=20;
L=2;
b_j = [3,3,3,3,3];
br_{jl}= [[2,9],[2,9],[2,9],[2,9],[2,9]];
β_{jl}= [[.1,.1],[.1,.1],[.1,.1],[.5,.1],[.5,.1]];
r_j= [1,3,5,7,9], [1,1,7,7,7] or [1,3,3,3,9];
v_j = [5,10,15,15,15] or [10,10,10,15,15];
$q_{jj'}$= [[0, 0, 0, 0, 0],[.2, .0, 0, 0, 0],[.2, .2, .0, 0, 0],[.2, .2, .0, 0, 0],[.2, .2, .0, .0, 0]],
[[0, .0, .0, .0, 0],[.5, .0, .0, .0, 0],[.0, .0, .0, .5, 0],[.0, .0, .0, 0, 0],[.0, .0, .0, 0, 0]] or
[[0, .2, .2, .2, .3,],[0, 0, .2, .2, .3,],[.0, 0, .0, .0, .0,],[.0, 0, .0, .0, .0,],[.0, 0, .0, .0, .0,]];
Precedence: 1 is predecessor of 3 and 5 or none.