A dynamic programming approach for resource allocation in oil and gas industry

M. R. Ghaeli[a*]

[a]School of Management, New York Institute of Technology, 1700 - 701 W Georgia St.,Vancouver, BC V7Y 1K8 Canada

CHRONICLE	ABSTRACT
	Maintenance engineering plays an important role in management of oil and gas projects. This paper presents a dynamic programming approach for resource allocation in oil and gas projects. The study presents a dynamic programming approach to allocate human resources for repairment of oil and gas equipment. Many heavy equipment normally needs to be repaired on predetermined scheduled and the process normally takes days or even weeks. The process can be divided into three stages of disassembling the equipment, executing the repairment and assembling the equipment. The proposed model of this paper models the problem into a classical dynamic programming and using a real-world case study, the implementation of the proposed model is described.

1. Introduction

One of the primary concerns with most project based maintenance problems is to manage human resource assignment. Assigning too many people to a particular task may increase the cost, on one hand, and decrease the productivity of the task, on the other hand. Therefore, it is important to place a trade-off between cost and time in heavy equipment repairment. Drexl (1991) is believed to be one of the pioneers to use dynamic programming for resource allocation. He explained that a recurring problem in project management could be dealt with the assignment of the resources to different jobs in the project. In several circumstances including audit scheduling, the resources are corresponded to individuals, which leads to an allocation project scheduling problem. Drexl (1991) investigated a non-preemptive type of a resource-constrained project job-assignment problem, where duration of the job and costs reply on the assigned resources and solved the resulted model using dynamic programming. To solve the problem of time-cost tradeoff in project management with available models, Robinson (1975) implemented a dynamic-programming method to find the optimal allocation for minimizing the duration of the project. One advantage of such approach is the ability to detect the best allocation among various activities with a given cost-time functions. Chan et al. (2007) used dynamic-programming to explain why companies vary in their research and development (R&D) project-management policies and proposed a dynamic approach to handle such issue.

* Corresponding author.
E-mail address: rghaeli@nyit.edu (M. R. Ghaeli)

e Silva and Costa (2013) used dynamic programming for allocating human resources in information system projects. They explained that Human resource allocation (HRA) could be considered as core processes of the project management of information systems and presented a dynamic programming approach to assign human resources to software development projects. Ruzhi et al. (2013) used dynamic programming for the management of different risks associated in software technology. Pang and Le (2014) provided weighted p-timed Petri nets and dynamic programming for total energy consumption management in flexible manufacturing systems using.

The due-dates of mould projects were hard to predict caused by different uncertain issues and there are several major random factors in mould manufacturing to be analyzed and the multi-mode concept has been considered by Wang et al. (2012). They presented a stochastic prediction model by integrating dynamic workshop load control theory where the ongoing projects was performed by integrating some priority rules and the mathematical distribution of due-date was computed. During this process, they applied dynamic programming to determine the optimal strategies which minimized penalty of project tardiness. They also proposed a project management module and implemented in the evolution of a project group where they could determine the distribution of due-date and optimal strategies. Sitek and Wikarek (2016) proposed a constraint-based method for modeling and optimizing the resource-constrained scheduling problems. They also used dynamic programming to solve the resulted problem formulation due to the complexity of the proposed model. There are many cases in the literature that researchers propose a mathematical problem but use the art of dynamic programing to solve the resulted problem (Lainati & Arena, 2010; Yeh & Kao, 2008; Qi et al., 2007).

2. The proposed model

Consider an equipment which needs to be repaired on a regular basis. The operations starts with disassembling the equipment, executing the repair and assembling the equipment. It is possible to reduce the time of accomplishing each task through assigning more resources. However, the cost of the operations will also increase. The objective of the paper is to finish the project on time T and it is not desirable to either finishing the project sooner or later. In other words, early accomplishment of the project does not create any advantage and it is not also desirable to finish the project late. The implementation of the discrete dynamic programming requires us to define seven attributes as follows,

1. Stage (n): For the proposed case study of this paper, disassembles, repairment and assembly are three stages.

2. State (i): At the beginning of each stage, we need to know how much resources are left to be assigned to each task. For this case, stage can be defined as the number of days left to be assigned to each task; namely, disassembles, repairment and assembly.

3. Action (k): Action is also defined to be the amount of resources assigned to each task. In this paper, action can be defined as the number of days allocated to each task given the number of days left.

4. Policy: Policy describes all possible choices we have for resource allocation.

5. Return ($r(n,i,k)$): The cost of assigning some resources, which are estimated upfront.

6. Transient function (j): This is an important part of dynamic programming which describes the link between two stages. In this paper, the number of days left for the next stage is equal to the number of days left at the beginning of the stage minus the number of days assigned by each action, i.e. $j=i-k$.

7. Recursive function: $f(n,i) = \min_{k \in K} \{r(n,i,k) + f(n-1,j)\}$.

2.1. Case study

The proposed study of this case study is involved with a regular repairment of some equipment. The operations is expected to finish on time and for the case study of this paper we consider $T = 10$. Table 1 demonstrates the cost of assigning a predetermined amount of time for each operation.

Table 1
The cost/time trade-off

Completion time (day)	Cost (thousand dollars)		
	Disassembly	Repair	Assembly
2	25	50	30
4	20	40	25
6	15	30	20
8	10	20	15

According to Table 1, in case we plan to disassemble the equipment in 4 days, it will cost us $20,000 and also the repairment operation cost $40,000 if we wish to finish it in four days. The implementation of the dynamic programming can be considered in two forms of forward and backward. When we solve a problem using backward approach, we first assume that all operations have been completed and start at the beginning of the last phase of the operation, which is Assembly. In this paper, we start with assembly operation. In other words, we assume that two operations of disassembly and repair have already been accomplished and we need to assign time to the last operation. Since each operation needs at least two working days, there is only 6 days left at most. Table 2 demonstrates all possible choices we may face,

Table 2
The result of the implementation of dynamic programming with $n=1$

i	k^*	$f^*(1,i)$
2	2	30
4	4	25
5	6	20

For the second stage of the operation, which is associated with repairment, we need to have between 4 to 8 days to repair the equipment. Table 3 presents details of the implementation of the dynamic programming.

Table 3
The result of the implementation of dynamic programming with $n=2$

$$f(2,i) = \min_{k \in K}\{r(2,i,k) + f(1,j)\}$$

i	$k=2$	$k=4$	$k=6$	$f^*(2,i)$	k_2^*	k_1^*
4	50+30	-	-	80	2	2
6	50+25	40+30	-	70	4	2
8	50+20	40+25	30+30	60	6	2

Finally, during the last stage, we may handle the disassembly operation where we have ten days left since it is the beginning of the operations but we can assign at most 6 days (See Table 4).

Table 3
The result of the implementation of dynamic programming with $n=3$

$$f(3,i) = \min_{k \in K}\{r(3,i,k) + f(2,j)\}$$

i	$k=2$	$k=4$	$k=6$	$f^*(3,i)$	k_3^*	k_2^*	k_1^*
10	25+60	20+70	15+80	85	2	6	2

According to the results of Table 3, we need to assign 2 days for each operation of disassembly and assembly and 6 days for the repairment.

3. Discussion and conclusion

During the past few decades, there have been significant attempts to solve complicated problems using the art of dynamic programming. Dynamic programming helps us split a complicated problem into different stages which makes it possible to escape from the curse of dimensionality. In many cases, it is not even possible to solve the original problem which is formulated as a combinatorial optimization using a commercial software package. The reason is simply because the complexity of many problems is either NP-complete or NP-hard. Therefore, as the size of the problem increases, we may expect the complexity of the proposed method could increase drastically. The proposed model of this paper has been formulated in a simple form of dynamic programming and it has been demonstrated that we could reach the optimal solution in just three stages. For the case study of this paper, we have assumed that the information of time and cost are determined precisely. However, in most practical problems it is possible to consider the numbers in an uncertain values such as fuzzy numbers and we leave it as a future studies for interested researchers.

References

Chan, T., Nickerson, J. A., & Owan, H. (2007). Strategic management of R&D pipelines with co-specialized investments and technology markets. *Management Science*, *53*(4), 667-682.

Drexl, A. (1991). Scheduling of project networks by job assignment. *Management Science*, *37*(12), 1590-1602.

e Silva, L. C., & Costa, A. P. C. S. (2013). Decision model for allocating human resources in information system projects. *International Journal of Project Management*, *31*(1), 100-108.

Lainati, A., & Arena, M. (2010). Project management models for long product rolling mills. In *Proc., AISTech Iron and Steel Technology Conf., SM Mills, ed., Association for Iron and Steel Technology, Warrendale, PA* (pp. 661-668).

Pang, C. K., & Le, C. V. (2014). Optimization of total energy consumption in flexible manufacturing systems using weighted p-timed Petri nets and dynamic programming. *IEEE Transactions on Automation Science and Engineering*, *11*(4), 1083-1096.

Qi, J. X., Wang, Q., & Guo, X. Z. (2007, August). Improved particle swarm optimization for resource leveling problem. In *2007 International Conference on Machine Learning and Cybernetics* (Vol. 2, pp. 896-901). IEEE.

Robinson, D. R. (1975). A dynamic programming solution to cost-time tradeoff for CPM. *Management Science*, *22*(2), 158-166.

Ruzhi, X., Leqiu, Q., & Xinhai, J. (2003, October). CMM-based software risk control optimization. In *Proceedings Fifth IEEE Workshop on Mobile Computing Systems and Applications*(pp. 499-503). IEEE.

Sitek, P., & Wikarek, J. (2016, September). A constraint-based approach to modeling and solving resource-constrained scheduling problems. In *International Conference on Computational Collective Intelligence* (pp. 423-433). Springer, Cham.

Wang, X. M., Chen, Q. X., Mao, N., & Liu, J. J. (2012). Mould projects due-date forecast methods under random environment. *Computer Integrated Manufacturing Systems*, *18*(2), 405-414.

Yeh, D. H., & Kao, H. H. (2008). A CPM-Based Bidirectional Approach for Type-I Assembly Line Balancing Problem. *Journal of the Chinese Institute of Industrial Engineers*, *25*(6), 481-487.