Using Sliding Frame Approach for Scheduling Large and Complex Projects

A. Sadeh¹, Y. Cohen², O. Zwikael³

¹Department of Management of Technology, Holon Institute of Technology, Holon 58102 sadeh@hit.ac.il

²The Open University of Israel, 108 Rabutzki St., P.O.Box 808, Raanana 43107

³School of Management, Marketing and International Business College of Business and Economics. The Australian National University Canberra ACT 0200 Australia

Abstract - There is a steadily increase in the size and complexity of projects. As a result, traditional project scheduling techniques, such as non-polynomial (NP) algorithms can no longer provide an optimal solution to minimum project duration under predecessors and resource constraints. This paper presents for the first time a general new approach that allows utilizing such algorithms in heuristic manner. This approach could be efficiently applied for solving most project scheduling problems. The main advantage of this approach is its ability to dissect the original problem to small controllable size sub problems for which exact techniques can be applied. Thus, it neutralizes the complexity of the applied algorithms (and their non-polynomial growth). We discuss our experience applying this approach and give some insight as to the choice of parameters of the sliding frame

Keywords - sliding frame; sliding window; project scheduling; project management; makespan

I. INTRODUCTION

Project scheduling is characterized by presence of both precedence constraints and resource constraints. This combination of constraints greatly complicates the scheduling problems and gives rise to many non-polynomial (NP) problems [1], starting from the classical Resource Constrained Project Scheduling (RCPSP) (see for example [2], [3] [4]) with one or many resources, and the many complications that have been added to it through the years: Multi mode resource availability, uncertainty and stochastic, multi mode execution modes, resources leveling, and discounted payments – to mention just a few. The aim of these techniques is to find a feasible schedule with the shortest duration.

However, when it comes to big projects with a large number of activities it is hard to find an exact solution for it [4]. The reason for this is that RCPSP is known to be Non-Polynomial (NP) hard ([2],[3],[4] [5]). This means that none of the exact optimal techniques can solve the problems faced by large projects in a timely manner. This requires a new approach, where this paper aims to suggest and investigate one.

The idea of dissecting or slicing large problems into sub-problems and solving them rather than the large one, has been very attractive in operations research [1]. Such an approach can effectively turn an exponential complexity into a polynomial complexity, e.g. an algorithm with an exponential computational time of 2^n . If

it could be sliced into equal pieces of, say, $n=10$, then solving each piece separately would give the optimal global solution. In such an instance the number of computations for solving each piece is 2^{10} (a constant) and the total complexity drops to: $(n/10)(2^{10}) \sim O(n)$. The biggest obstacle of this approach is that solving the sub-problems usually does not lead to the global optimum.

The objective of this paper is to propose a different approach for project scheduling problems, but its effect on the solution complexity is no less dramatic. The approach utilizes the structure of the precedence diagram to select a sequence of subgroups of activities called frames, and generate an optimal schedule for each frame. Since the last few activities in each frame are affected the most by the existence of the frame, the last part of each frame's schedule is disposed and recycled in the next frame. Combining the schedules of the consecutive frames generates the overall heuristic schedule. An example of the above process is detailed in section II.

The rest of the paper continues as follows: section II illustrates the process through an example of RCPSP. Section III discusses the effects of the sliding frame parameters, and section IV concludes the paper.

II. SIMPLE ILLUSTRATION OF SLIDING FRAME ON RCPSP

This section introduces the sliding frame concept through the basic Resource Constrained Project Scheduling Problem (RCPSP) with one constraining resource. The example is a 30 activity network taken from the benchmark set j30 of [3] (problem J301_1 type SM).

A. Introducing the Case for the Example

The example (depicted in Figure 1) is composed of the activity network with duration and resource consumption of each activity. Two artificial activities signify the start and finish points of the project. There are four resources renewable with the following limited availability at all times: $\text{Max}(R_1)=12$, $\text{Max}(R_2)=13$, $\text{Max}(R_3)=4$, $\text{Max}(R_4)=12$. The objective is to minimize the project duration ($C_{\max}$).

The first step in applying the sliding frame approach is to build a sequence of activities on which the frame slides. In this example, we arbitrarily chose the positional weight (PW) [6] as the sequencing method. The PW computed for each activity. (PW is the sum of processing

Fig 1. Activities on nodes network for benchmarking Problem "J301_1" (type SM) by E. Demeulemeester and W. Herroelen

times of the activity and all its follower activities.) Sequencing is done by computing and sorting the activities by decreasing order of PW [6]. Table 1 depicts the results of this process.

In this example we arbitrarily chose a frame size of 12 activities as shown in Table 1. Determining the frame size and the related effects are discussed in section III. The second frame is determined after scheduling these 12 activities, and recycling the last activities. According to the above calculation, the frame is composed of the 12 activities with the largest PW, namely: A1, A3, A2, A8, A4, A10, A13, A11, A16, A18, A9, A12. The next step is to apply an exact line-balancing algorithm for assigning these activities to stations optimally.

B Solution Procedure

Optimizing the first frame is done by complete enumeration method. Efficient enumeration procedure was applied to the first frame of the 12 activities. The results are presented in Figure 2.

The last three scheduled activities are 12, 16 and 11. These three activities are recycled in the second frame and the rest of the schedule stays permanent. Since the earliest start time of the recycled activities {12, 16, 11} is Day 13. The scheduling of the second frame starts at day 13. Table 2 depicts the second frame.

In the next iteration, the second frame is optimally solved and last three activities are recycled. Note that at each iteration nine activities are recycled. Note that at each iteration nine activities are added to the permanent solution. So after the third iteration, 27 activities are already permanently scheduled, and only five activities are left for the last (fourth) iteration in which the total project time is set to 38.

This result is identical to the optimal solution given by the benchmark database for this problem.

Investigating the step size is also part of the planned future research. Step size determines the recycled portion of the frame.

III THE EFFECTS OF FRAME SIZE AND STEP SIZE

After determining the sliding basis sequence of activities, the next step is to determine the frame size in terms of the number of activities in the frame. Increasing the frame size may contribute to the quality of the solution, but it increases the computational complexity substantially. The trade-off is clearly visible at the extremes: the largest frame includes all the activities (equivalent to the original problem with no frames) and its exact solution is the optimal, but its complexity is maximal. The smallest possible frame includes one

TABLE 1.
ACTIVITY LIST OF FIGURE 1 SORTED BY
POSITIONAL WEIGHT (PW) - THE FIRST FRAME IS
UNSHADED AND INCLUDES THE FIRST 12 ACTIVITIES
IN THE SORTED LIST.

Serial #	Activity #	Duration	PW
1	1	0	158
2	3	4	69
3	2	8	63
4	8	9	49
5	4	6	48
6	10	7	41
7	13	6	41
8	11	9	35
9	16	10	31
10	18	5	29
11	9	2	25
12	12	2	25
13	14	3	23
14	5	3	20
15	17	6	20
16	7	5	18
17	20	7	17
18	19	3	15
19	15	9	14
20	22	7	14
21	27	8	13
22	6	8	10
23	26	7	9
24	21	2	7
25	23	2	7
26	29	7	7
27	24	3	5
28	25	3	5
29	28	3	5
30	30	2	2
31	31	2	2
32	32	0	0

activity (moving on the sequence and assigning one activity at a time) which is the most simple, but its quality is the lowest. With this insight, the frame size should be increased as long as the inner frame complexity is not an impediment. It is assumed that the complexity related to frames up to twelve activities is not an impediment for

modern computers.

Another consideration for the frame size is the size of the steps when the frame slides. The step size signifies the portion of the activity assignment to stations that becomes part of the permanent solution, and determines the portion that is recycled. The following notation is used in the discussion below:

Notation

N - Total number of activities in the project

n - Number of activities in a frame

$LB = N/n$ = lower bound on the number of frames

s - Step size (in number of activities)

$r = n-s$ = recycled portion of the frame.

$NF = [N/s]^+ =$ Number of frames in practice

TABLE 2.
ACTIVITY LIST OF THE SECOND FRAME
INCLUDING THE RECYCLED ACTIVITIES {12, 16, 11}.

Serial #	Activity #	Duration	PW
1	11	9	35
2	16	10	31
3	12	2	25
4	14	3	23
5	5	3	20
6	17	6	20
7	7	5	18
8	20	7	17
9	19	3	15
10	15	9	14
11	22	7	14
12	27	8	13

The frame's utilization is (s/n) and is the percentage of the frame that is not recycled. Its complementary percentage $(1-(s/n))$ is found in the lower bound for the number of recycled activities: $(LB)(r) = (N/n)(n-s) = N(1-(s/n))$.

In practice the number of recycled activities is: $(NF)(r) = (N/s)(r) = (N/s)(n-s) = N((n/s)-1)$. This shows that when the step size is half the frame size ($s=n/2$) all N activities are recycled (except, the first $n/2$ activities, and the same number that is recycled twice.)

Substituting $s=n/3$ and $s=n/4$ we find that the

Fig 2. Gantt chart for minimal maxspan schedule for the first frame of twelve activities with resources consumption (different resource types are tinted in different shades).

corresponding total numbers of recycled activities are $N/2$ and $N/3$ respectively. This indicates the direct growth of the number of frame computations (50% and 30% more frames than LB respectively).

Using a frame size of at least ten activities facilitates recycling between third to one fourth of the frame and enables experimenting with the step size. So for this paper we use 12 activities per frame and a step size between eight and nine (dictating recycled portion of three to four activities).

IV. CONCLUSION

This paper illustrates the concept of the sliding frame and its usage for project scheduling under resource constraints. The example on section II demonstrates the great potential of this approach for project scheduling. The paper discusses variety of possible "rails" on which the frame can move. It also discusses the frame size and the step size. While preliminary results are encouraging, future research is required for comparing the performance of these parameters.

REFERENCES

- [1] R. E. Evans and E. Minieka, *Optimization Algorithms for Networks and Graphs*, Marcel Dekker Inc. New York, 1992, p. 39.
- [2] J. Y. T. Leung,, *Handbook of scheduling*, CRC press/Chapman & Hall., 2004.
- [3] R. Kolisch and A. Sprecher, "PSPLIB - A Project Scheduling Problem Library". *European Journal of Operational Research* 96, 205-216, 1996.
- [4] E.L. Demeulemeester, and W. S. Herroelen, *Project Scheduling: A Research Handbook*, , Kluwer Academic Publishers, Norwell MA, 2002, p. 203.
- [5] R. Kolisch, and S Hartman, "Heuristic algorithms for solving the resource constrained project scheduling problem: classification and computational analysis," Chapter 7 in Weglarz J. (Ed.) *Handbook of recent advances in Project scheduling*, Kluwer academic publishers, 1999. pp 147-178.
- [6] W. B. Helgeson and D. P. Birnie, "Assembly line balancing using the ranked positional weight technique", *Journal of Industrial Engineering*, 12, 394-398, 1961