Yakov Genis[1], *Igor Ushakov*[2]

OPTIMAL UNIFORM-LIKE SCHEDULING OF MAINTENANCE

The paper presents the algorithm for the optimal scheduling of work performance when the process of maintenance cannot be interrupted but there is some "possession" to perform such maintenance. The maintenance resources are limited. The optimal maintenance means that the schedule provides for as uniform-like distribution of activities of maintenance crews as possible.

1. Formulation of the scheduling problem

There are n requests for works with volumes $v_1, v_2, \ldots, v_n$ (see an example in Fig.1). Each work, k, has to be fulfilled during interval $[s_k, e_k]$, which lies between the allowed start moment, S_k, and the permissible end moments E_k, i.e.

$$[s_k, e_k] \subseteq [S_k, E_k]. \quad (1)$$

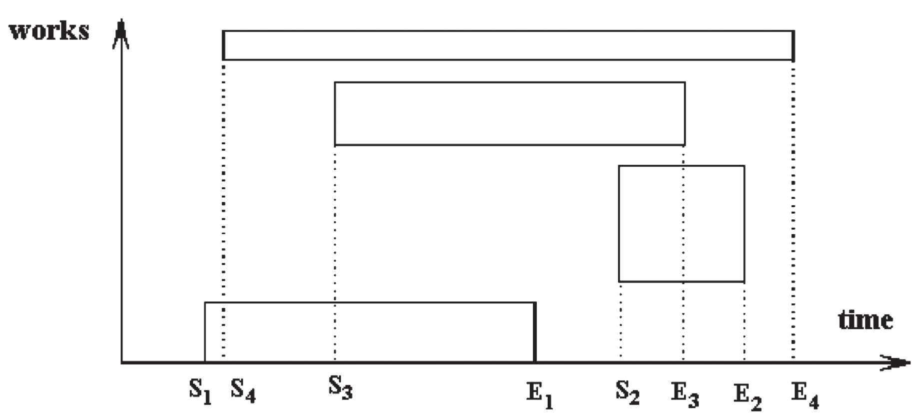

Fig. 1. *The initial intervals and their volumes*

During its performance each work cannot be interrupted and rate of its performance must be constant.

[1] Yakov Genis <yashag5@yahoo.com>
[2] Igor Ushakov <iushakov2000@yahoo.com>

The rate *rk* of performance of work *k* within the interval $[s_k, e_k]$ is equal to:

$$r_k(t) = \begin{cases} \dfrac{v_k}{e_k - s_k} & \text{if } t \in [e_k - s_k] \\ 0, & \text{otherwise} \end{cases} . \tag{2}$$

It is clear that the total rate of work performance for a given allocation of works, to be denoted as *G*, is equal to

$$R(t) = \sum_{k \in G} r_k(t). \tag{3}$$

where *G* is the chosen allocation of works.

Notice that for any chosen schedule function, *R(t)* is a step function of the type presented in Fig. 2.

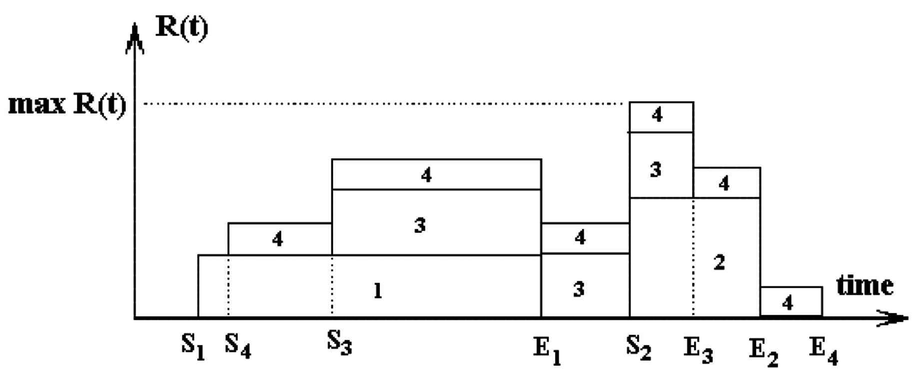

Fig. 2. *The initial distribution of work rates*

The problem is to find such subintervals $[s_k, e_k]$ that
a) the maximums of the sum of work rates should be minimal

$$\min_g \max_t R(t) \tag{4}$$

and/or the minimum(s) of the sum of work rates should be maximum

$$\max_g \min_t R(t) \tag{5}$$

where *g* is some allocation of intervals $[s_k, e_k]$;

b) the distribution of the total work rate *R(t)* on the whole maintenance interval has to be the most uniform-like (minimum difference between max $R(t)$ and min $R(t)$).

Note: The obtained *g* is not unique due to the discrete nature of time quanta.

2. Verbal description of the algorithm FV&CH

The title of the algorithm FV & CH is the abbreviation of its whole name "FILL THE VALLEYS & CUT THE HILLS". It is funny but the literal verbal description of this algorithm is given in the following words of the Gospel: "Every valley shall be filled, and every mountain and hill shall be brought down" (Saint Luke, Chapter 3, Verse 5).

But let us go from the Bible to mathematics, and give the strict (though a verbal) description of the algorithm that was implemented in Visual Basic.

So, we should find such splitting of subintervals (1) that allows us to deliver (4) and/or (5), and to make *R(t)* as uniformly distributed as possible under the given restrictions. The FV&CH program developed in Visual Basic has two buttons: "Cut Hills" and "Fill Valleys". The first button performs, and the second one (5). Choosing the sequence of activation of these buttons, we can find the splitting of subintervals (1), which aligning the distribution *R(t)* as much as possible.

Let us explain the algorithm using some illustrative example. Let there be five works with specified volumes v_k and corresponding admissible time intervals $[S_k, E_k]$, $k=\overline{1,5}$. For convenience, introduce discrete quanta of time with whose precision the time is measured. They may be equal to, for example, an hour, or 15 minutes, or one minute, etc. The initial data are given in Table 1, wherein the values d_1, d_2, d_3, d_4 and d_5 are some time slots. For example, the work No. 1 initially had to be started at the beginning of slot d_1 and had to be finished at the end of slot d_4.

Table 1 Initial distribution of works

Work No.	Volume	d_1	d_2	d_3	d_4	d_5
1	8	2	2	2	2	
2	9	3	3	3		
3	15		5	5	5	
4	12			4	4	4
5	3		1	1	1	
	9.4^{opt}	5	11	15^*	12	4_*

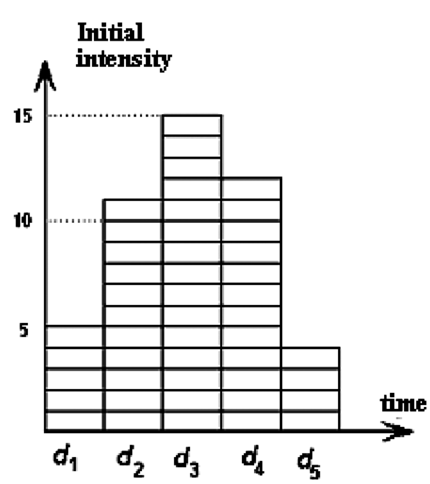

Fig. 3. *The initial total work rate distribution in the illustrative example*

The last value in the column "Volume" gives the optimal rate for the ideal case when we manage to distribute all works on the entire time interval staring from d_1 and ending by d_5. In the last row the superscript asterisk denotes the maximum rate and the subscript asterisk denotes the minimum rate for the initial distribution of works. The initial distribution of works is given in Fig.3.

Table 1 and Fig. 3 show that the maximum rate for initial work distribution locates at slot d_3. Let us find what works had to be fulfilled in slot d_3 according to the initial distribution and whose shifting to the left or to the right may decrease the maximum rate.

Step 1. Let us begin with work No.1. Since any work has to be fulfilled with no interruptions, we can move the beginning of this work either to the left or to the right of slot d_4. In this particular case, only shifting to the left leads to decrease of the peak load.

Note: At step 1, one may choose any other work that had to be performed at this interval according to the initial schedule. However, the peak rate should not increase, and the minimum rate should not decrease. The new schedule is given in Table 2.

Table 2 Step 1: Distribution after shifting work No.1

Work No.	Volume	d_1	d_2	d_3	d_4	d_5	Action
1	8	4	4				Shift to the left
2	9	3	3	3			
3	15		5	5	5		
4	12			4	4	4	
5	3		1	1	1		
	9.4	7	13^*	13^*	10	4_*	

We should try to shift other works as well. The best solutions of all will be chosen as the solution at the first step. The computer program performs these actions easily.

In this example, when doing actions manually, we intuitively select work No.4 as the next work subject to possible shifting. This work can be entirely moved to time slot d_5 wherein to be performed. Let us this shift "right-right". This action leads to improvement of the schedule. Note that this action seems to be the best one at the first step (see the Table 3).

Table 3 Step 2: Distribution after moving work No.4

Work No.	Volume	d_1	d_2	d_3	d_4	d_5	Action
1	8	2	2	2	2		
2	9	3	3	3			
3	15		5	5	5		
4	12			0	0	12	"Right-right"
5	3		1	1	1		
	9.4	5_*	11	11	8	12^*	

<u>Step 2</u>. This step should “fill the hole” in slot d_1. The entire work No.1 is moved to this slot (we call this move conditionally as “left-left”). The result is shown in Table 4.

Table 4. Step 2: Distribution after moving work No.1

Work No.	*Volume*	d_1	d_2	d_3	d_4	d_5	*Action*
1	8	8					“Left-left”
2	9	3	3	3			
3	15		5	5	5		
4	12			0	0	12	
5	3		1	1	1		
	9.4	11	9	9	6_*	12^*	

By this action we simultaneously “killed two birds with one stone”: we increased the minimum rate and made the rate distribution more uniform-like.

<u>Step 3</u>. Move the entire work No.5 to time slot d_4, i.e. do the procedure of shifting over this work called “right-right”. The result is shown in Table 5.

Table 5. Step 3: Distribution after moving work No.5

Work No.	*Volume*	d_1	d_2	d_3	d_4	d_5	*Action*
1	8	8					
2	9	3	3	3			
3	15		5	5	5		
4	12			0	0	12	
5	3		0	0	3		“Right-right”
	9.4	11	8	8	8_*	12^*	

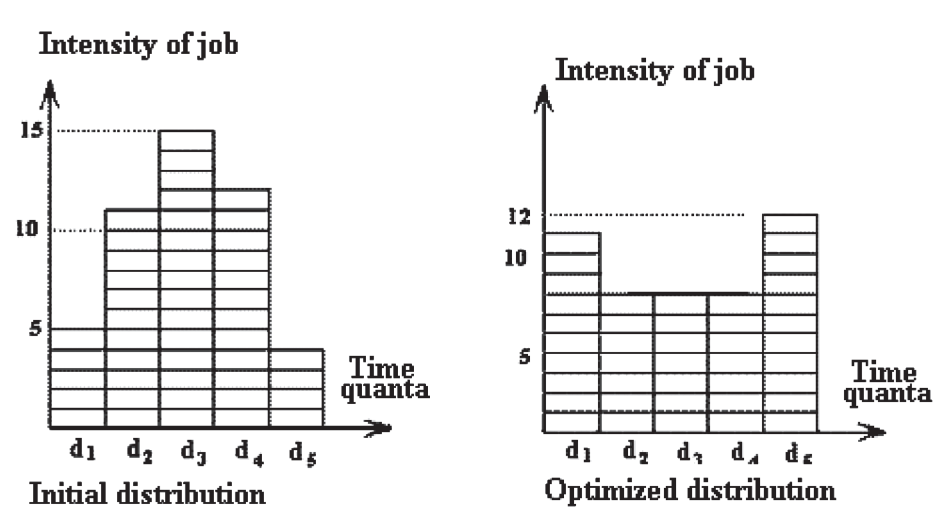

Fig. 4. *The comparison of the initial distribution and the distribution after optimization*

At this step we finish building the most uniform-like distribution with a minimum possible peak rate and maximum underuse.

The comparison of the initial distribution and the distribution after optimization is shown in Fig. 4.

Conclusion

1. The suggested algorithm FV&CH gives the strict solution of the problem in the sense of finding of the optimal uniform-like maintenance scheduling.

2. The obtained result is optimal (in terms of aligning rates by reaching the minimum of a maximum rate, and the maximum of a minimum rate), though is not unique.

3. The described algorithm is simple for programming.

4. The program has been developed using Visual Basic that applies this algorithm. The program has a simple and convenient interface and permits to work with an unlimited number of works with slots that may be as small as user needs. Everybody interested in the program, please contact the authors.