OPTIMAL UNIFORM-LIKE SCHEDULING OF MAINTENANCE

Yakov Genis[3]
New York, USA

Igor Ushakov[4]
San Diego, USA

The algorithm for the optimal scheduling of work performance is suggested. The every work's maintenance cannot be interrupted and it has an acceptable interval to be performed. The maintenance resources are limited. The optimal maintenance means that the distribution of the total sum of the rates of the works' maintenance should be made more uniform-like.

1. FORMULATION OF THE SCHEDULING PROBLEM

study

There are some "works" with volumes v_1, v_2, …, v_n (see an example in Fig.1). Each work, k, has to be fulfilled during interval $[s_k, e_k]$, which lies between is the allowed start moment, S_k, and permissible end moments, E_k, that are given in advance, i.e.

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

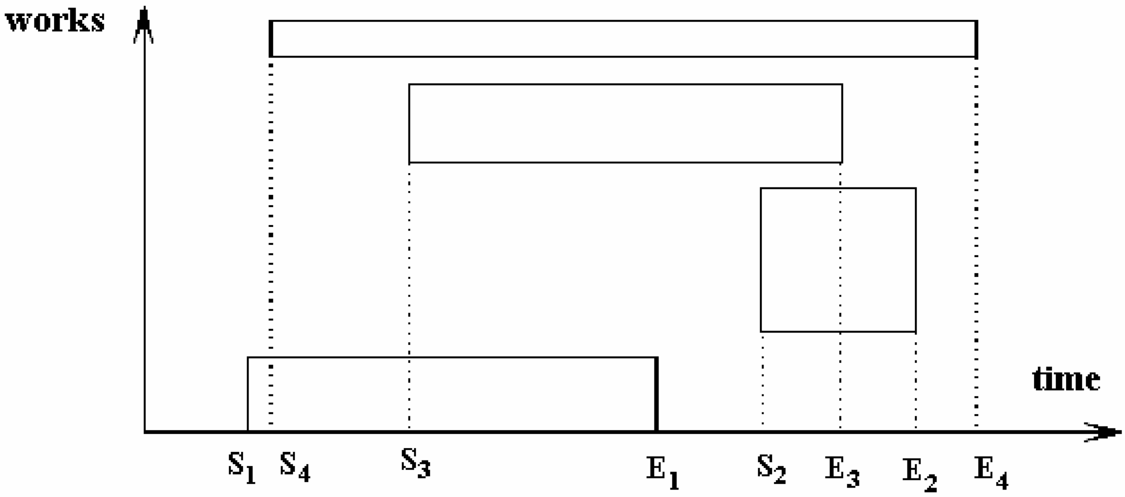

Fig. 1 The initial intervals and the volumes

[3] Yakov Genis <yashag5@yahoo.com>
[4] Igor Ushakov <iushakov2000@yahoo.com>

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

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

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

It is clear that the total rate of work performance for a given allocation of works is equal to

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

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 the Fig.2.

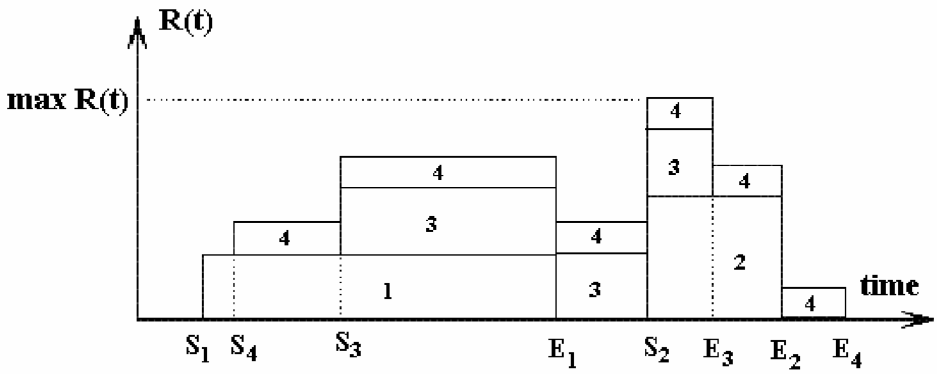

Fig. 2 The initial distribution of the work rates.

The problem is to find such subintervals $[s_k, e_k]$ that

a) the maximums of the sum of work rates should be minimal

$$\min_g \max_t R(t) \qquad (4)$$

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

$$\max_g \min_t R(t), \qquad (5)$$

where g is allocation of works, under the condition that all the works have been fulfilled ;

b) the distribution of the total work rate $R(t)$ has to be the most uniform-like on the whole maintenance interval.

Note: The obtained schedule is not unique because the (4) and the (5) may give different results due to discrete nature of the 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 next 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 тo mathematics, and give the strict (though a verbal) description of the algorithm that was implemented on Visual Basic.

We should find such set g of subintervals (1) to deliver (4) and/or (5), and to make $R(t)$ as uniformly distributed as it possible under the given restrictions. The developed FV&CH Visual Basic program has two subprograms: "Cut Hills" and "Fill Valleys". The sequential switch from one program to another allows finding the optimal solution in an interactive regime.

Explain the algorithm on an illustrative example. Let there be five works with volumes v_k and corresponding admissible time intervals $[S_k, E_k]$, $k=\overline{1,5}$. These data are given in the Table 1. We measure the time with accuracy of the discrete quantum (slot). It may be for example hour, or 15 minutes, or one minute, etc. In Table 1 values d_1, d_2, d_3, d_4, and d_5 are some slots. For example, the work #1 initially may be started at the beginning of the slot d_1 and has to be finished at the end of the slot d_4.

Table 1. The initial works distribution

Work #	**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_*

The value on the bottom of the column "Volume" gives the optimal rate for the ideal case when it would be permissible to perform each work during entire given interval (from d_1 to d_5), i.e. at any time the sum of work rates is constant. In the lower row the superscript asterisk denotes the maximum rate and the subscript asterisk denotes the minimum rate for the initial works' distribution. The initial works distribution is given in the Fig.3.

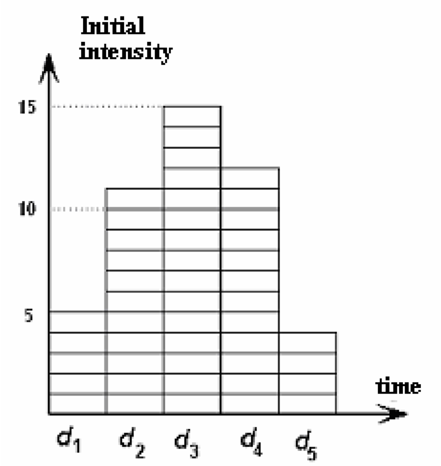

Fig. 3. The initial total work rate distribution.

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

> Step 1. Let us begin with work #1. Since any work has to be fulfilled with no interruptions, we can move the beginning of this work to slot d_4 or move it to the left, so it will be started at moment S_1 and have fulfilled before slot d_3. In this concrete case, the only possibility to make shift to the left.
>
> Note: At the step 1, one may start with moving any work that covers slot d_3 (maximum total rate), however for the algorithm description some ordering of works is necessary

The new distribution is given in the Table 2.

Table 2. Step 1: Distribution after moving work #1

Work #	**Volume**	d_1	d_2	d_3	d_4	d_5	***Action***
1	8	4	4				Move "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*	

The computer algorithm at Step 1 tried to move other works and remember the best solution (a champion) of all of them before moving to Step 2.

In this example (doing actions manually) we select the work #4 for moving to the right. Moreover, avoiding intermediate steps, we move entire this work to slot d_5 (we call this move conditionally as "right-right"). This action decreases the peak rate and also improves the rate's distribution. Indeed this action is now the best (see the Table 3).

Table 3. Step 2: Distribution after moving work #4 .

Work #	**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	Move "right-right"
5	3		1	1	1		
	9.4	5_*	11	11	8	12^*	

Step 3. This step should "fill the hole" in slot d_1. Entire work #1 is moved to this slot (we call this move conditionally as "left-left"). The result is shown in Table 4.

Table 4. Step 3: Distribution after moving work #1.

Work #	**Volume**	d_1	d_2	d_3	d_4	d_5	**Action**
1	8	8					Move "left-le
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 did the rate's distribution more uniform-like.

Step 4. Move entire work #5 to the d_4. The result is shown in the Table 5.

Table 5. Step 4: : Distribution after moving work #5.

Work #	**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		Move "right-right"
	9.4	11	8	8	8_*	12^*	

This step is final.

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

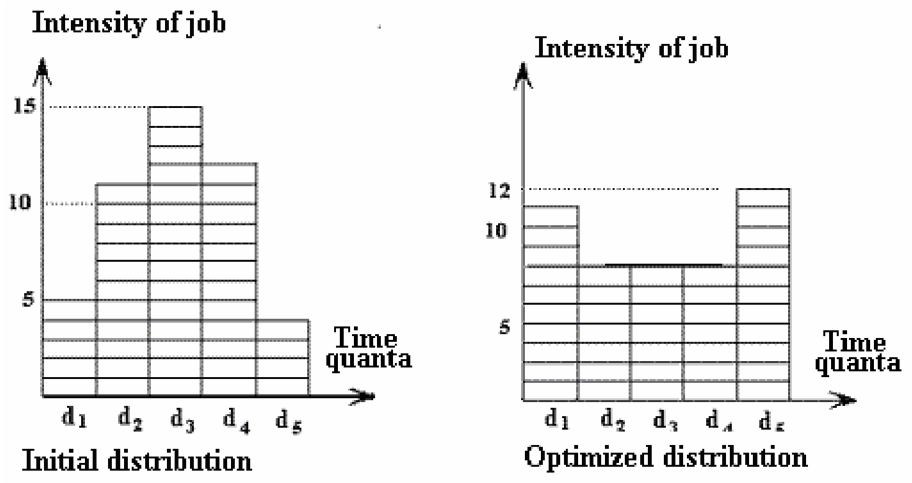

Fig. 4. The comparison of the initial distribution with the distribution after the optimization

SUMMARY

1. The suggested algorithm FV&CH gives the strict decision of the problem in the sense of finding of the optimal uniform-like maintenance’s scheduling.
2. The obtained result is optimal (in mentioned above terms), though is not unique.
3. The described algorithm is simple for programming.
4. There is developed a program on Visual Basic that uses this algorithm. The program has a simple and convenient interface and permits to work with unlimited number of works with slots that may be as small as user needs. Everybody interested in the program, please ask the authors.