PHYSICS-TECHNOLOGY INSTITUTE FOR LOW TEMPERATURES
ACADEMY OF SCIENCES OF THE UKRAINIAN SSR

NUMERICAL MATHEMATICS AND COMPUTER TECHNOLOGY
1972, VOLUME 3

On the comparative complexity of some discrete optimization problems[1]

E.M. Livshits, V.I. Rublinetsky

Kharkov

Suppose that we have n stones. We have to arrange them in two piles of maximal similarity by weight. For this problem no simple algorithm is known, although as a special case of the knapsack problem it can be solved by enumerative methods [10, 12].

For many discrete optimization problems in turns out that, in order to solve them, one needs to know how to solve the stones problem; in that sense, such problems are harder than the stones problem. This type of comparative analysis, which does not require knowledge of the absolute complexity, seems to be useful to us; in any case, when solving discrete optimization problems, we have since long used this type of analysis as a test, and when finding "stones" we do not waste our time looking for simple algorithms.

In this paper we introduce a concept of reducing a given problem to another one, which allows us to say that a problem is not harder than another problem in the sense mentioned above, provided that the reduction itself is simple. We will show that the stones problem is a sufficiently universal minorant, since it can be reduced to many well-known discrete optimization problems in a simple way.

§1. General concepts

Let A be the input of a given problem and let a function $F(A, v)$ be defined for each $A \in X$ and $v \in V$, where V is a finite set. A *discrete optimization problem* T is the problem of finding a $\breve{v}(A) \in \breve{V}(A)$, $\breve{V}(A) \subseteq V$, such that

$$\min_{v \in V} F(A, v) = F(A, \breve{v}).$$

In this way, the problem T defines a mapping $T(A) = \breve{V}(A)$.

A *subproblem* T' is a restriction of T to X', $X' \subseteq X$, i.e., the inputs of the subproblem are taken from a subset $X' \subseteq X$, notation $T' \trianglelefteq T$.

Definition. Problem T_1 *directly reduces* to problem T_2 if there exist surjections $\alpha : X_1 \to X_2$ and $\beta : V_2 \to V_1$ such that

$$T_1 = \beta T_2 \alpha.$$

Such a relation is clearly reflexive and transitive.

Problem T_0 reduces to problem T if there exists a subproblem T' to which T_0 reduces directly. If $T_0 = \beta T' \alpha$, then the pair (α, β) gives a *reduction* of T_0 to T. This relation is also a partial order.

[1] This is a translation of a paper that was written in Russian and published in 1972. The translation was prepared by Jan Karel Lenstra, Vitaly Strusevich and Milan Vlach. They review the approach and the results of Livshits and Rublinetsky in "A historical note on the complexity of scheduling problems", *Operations Research Letters 51* (2023) 1–2.

We shall comment on our definitions. If T_0 reduces to T, then an individual problem $T_0(A_0)$ can be solved as follows: 1) transform the input A_0 into the input A' using α, 2) find $\breve{V}(A')$ by solving problem $T(A')$, 3) transform the solution $\breve{V}(A')$ into $\breve{V}_0(A_0)$ using β. If in this reduction the transformation $\alpha(A_0) = A'$ and the backward transformation β are computationally simple, then the reduction of T_0 to T gives an intuitively understandable way to solve problem T_0 by the same algorithm as for T, i.e., it gives evidence that T is no less complex. We will not define more precisely what is meant by "computationally simple", since in what follows all reductions will be given explicitly and need $\sim n$ arithmetic operations.

In the sequel an input A is given by an n-tuple $(a_1, a_2, \dots, a_n)$; we will call the elements $a_i \in \Omega$ operations, bearing in mind that the problems we shall deal with relate to scheduling theory.

The *stones problem* T_0 has input $a_i = p_i$, $p_i \geq 0$, i.e., $\Omega = R_1^+$, $X = R_n^+$. We have to determine the minimum of the function

$$F(A_0, w) = \Big| \sum_{i \in w_1} p_i - \sum_{j \in w_2} p_j \Big|,$$

where w_1 is a subset of indices from $I = (1, 2, \dots, n)$, $w_2 = I - w_1$, $w = (w_1, w_2)$.

For convenience we introduce the function

$$\Delta(A_0, w) = \Big| \frac{1}{2} \sum_{i=1}^{n} p_i - \sum_{i \in w_1} p_i \Big| = \Big| \frac{1}{2} \sum_{i=1}^{n} p_i - \sum_{i \in w_2} p_i \Big|.$$

Since $F(A_0, w) = 2\Delta(A_0, w)$, problems with these objective functions have the same optimal solutions. We shall always denote

$$\sigma_1 = \sum_{i \in w_1} p_i; \;\; \sigma_2 = \sum_{i \in w_2} p_i; \;\; \sigma = \frac{1}{2} \sum_{i=1}^{n} p_i; \;\; |w_1| = k; \;\; |w_2| = n - k.$$

The following assertion will be useful to us: if $F_2 = \phi(F_1)$ and ϕ is an increasing function, then problem $T_1 = (X, F_1, V)$ reduces to $T_2 = (X, F_2, V)$ by the identity mappings α and β.

§2. Scheduling problems with one or two processors

Given is a single processor which has to perform n operations. For operation i we have a processing time p_i, $p_i > 0$, a point in time $r_i \geq 0$ after which the operation can start, and a nondecreasing function $\phi_i(t)$ – the penalty for completing operation i at time t. We thus have

$$a_i = (p_i, r_i, \phi_i(t)).$$

A schedule s consists of a sequence of the operations and is defined by a set of the completion times $t_i(s)$, $i = 1, 2, \dots, n$, where

$$t_j(s) = \max(r_{[j]}, t_{[j-1]}(s)) + p_{[j]}. \tag{1}$$

From here on we use $[j]$ to denote the operation that occupies the jth position in the sequence.

In minmax problems the objective function has the form

$$F(A, s) = \max_{1 \leq i \leq n} \phi_i(t_i(s)),$$

in minsum problems the objective function has the form

$$F(A, s) = \sum \phi_i(t_i(s)).$$

In both cases we have to find the minimum of F over all feasible schedules s.

In the case of two identical processors, a schedule $v = (w_1, w_2, s_1, s_2)$ splits the operations into two groups w_1 and w_2, to be performed by the first and the second processor, respectively; on the processors schedules s_1 and s_2 satisfying condition (1) are given. The objective function for minmax problems has the form

$$F(A, v) = \max(F(A, s_1), F(A, s_2))$$

and for minsum problems

$$F(A, v) = F(A, s_1) + F(A, s_2).$$

Such problems are described, for example, in the book [8], which also gives references to the literature.

1. Minmax problems. Jackson [11] considered the problem of minimizing the maximum penalty on a single processor, which has to perform n operations with parameters

$$p_i > 0, \;\; r_i = 0, \;\; \phi_i(t) = \begin{cases} 0, & t < d_i, \\ t - d_i, & t \geq d_i. \end{cases}$$

In this problem we have that operation $a_i = (p_i, 0, d_i)$. The problem is solved by sequencing the operations in increasing order of d_i. A more general problem with arbitrary nondecreasing penalty functions also admits a simple solution [4, 7]. In applications, in particular for optimization problems in industrial control systems, one has to solve Jackson's problem for several processors or for $r_i \neq 0$ [5, 6]. A simple solution to these problems is not known.

Proposition 1. *The stones problem reduces to Jackson's problem on two processors.*

Proof. Consider subproblem $T' \trianglelefteq T$ defined by

$$a' = (p, 0, 0), \;\; X' = \Omega'^n = R_n^+.$$

The objective function of subproblem T' is

$$F(A', v) = \max(\max_{i \in w_1} \phi(t_i(s_1)), \max_{j \in w_2} \phi(t_j(s_2))) = \max\left(\sum_{i \in w_1} p_i, \sum_{j \in w_2} p_j\right) = \max(\sigma_1, \sigma_2) = \Delta + \sigma.$$

This implies that, if we define $\alpha(A_0)$, $A_0 = (p_1, \ldots, p_n)$, by $a'_i = \alpha(A_0) = (p_i, 0, 0)$ and use β to transform the schedule $v = (w_1, w_2, s_1, s_2)$ into $\beta(v) = (w_1, w_2) = w$, then $T_0(A_0) = \beta T' \alpha(A_0)$. The stones problem has been reduced to T.

Proposition 2. *The stones problem reduces to Jackson's problem with $r_i \neq 0$.*

Proof. Consider problem T with $n+1$ operations and take subproblem T' with operations

$$\begin{aligned} a'_0 &= (p_0, \; \sigma, \; \sigma + p_0), \\ a'_i &= (p_i, \; 0, \; 2\sigma + p_0), \;\; i = 1, 2, \ldots, n, \end{aligned} \tag{2}$$

where $p_0 = \text{const}$, $X' \cong R_n^+$. In subproblem T' we have

$$\begin{aligned} F(A', s) &= \max(\max_{i \in w_1} \phi_i(t_i(s)), \phi_0(\max(\sigma_1, \sigma) + p_0), \max_{j \in w_2} \phi_j(t_j(s))) = \\ &= \max(\phi(\sigma_1), \phi_0(\max(\sigma_1, \sigma) + p_0), \phi(\max(\sigma_1, \sigma_0) + p_0 + \sigma_2)). \end{aligned}$$

Here and in similar cases below w_1 denotes the set of operations that precede a_0 in the schedule and w_2 those that follow a_0. For $\sigma_1 > \sigma$ we have $F = \max(0, \sigma_1 - \sigma, 0)$. For $\sigma_1 \leq \sigma$ we have $F = \max(0, 0, \sigma_2 - \sigma)$. It follows that

$$F(A', s) = |\sigma_1 - \sigma| = \Delta.$$

Hence, if α is given by formula (2) and the schedule s is transformed by β into $\beta(s) = (w_1, w_2)$, then $T_0(A_0) = \beta T' \alpha(A_0)$.

We have shown that the problem with n stones reduces to a problem with n operations in case of two processors and to a problem with $n+1$ operations in case of a single processor. In the sequel we shall proceed analogously. In the former case the reduction required no computations, in the latter case we needed to compute $\sigma + p_0$ and $2\sigma + p_0$. The backward transformation β was trivial. In what follows we shall not present α and β explicitly; we only note that further below the description of the subproblem defines α as it did in the proof of Proposition 2.

2. McNaughton's problem. The well-known problem of McNaughton [14] on a single processor concerns the minimization of the total penalty for the operations

$$a_i = (p_i > 0,\ r_i = 0,\ \phi_i(t) = c_i t),\quad i = 1, 2, \ldots, n.$$

This problem is solved by placing the operations in decreasing order of c_i/p_i. In [14] and subsequent publications the problem has been generalized in several ways. The problem with $\phi(t) = c_i e^{rt}$ admits a simple solution [17]. It has been shown that for other classes of smooth nondecreasing functions simple solutions of this type do not exist [3]. Here we shall demonstrate that generalizations of the problem either with two processors or with arbitrary r_i are not easier than the stones problem.

Proposition 3. *The stones problem reduces to McNaughton's problem with two processors.*

Proof. Take subproblem T_1', $T' \trianglelefteq T$, with operations

$$a_i' = (p_i > 0,\ r_i = 0,\ \phi_i(t) = p_i t).$$

The contribution to the objective function for the schedule $v = (w_1, w_2, s_1, s_2)$ is

$$\begin{aligned} F_1(A_1', w_1, s_1) &= p_{[1]}p_{[1]} + p_{[2]}(p_{[1]} + p_{[2]}) + \cdots + p_{[k]}(p_{[1]} + \cdots + p_{[k]}) = \\ &= \sum_{i \in w_1} p_i^2 + \sum_{i<j, i,j \in w_1} p_i p_j = \frac{1}{2} \sum_{i \in w_1} p_i^2 + \frac{1}{2}\sigma_1^2. \end{aligned}$$

We calculate F_2 analogously. Hence, the objective function is equal to $F = 2\Delta^2 + (2\sigma^2 + \frac{1}{2}\sum_{i=1}^n p_i^2)$. The equality we have thus obtained proves the reducibility of the stones problem to T.

Proposition 4. *The stones problem reduces to McNaughton's problem with a single processor and* $r_i \neq 0$.

Proof. Consider problem T with $n+1$ operations and define a subproblem T' of T with operations

$$\begin{aligned} a_0' &= (p_0,\ r_0 = \sigma,\ \phi_0 = c_0 t),\quad c_0 = 2(\sigma + p_0), \\ a_i' &= (p_i,\ r_i = 0,\ \phi_i = p_i t),\quad i = 1, 2, \ldots, n. \end{aligned}$$

The contribution of the operations preceding a_0 is

$$F_1(A', w_1) = \frac{1}{2}\sigma_1^2 + \frac{1}{2}\sum_{i \in w_1} p_i^2.$$

The contribution of the operations following a_0 is

$$F_2(A', w_2) = t_0\sigma_2 + \frac{1}{2}\sum_{i \in w_2} p_i^2 + \frac{1}{2}\sigma_2^2.$$

Here $t_0 = \max(\sigma_1, \sigma) + p_0$ is the completion time of a_0. Additionally, taking into consideration the contribution of a_0 we obtain

$$F(A', v) = \frac{1}{2}(\sigma_1^2 + \sigma_2^2) + \sigma_2 t_0 + c_0 t_0 + \frac{1}{2}\sum_{i=1}^n p_i^2.$$

We shall consider two cases: $\sigma_1 \geq \sigma$ and $\sigma_1 < \sigma$.

1) $\sigma_1 \geq \sigma$. Then $\sigma_1 = \sigma + \Delta$, $\sigma_2 = \sigma - \Delta$, and the objective function is equal to

$$F^{(1)}(A', v) = \Delta(\sigma + p_0) + \Big(4\sigma^2 + 5p_0\sigma + 2p_0^2 + \frac{1}{2}\sum_{i=1}^{n} p_i^2\Big).$$

2) $\sigma_1 < \sigma$. Then $\sigma_1 = \sigma - \Delta$, $\sigma_2 = \sigma + \Delta$, and the objective function is equal to

$$F^{(2)}(A', v) = \Delta^2 + F^{(1)}(A', v).$$

The minimum is achieved in the first case, since $F^{(1)} < F^{(2)}$. $F^{(1)}$ attains its minimum simultaneously with Δ, which proves the proposition.

3. Minimizing total weighted tardiness. This problem concerns the minimization of the total penalty on a single processor for the operations

$$a_i = \Big(p_i,\ \ r_i = 0,\ \ \phi_i = \begin{cases} 0, & t < d_i \\ c_i(t - d_i), & t \geq d_i \end{cases}\Big).$$

There is a vast literature on this problem [9, 16, 18, 19]; some special cases have been solved, lower and upper bounds have been obtained, there are enumerative algorithms and heuristic procedures.

Proposition 5. *The stones problem reduces to the problem of minimizing total weighted tardiness.*

Proof. Consider problem T with $n+1$ operations and define a subproblem of T with operations

$$a_0' = \Big(p_0,\ r_0 = 0,\ \phi_0 = \begin{cases} 0, \ t < \sigma + p_0 \\ c_0(t - (\sigma + p_0)) \end{cases}\Big),\ c_0 = 2p_0,$$
$$a_i' = (p_i,\ r_i = 0,\ \phi_i = p_i t),\ \ i = 1, 2, \ldots, n.$$

The objective function for subproblem T' is

$$F(A', v) = \frac{1}{2}(\sigma_1^2 + \sigma_2^2) + \sigma_1\sigma_2 + \sigma_2 p_0 + \phi_0(\sigma_1 + p_0) + \frac{1}{2}\sum_{i=1}^{n} p_i^2.$$

1) If $\sigma_1 \leq \sigma$, then $\sigma_1 = \sigma - \Delta$, $\sigma_2 = \sigma + \Delta$, and in this case the objective function is

$$F^{(1)} = \Delta p_0 + \Big(2\sigma^2 + \sigma p_0 + \frac{1}{2}\sum_{i=1}^{n} p_i^2\Big).$$

2) If $\sigma_1 > \sigma$, then $\sigma_1 = \sigma + \Delta$, $\sigma_2 = \sigma - \Delta$, and in this case the objective function is

$$F^{(2)} = \Delta p_0 + \Big(2\sigma^2 + \sigma p_0 + \frac{1}{2}\sum_{i=1}^{n} p_i^2\Big).$$

Both functions are the same and attain their minimum simultaneously with Δ.

4. Minimizing the weighted number of late jobs. We have to minimize the sum of the penalties for processing the operations

$$a_i = \Big(p_i,\ \ r_i = 0,\ \ \phi_i = \begin{cases} 0, & t \leq d_i \\ c_i, & t > d_i \end{cases}\Big).$$

For the unweighted case ($c_i = 1$) the problem admits a simple solution [15]. For the weighted case Lawler and Moore [12] proposed a dynamic programming algorithm and noticed the similarity of the functional equations occurring in this problem to those in the knapsack problem. Maxwell [13], explaining this similarity, formulated a subproblem for which it is necessary to solve the corresponding knapsack problem.

Proposition 6. *The stones problem reduces to the problem of minimizing the weighted number of late jobs.*

Proof. Consider subproblem T' with operations

$$a'_0 = \left(p_0 = 2\sigma,\ r_0 = 0,\ \phi_0 = \begin{cases} 0, & t \le 3\sigma \\ 2\sigma, & t > 3\sigma \end{cases}\right),$$
$$a'_i = \left(p_i,\ r_i = 0,\ \phi_i = \begin{cases} 0, & t \le 2\sigma \\ p_i, & t > 2\sigma \end{cases}\right),\ i = 1, 2, \ldots, n.$$

The objective function decomposes into three components:

$$F(A'v) = F_1(A', w_1) + \phi_0(\sigma_1 + 2\sigma) + F_2(A'_1, w_2).$$

1) Let $\sigma_1 \le \sigma$. Then $F_1 = 0$, $\phi_0(\sigma_1 + 2\sigma) = 0$, and each operation from w_2 is late, i.e. $\phi_i = p_i$, so that

$$F^{(1)} = \sigma_2 = \sigma + \Delta.$$

2) Let $\sigma_1 > \sigma$. Then $F_1 = 0$, $\phi_0 = 3\sigma$, $F_2 = \sigma_2$, and

$$F^{(2)} = 3\sigma + \sigma_2 = 4\sigma - \Delta.$$

Since $\Delta \le \sigma$, it follows that $F^{(1)} \le F^{(2)}$, and hence the objective function of the subproblem attains its minimum in the first case. Reducibility has been proved.

§3. Three-stage scheduling problems

Given are three types of processors, M_1, M_2, M_3, generally not necessarily different, on which we have to perform n operations. Operation i consists of three stages; the first stage requires a processing time p_i on processor M_1, the second one – q_i on M_2, the third one – r_i on M_3. $p_i, q_i, r_i \ge 0$. For each type, either there is a single processor or there are "many". By the latter we mean that the processor can perform an arbitrary number of operations at the same time; we shall say that such processors are of type 0. There are no constraints on the starting times of the stages, except those implied by the occupancy of the processors and by the requirement not to start a subsequent operation before the preceding operation has been completed. We have to minimize the total time needed to perform all operations over all schedules $v = (s_1, s_2, s_3)$, where s_k is the order in which the operations in the kth stage are performed.

Different problems will be denoted by different triples (M_1, M_2, M_3). For example, the three-machine problem of Johnson [2] is denoted by $(1, 2, 3)$.

If we define two-stage problems analogously, then there are five of them: $(1, 2)$, $(1, 0)$, $(0, 2)$, $(0, 0)$ and $(1, 1)$. The first problem is Johnson's two-machine problem; it admits a simple solution from a computational point of view. The second one, close to Jackson's problem, is solved by sequencing the operations in decreasing order of q_i. The solution of the third problem – a "mirror reflection" of the second problem, while the fourth and the fifth problems are trivial.

The three-stage problems are somewhat different. If two consecutive symbols in the triple (M_1, M_2, M_3) are identical, then the problem reduces to a two-machine problem in which these two symbols appear, and it admits a simple solution. The other problems, which we call *essentially three-stage*, do not admit a simple solution. The purpose of the following exposition will be to prove that *the stones problem reduces to each essentially three-stage problem*.

We shall first consider essentially three-stage problems in which the second processor is of a nonzero type. These are Johnson's problem $(1, 2, 3)$, the problem $(0, 2, 0)$, which was studied in [5, 6], the problem $(0, 2, 3)$, which may be interpreted as Johnson's two-machine problem with different release times for the operations in the first stage, and its "mirror reflection" $(1, 2, 0)$. Reference [5] demonstrated that the stones problem reduces to problem $(0, 2, 0)$. This reduction is applicable to all problems we mentioned.

Proposition 7. *The stones problem reduces to each essentially three-stage problem in which the second processor is of a nonzero type.*

Proof. For all problems mentioned take a corresponding subproblem with operations

$$a'_0 = (p_0 = \sigma,\ q_0,\ r_0 = \sigma),$$
$$a'_i = (p_i = 0,\ q_i,\ r_i = 0),\ \ \sum_{i=1}^n q_i = 2\sigma.$$

In each subproblem we need to put operation a_0 with processing time p_0 on processor M_1, from a point in time $\tau > p_0$ the second stage of operation a_0 with processing time q_0 is being performed on processor M_2, and immediately after that the third stage of operation a_0 with processing time r_0 is being performed. Different schedules differ in the sets w_1 and w_2 of second-stage operations that, on processor M_2, precede or follow the operation of length q_0. If $\sum_{i\in w_1} q_i < \sigma$, then the objective function has value $p_0 + q_0 + \sum_{j\in w_2} q_j$; in the opposite case the objective function has value $\sum_{i\in w_1} q_i + q_0 + r_0$. In all cases

$$F(A', v) = 2\sigma + q_0 + \Delta.$$

The objective function attains its minimum value simultaneously with Δ. The reduction has been completed.

Two remaining essentially three-stage problems are described by the triple $(M_1, 0, M_3)$. Problem $(1, 0, 3)$ was investigated in papers by Jackson, Mitten and Johnson ([8], Ch. 5), and the "editor's problem" was formulated in [1].

Proposition 8. *The stones problem reduces to each essentially three-stage problem in which the second processor is of type* 0.

Proof. Consider problem $(1, 0, 3)$ with $n + 1$ operations and define its subproblem T' with operations

$$a'_0 = (p_0 = \sigma,\ q_0 = d + 2\sigma,\ r_0 = 0),$$
$$a'_i = (p_i,\ q_i = d - p_i,\ r_i = 2p_i),\ \ \sum_{i=1}^n p_i = 2\sigma,\ \ d \geq 5\sigma.$$

First consider a schedule for a'_i, $i = 1, 2, \ldots, n$, only. It is easy to verify that we obtain an optimal solution if we take any sequence of operations in the first stage and the same sequence in the third stage. Hence, the objective function has value

$$F_1 = d + \sum_{i=1}^n r_i = d + 2\sum_{i=1}^n p_i = d + 4\sigma.$$

If we now add $a''_0 = (p_0, 0, 0)$ to these operations, then the objective function has value

$$F_2 = \max\Big(F_1, \sum_{i\in w_1} p_i + p_0 + d + \sum_{j\in w_2} r_j\Big).$$

Hence, for subproblem T' with $a'_0 = (p_0, q_0, 0)$ the objective function has value $F = \max(F_1, F_2,$ $\sum_{i\in w_1} p_i + p_0 + q_0)$. Substituting $p_0 = \sigma$, $q_0 = d + 2\sigma$, we obtain $F = \Delta + 4\sigma + d$. The proposition has been proved for problem $(1, 0, 3)$. Since in the construction the operations on M_1 and M_3 were distributed over time by the choice of q_0, one can assume that $M_1 = M_3$ with the same effect, i.e., the proposition has also been proved for problem $(1, 0, 1)$.

Literature

1. V.N. Burkov, C.E. Lovetski. Combinatorics and technology development. Znanie, 1968 (in Russian).
2. S.M. Johnson. Calendar planning. Progress, 1966 (in Russian).
3. G.K. Kladov, E.M. Livshits. Kybernetika 6, 33–41, 1968 (in Russian).
4. E.M. Livshits. Proceedings of the First Winter Workshop on Math. Progr., Vol. 3, 474–476, 1969 (in Russian).
5. E.M. Livshits. Proceedings of the First Winter Workshop on Math. Progr., Vol. 3, 477–497, 1969 (in Russian).

6. V.I. Rublinetsky. Proceedings of the First Winter Workshop on Math. Progr., Vol. 3, 523–530, 1969 (in Russian).
7. V.V. Shkurba et al. Scheduling problems and solution methods. Naukova dumka, Kiev (in Russian).
8. R.W. Conway, W.L. Maxwell, L.W. Miller. Theory of Scheduling. Addison-Wesley Publ. Co. Reading, Mass., 1967.
9. W.L. Eastman et al. Mgmt. Sci. 11.2, 268–279, 1964.
10. H. Greenberg, R.L. Hegerich. Mgmt. Sci. 16.5, 327–332, 1970.
11. J.R. Jackson. Mgmt. Sci. Res. Project, Res. Report 43, UCLA, 1955.
12. E.L. Lawler, J.M. Moore. Mgmt. Sci. 16.1, 77–84, 1969.
13. W.L. Maxwell. Mgmt. Sci. 16.5, 295–297, 1970.
14. R. McNaughton. Mgmt. Sci. 6.1, 1–12, 1959.
15. J.M. Moore. Mgmt. Sci. 15.1, 102–109, 1968.
16. J.G. Root. Mgmt. Sci. 11.3, 460–475, 1965.
17. M.H. Rothkopf. Mgmt. Sci. 12.5, 437–447, 1966.
18. A. Schild, I. Fredman. Mgmt. Sci. 7.3, 280–285, 1961.
19. A. Schild, I. Fredman. Mgmt. Sci. 8.1, 73–81, 1962.