DIGRAPH DESCRIPTION OF K-INTERCHANGE TECHNIQUE FOR OPTIMIZATION OVER PERMUTATIONS AND ADAPTIVE ALGORITHM SYSTEM
(Foundation of Computing and Decision Sciences, 26/3, 225-235, 2001)

Mark Sh. LEVIN*

Abstract. The paper describes a general glance to the use of element exchange techniques for optimization over permutations. A multi-level description of problems is proposed which is a fundamental to understand nature and complexity of optimization problems over permutations (e.g., ordering, scheduling, traveling salesman problem). The description is based on permutation neighborhoods of several kinds (e.g., by improvement of an objective function). Our proposed operational digraph and its kinds can be considered as a way to understand convexity and polynomial solvability for combinatorial optimization problems over permutations. Issues of an analysis of problems and a design of hierarchical heuristics are discussed. The discussion leads to a multi-level adaptive algorithm system which analyzes an individual problem and selects / designs a solving strategy (trajectory).

1. Introduction

For many years efforts of researchers in combinatorial optimization were oriented to the design of effective (polynomial) algorithms for problems on permutations. Scheduling problems and linear ordering problems are representatives of the problems over permutations. In many cases, effective algorithms are based on the use of local optimization techniques as *two neighbor elements exchange* techniques which effectively lead to a global optimum. The following works can be pointed out, for example: Adolphson and Hu [1], Borie [2], Conway et al. [3], Hardy et al. [6], Johnson [7], Levin [8], Monma and Sidney [10], Sidney [13], and Smith [15].

Based on the work of Smith, Elmaghraby proposed a graph-theoretical interpretation for the corresponding 2-search problem (i.e., interchange of two neighbor elements) [4].

This paper describes an extension of the result of Elmaghraby for k-search problems. Here a general multiple level digraph-description for the optimization problems

* Ben-Gurion University, Beer Sheva, Israel; email: mslevin@acm.org

over permutations on the basis of k-element exchange ($k = n, (n-1), ..., 2$) is proposed [14]. The viewpoint herein provides insight which can be incorporated into the design and analysis of a hierarchical algorithm system. The system involves a control unit with the following functions: (a) an analysis of individual problems, (b) the selection / design of a solving strategy, and (c) on-line adaptation of the problem solving process. The described material is a research in progress.

2. Graph Description

2.1. Formulation of Problem Instance

In this section, an example is presented as a basis for our further problem analysis and formulation. The following problem instance is considered. There is a set of elements S and a function $f: S \Rightarrow R$. The problem is:

$$\mathit{Find} \quad \min\{f(s)|s \in S\}.$$

A *Neighborhood* $A(s_o) \subseteq S$ is associated with $\forall s \in S$ such that $s \in A(s_o)$.

Neighborhood Search is described as follows: Given $\forall s \in S$, try to find $t \in A(s)$ such that $f(t) < f(s)$. If no such t exists, then s is locally optimal STOP. Otherwise, replace s by t and repeat until a local optimum is found.

Given a set $A = \{A(s)|s \in S\}$ we are particularly interested in the question of whether neighborhood search using A is guaranteed to arrive at local optimum which is a global optimum as well.

From now on we will focus upon problems with $S =$ *the set of permutations of* $\{1, ..., n\}$. We will define the $k-$*neighborhood* $A_k(s)$ of $s \in S$ for $k \leq n-1$ to be the set of permutations which can be obtained from s by selecting k adjacent elements of s and replacing them with any permutation of these k elements (obviously $s \in A_k(s)$). Let $A(k) = \{A_k|s \in S\}$.

A $k-$*search algorithm* is a neighborhood search algorithm which uses $A(k)$ for its neighborhoods. The $2-$*search algorithm* is the basic *adjacent interchange algorithm*. This method is well-known to be optimal for several sequencing problems without precedence constraints, e.g., weighted average completion time [15], two-machine flow shop [7], etc. The following question is posed (Jeffrey B. Sidney):

Are there objective functions $f: S \Rightarrow R$ for which the $k-$*search algorithm* always produces an optimal solution, but the $(k-1)-$*search algorithm* does not, for some $k \geq 3$?

The answer is YES.

Let $n = 4$ and set $f((1,2,3,4)) = 0$. Define $f(s)$ to be the minimum number of neighborhood search steps that must be executed using neighborhoods $A(3)$ as the set of neighborhoods to convert the permutation s to the permutation $(1,2,3,4)$. This function is tabulated in Table 1. Then the $3-$*search algorithm* is sure to yield the optimum, while $2-$*search algorithm* will not.

Consider $s = (4,3,1,2)$. Then $f(s) = 2$. Using pairwise interchange ($2-$*search*), we may reach the following other neighborhood points:

$(3,4,1,2)$ with $f(3,4,1,2) = 2$,

$(4,1,3,2)$ with $f(4,1,3,2) = 2$, and

$(4, 3, 2, 1)$ with $f(4, 3, 2, 1) = 3$.

Thus, pairwise interchange ($2-$*search*) terminates with the local optimum $(4, 3, 1, 2)$, while ($3-$*search*) finds the optimum in two iterations of neighborhood search.

Table 1. Numerical Example

No	s	$f(s)$	No	s	$f(s)$
1	3124	1	**13**	1234	0
2	3142	2	**14**	1243	1
3	3214	1	**15**	1324	1
4	3241	2	**16**	1342	1
5	3412	2	**17**	1423	1
6	3421	2	**18**	1432	1
7	4123	2	**19**	2134	1
8	4132	2	**20**	2143	2
9	4213	2	**21**	2314	1
10	4231	3	**22**	2341	2
11	4312	2	**23**	2413	2
12	4321	3	**24**	2431	2

2.2. General Description

Here a graph description of the initial domain of permutations (an analogue of argument space X for a function $f(x), x \in X$) is examined. Let $G = (P, E)$ be a graph in which vertex set P corresponds to permutations and edge set E corresponds to a "closeness" of permutation pairs. Evidently, some of the edges of E can be considered as defining possible element interchanges.

Thus, we can consider graph of $k-$*closeness* (from $k-$*interchange* viewpoint of view) as follows: $G^k = (P, E^k)$. By analogy, we get digraph $D^k = (P, O^k)$ where the following conditions hold:

(1) $p_i, p_j \in P$, $p_i, p_j \in E$, and p_i and p_j are "close";

(2) $(p_i, p_j) \in O^k$ if and only if p_j can be obtained from p_i by $k-$interchange.

We can examine a multi-level description of the above-mentioned operation set O^k ($O^k \subseteq E^k$): k corresponds to possible $k-$*element* interchange. Thus we arrive at the following possibilities:

$n-$*exchange* algorithm on a graph $G = (P, E)$: every permutation is adjacent to every other one,

$(n-k)-$*exchange* algorithm on digraph $D^{n-k} = (P, O^{n-k})$ for $1 \leq k \leq n-k$.

...

The $2-$*interchange* (adjacent interchange) algorithm uses the digraph $D^2 = (P, O^2)$. A related generalized description for the traveling salesman problem has been described by Reinelt in [11]. The following result is obvious:

$$O^k \subseteq O^{k+1}, \ \forall k = 2, ..., n-1.$$

Two properties of interest, which may or may not hold for given problem, are:

Property 1. $\forall p \in P$ exists a path in D which leads to an optimal permutation.

Property 2. $\forall p \in P$ exists a path in D for which the following hold:
(1) the path leads to an optimal point;
(2) the length of the path which corresponds to the number of interchanges is polynomial in n (steps of interchanges).

It is evident that Property 2 implies Property 1.

Now it is reasonable to consider the following observations:

1. The structure of the digraph D^k (i.e., its connectivity) for a certain kind of problem defines its complexity, e.g., the existence and length (polynomial in n or not) of the shortest path from a point to the optimal.

2. Not all optimization problems on permutations have connected digraph D^k.

3. Known combinatorial problems for which polynomial $2-$*interchange* algorithms exist have connected digraph D^2 with very "good" structure (e.g., tree).

4. For "hard" combinatorial problems the digraph D^k is unconnected at small levels of k. In other words, only the use of $k-$*interchange algorithm* for higher k, perhaps even n will guarantee reaching the optimum.

5. A digraph D^k may correspond to a problem with more than one path to an optimal point(s).

3. Neighborhoods and Operational Digraph

In section 2, neighborhood $A_k(s)$ for element $s \in S$ was defined. Now we examine a function $f(x)$ where $x = (x_1, ..., x_i, ..., x_n)$ is a permutation. It is assumed that $f(x)$ is integer-valued. We define specific types of neighborhoods as follows:

Definition.

Let $V^k(s)$ be a $k-$*interchange* neighborhood of point s defined by $x \in V^k(s)$ if and only if $x \neq s$ and x can be obtained from s via a single $k-$*interchange*.

Let $V^{k<}(s) \subseteq V^k(s)$ be that subset of $V^k(s)$ such that $f(x) < f(s)$.

Let $V^{k<=}(s) \subseteq V^k(s)$ be that subset of $V^k(s)$ such that $f(x) \leq f(s)$.

$D^{k<}$ and $D^{k<=}$ are digraphs which correspond to $k-$*interchange* algorithms on the basis of *improvement* of $f(x)$ and *improvement* or *equivalence* of $f(x)$, respectively. Note in the case of *equivalence* each *equivalence*-edge in $D^{k<=}$ will correspond to two arcs with opposite directions.

As a result of the definition above we obtain the following:

$$O^{k<} \subseteq O^{k<=}, \; V^{k-1} \subseteq V^k, \; V^{k<} \subseteq V^{k<=}.$$

Now let us examine numerical examples based upon the function shown in Table 1. Fig. 1. shows the digraph $D^{2<} = (P, O^k)$ where $(p_i, p_j) \in O^k$ if and only if $p_j \in V^{2<}(p_i)$, i.e., p_j can be obtained from p_i by adjacent interchange, and $f(p_j) < f(p_i)$. Note that the graph is not connected, and in fact the optimal point $(1, 2, 3, 4)$

is not connected to and can be reached by 2−*interchange* from only four other points, mainly $(2,1,4,3)$, $(1,2,4,3)$, $(1,3,2,4)$, and $(2,1,3,4)$.

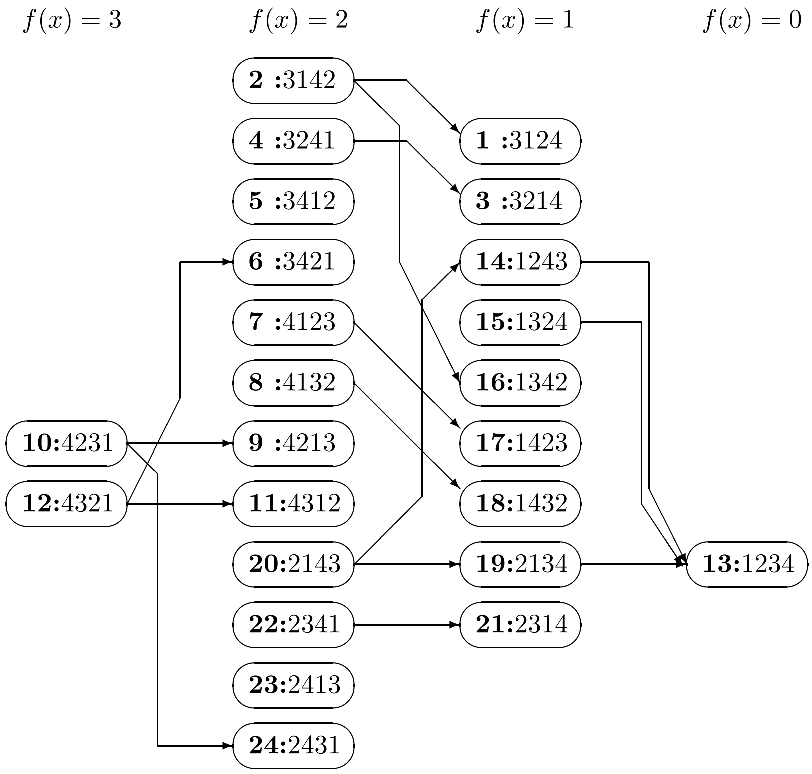

Fig. 1. Illustration for digraph $D^{2<}$

Fig. 2 depicts $D^{2<=}$. In this case, there exists a path from each permutation $\forall x \in X$ to the optimum point. Fig. 3 demonstrates a simple procedure for finding the 3−*neighborhood* of the permutation $(1,2,3,4)$. Every permutation of every contiguous set of length 3 in $(1,2,3,4)$ is listed, and duplicates are crossed out. Note that the digraph $D^{3<}$ (3−*interchange* algorithm) includes a path from every point to the optimal point.

Furthermore, the graphical structure of our problem can be analyzed. Without loss of generality we specify $(1,...,n)$ as an optimal permutation, and, in a similar fashion to section 2, define for $2 \leq k \leq n$ the functions $g_k(s) : P \to R$ to be the minimal number of neighborhood search steps needed to transform a permutation s into $1,...n$. The graph $D^{k<}$ is defined as before. Let $L = \max\{g_k(s)|s \in P\}$. $L+1$ represents the number of levels in $D^{k<}$ where level i is defined to be the set $\{s|g_k = i\}$. It is clear from the above definition that $D^{k<}$ is a connected digraph, and that there is a directed path from any permutation to $(1,...,n)$.

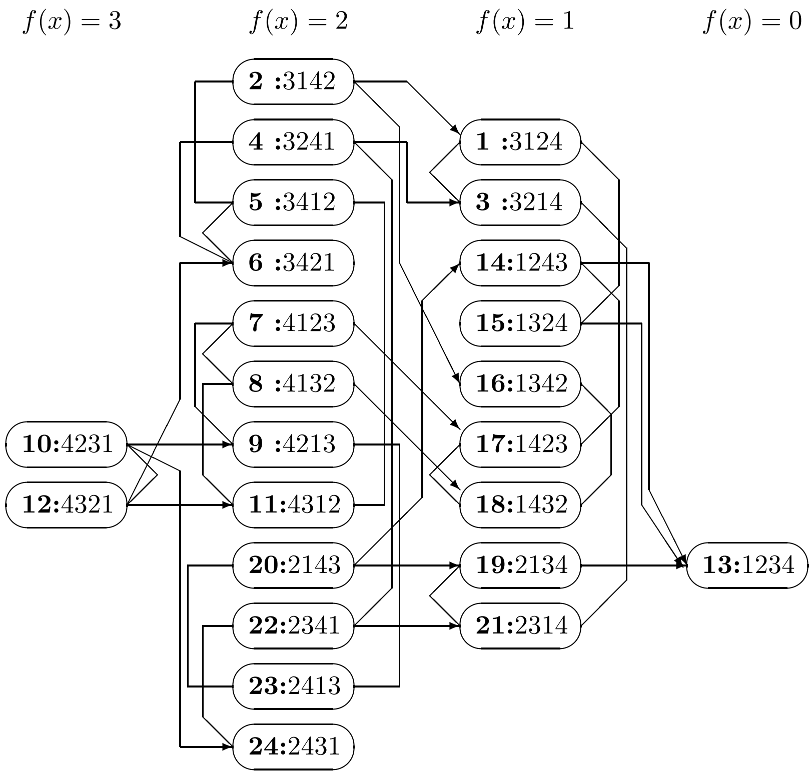

Fig. 2. Illustration for digraph $D^{2<=}$

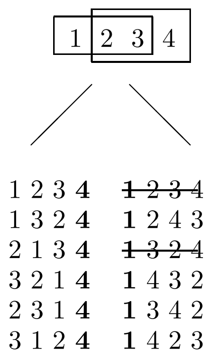

Fig. 3. Illustration for 3-neighborhood

Now let $f : P \rightarrow R$ represents a function to be minimized, and without loss of generality let $s = (1, ...n)$ be a permutation which minimizes f. Define $V^{k<}$ and $V^{k<=}$ based upon the function f. If t is a local optimum, it is follows that $|V^{k<}(t)| = 0$.

However, $V^{k<=}(t)$ may be non-empty in the case of ties, and this applies to $V^{k<=}(s)$ as well.

4. Algorithm System

4.1. Implementation Issues

Now let us consider two basic implementation issues . Several observations are in order:

1. If for every non-optimal x, $|V^{k<}| > 0$ then there exists a path in $D^{k<}$ from every x to an optimal point.

2. A necessary and sufficient condition for there to be a directed path in $D^{k<=}$ from every non-optimal permutation is the following:

For all non-optimal x either $|V^{k<} > 0|$ or there exists a sequence $(x = y_1, ..., y_h)$ such that for $1 \leq j \leq h-1$ the relationship $y_{j=1} \in V^{k<=}(y_j)$ and also $|V^{k<}| \neq 0$.

Note that the number of levels in the graph D^2 is $n(n-1)/2+1$ and the number of levels is less for D^k with $k > 2$.

Now using (i) and (ii) we get:

3. The $k-$*interchange* algorithm yields an optimal in polynomial time if

(a) $\forall x \ \ |V^{k<}(x)| > 0$ and

(b) $|V^{k<}(x)|$ is polynomial in n.

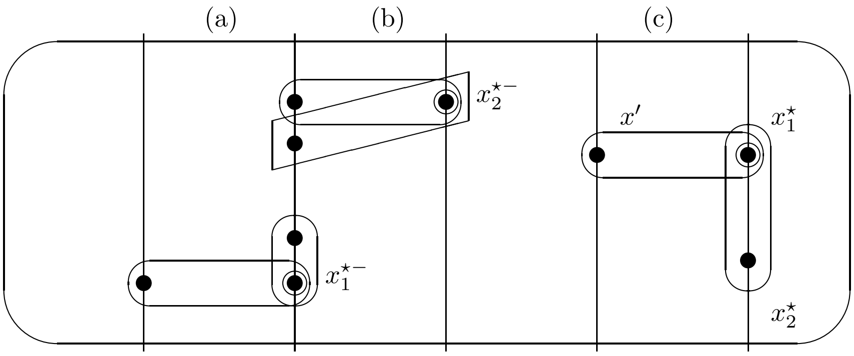

Fig. 4. Illustration for local and global optimum

There are many approaches to using $k-$*interchange*. The choice of the initial point, and, where choice exists, the choice of next point, are crucial parts of such algorithm. In line with many modern optimization approaches, probabilistic methods may be in order. Another key issue is identification of the optimum when it is found. Such identification depends upon the nature of the function being optimized. It may be useful also to start with small k (say 2) and only increase k to $k+1$ when a local (but not global) optimum is reached with $k-$*interchange*. After an improvement, the algorithm could return to using $k=2$. Fig. 4 illustrates possible situations with local and global optimum. Fig. 5 and 6 depict paths (strategies) to a global optimum for $D^{k<}$ and $D^{k<=}$.

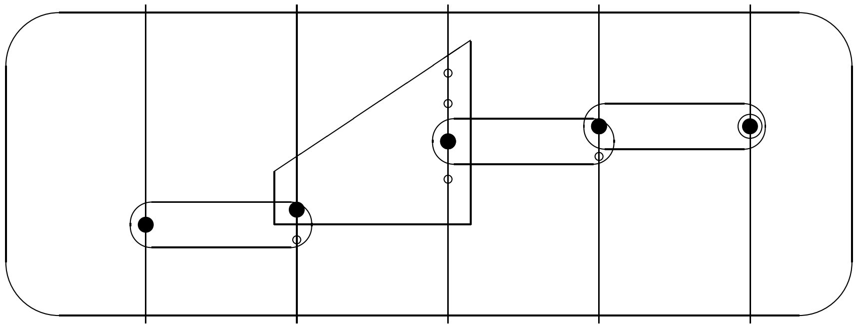

Fig. 5. Illustration for path $l(x)$ $D^{k<}$

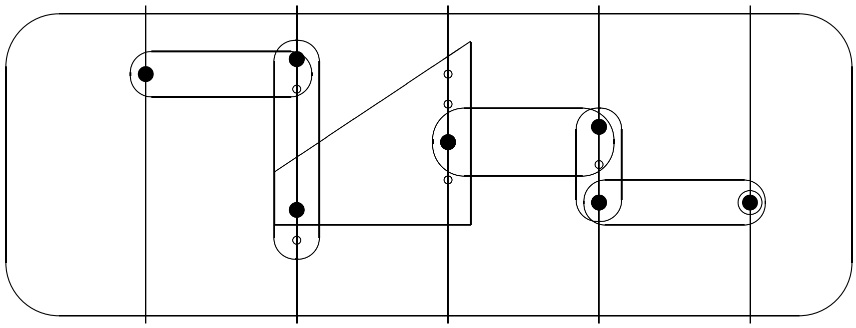

Fig. 6. Illustration for path $l(x)$ $D^{k<=}$

4.2. Solving Trajectory

First, the following three kinds of strategy steps exist: (a) *forward* as an improvement; (b) *aside*; and (c) *backward*. Thus we can consider three one-line trajectories as follows:

(1) kind **F** as *forward* steps;
(2) kind **FA** as *forward* and *aside* steps; and
(3) kind **FAB** as *forward*, *aside*, and *backward* steps.

In addition, it is reasonable to consider multi-line trajectories which consist of several one-line trajectories:

(1) kind **nF** as several trajectories of kind **F**;
(2) kind **nFA** as several trajectories of kinds **FA**;
(3) kind **nFAB** as several trajectories of kinds **FAB**.

4.3. Space of Algorithm Control

It is reasonable to study properties of an individual problem. In this case, the following is crucial:

(a) choice of an initial point (or a set of initial points);

(b) for a current point x selection of a next path step (i.e., a point y in neighborhood of x) because selected y has to lead to an optimal point;

(c) design a composite strategy or trajectory (composite path);

(d) on-line analysis of the solved problem and change (adaptation) of solving strategy (trajectory) by the following ways: (i) examination of new initial point(s); (ii) change of strategy types, (iii) change of algorithm types (decreasing or increasing k).

An initial information which is a basis to the problem analysis consists in types of points (permutations) and their neighborhoods from viewpoint of a quality of neighbor elements: (1) basic cases: (i) improvement (improvement is possible by moving to a neighbor element), (ii) equivalence, (iii) optimum; and (2) composite cases.

4.4. Structure of Algorithm System

We consider the following three-level structure of the algorithm system [9]:

1. Control unit (planning, adaptation): (a) selection of algorithms; (b) selection / design of composite solving strategy.

2. Level of execution: (a) analysis of an individual ordering problem; (b) executing some steps of solving process; and (c) analysis of obtained results.

3. Level of bases / repositories: (a) problems and examples; (b) base of $k-$exchange algorithms $k = 1, ..., n$; and (c) base of solving strategies.

The structure is oriented to concepts: (i) problem (analysis, approximation), (ii) model (selection, prediction), (iii) algorithm (selection), and (iv) solving strategy (selection, design).

5. Conclusion

Our digraph description of $k-$*interchange* approach is a good fundamental to analyze types of optimization ordering problems. The kinds of the proposed operational digraphs correspond to a problem property that is an analogue of convexity in continuous optimization. The considered adaptive algorithm system is close to optimization method macro-structures which are applied in global continuous optimization. Some possible future topics for investigation are the following:

1. Studies of well-known combinatorial problems including scheduling problems on the basis of the proposed approach.

2. Design of a special solving environment and execution of computing experiments for some well-known ordering problems.

3. Development of probabilistic analysis methods upon the approach.

4. Examination of sensitivity and/or stability for discrete problems on the basis of our approach. Problems of graph stability ([5], [12]) are a fundamental for this initiative.

5. Logical functions: Our description of scheduling problems leads to a special class of multi-valued logic functions when arguments (n-size vector) correspond to permutations and an ordinal scale is used for the objective function. Thus, a scheduling problem can be reformulated as optimization of a certain logical function. The study of properties for the logical functions is of interest (e.g., monotonicity).

6. Examining the possibility of effective algorithm design for problems where the probability of the existence of paths from each point to an optimal point is high, although not 100 per cent.

7. Investigation of multicriteria ordering problems.

8. Usage of artificial intelligence approaches (e.g., anytime algorithms [16]) for the problem analysis and monitoring the solving process.

6. Acknowledgments

The author acknowledges Jeffrey B. Sidney (University of Ottawa, Canada) who is the initiator of the research direction [14]. The author thanks the anonymous referee for careful consideration of the paper and useful comments.

References

[1] Adolphson D. and Hu T.C., Optimal linear ordering, *SIAM J. Appl. Math.*, **25**, 1973, 403-423.

[2] Borie R.B., Generation of polynomial-time algorithms for some optimization problems on tree-decomposable graphs. *Algorithmica*, **14**, 2, 1995, 123-137.

[3] Conway R.W., Maxwell W.L., and Miller L.W., *Theory of Scheduling*, Addison-Wesley, Readings, Mass., 1967.

[4] Elmaghraby S.E., A graph theoretic interpretation of the sufficiency condition for the contiguous binary-switching (CBS)-rule, *Naval Research Logistics Quarterly*, **18**, 3, 1971, 339-344.

[5] Harary F., Norman R.Z., and Carthwright D., *Structural Models: An Introduction to the Theory of Directed Graphs*, Wiley & Sons, New York, 1965.

[6] Hardy G.H., Littlewood J.E., and Polya G., *Inequalities*, 2nd ed., Cambridge University Press, Cambridge, 1952.

[7] Johnson S.M., Optimal two- and three-stage production schedules with setup times, *Naval Research Logistic Quarterly*, **1**, 1954, 61-67.

[8] Levin M.Sh., Effective solution of certain problems of theory of schedulings of nets, *Cybernetics (translated from Kibernetika)*, **16**, 1-6, 1980, 148-153.

[9] Levin M.Sh., Algorithm systems for combinatoral optimization: hierarchical multistage framework, in: *Proc. of 13th Intl. Conf. on Systems Engineering, Las Vegas,* 1999, CS109-114.

[10] Monma C.L. and Sidney J.B., Sequencing with series-parallel precedence constraints. *Mathematics of Operations Research,* **4**, 1979, 215-224.

[11] Reinelt G., *The Traveling Salesman, LNCS,* Vol. 840, Springer-Verlag, Berlin, 1994.

[12] Roberts F.R., *Discrete Mathematical Models with Applications to Social, Biological and Environmental Problems*, Prentice Hall, NJ, 1976.

[13] Sidney J.B., Decomposition algorithm for single-machine sequencing with prece-

dence relations and deferral costs, *Operations Research*, **23**, 1975, 283-298.

[14] Sidney J.B. and Levin M.Sh., *K-Exchange Technique for Optimization Over Permutations*, Working Paper No. 98-04, Faculty of Business Administration, Univ. of Ottawa, 1998.

[15] Smith W.E., Various optimizers for single-stage production, *Nav. Res. Log. Quart.*, **3**, 1956, 59-66.

[16] Zilberstein S., Using anytime algorithms in intelligent systems, *AI magazine*, **17**, 3, 1996, 73-83.