MSC 49L20, 90C39 DOI: 10.14529/mmp180106

SOLVING A ROUTING PROBLEM WITH THE AID OF AN INDEPENDENT COMPUTATIONS SCHEME

A.G. Chentsov[1,2], *A.M. Grigoryev*[1], *A.A. Chentsov*[1]
[1]Krasovskii Institute of Mathematics and Mechanics UrB RAS, Ekaterinburg, Russian Federation
[2]Ural Federal University, Ekaterinburg, Russian Federation
E-mail: chentsov@imm.uran.ru, ag@uran.ru, a_a_chentsov@mail.ru

This paper is devoted to the issues in development and implementation of parallel algorithms for solving practical problems. We consider a routing problem with constraints and complicated cost functions. The visited objects are assumed to be clusters, or megalopolises (nonempty finite sets), and the visit to each one entails certain tasks, which we call interior jobs. The order of visits is subject to precedence constraints. The costs of movements depend on the set of pending tasks (not yet complete at the time of the movement), which is also referred to as "sequence dependence", "position dependence", and "state dependence". Such dependence arises, in particular, in routing problems concerning emergencies at nuclear power plants, similar to the Chernobyl and Fukushima Daiichi incidents. For example, one could consider a disaster recovery problem concerned with sequential dismantlement of radiation sources; in this case, the crew conducting the dismantlement is exposed to the radiation from the sources that have not yet been dealt with. Hence the dependence on pending tasks in the cost functions that measure the crew's radiation exposure. The latter dependence reflects the "shutdown" operations for the corresponding radiation sources. This paper sets forth an approach to a parallel solution for this problem, which was implemented and run on the URAN supercomputer. The results of the computational experiment are presented.

Keywords: dynamic programming; route; sequencing; precedence constraints; parallel computation.

Introduction

Many real-life problems feature the issues connected with routing (transportation, routing the tool of CNC plate cutting machines, minimization of staff exposure to radiation during operations in radioactively contaminated environment). This paper is devoted to a study of an important class of discrete optimization problems, namely, the problems concerning multiple movements under constraints. The prototype of the considered problem is the well-known intractable travelling salesman problem (TSP), see [1, Ch. 3]; to name just a few works on TSP, [2–9]. However, in applications, one often has to satisfy certain additional constraints. In particular, constraints arise in the problem of minimizing the exposure of nuclear power plant staff conducting operations related to dismantling radiation sources. A typical characteristic of the latter problem is dependence on the set of pending tasks: only the sources that are not yet dismantled at the time of each operations do "radiate". There are also other applications [10–12]. Another peculiarity, which arises, for example, in the known problem of dismantling a decommissioned nuclear power generation unit, consists in precedence constraints on the sequence of operations; this is formalized by specifying ordered pairs (OP) of operations where the first component of the pair must be conducted before the second one. Finally, in contrast with "ordinary"

TSP, in applications, the "cities" visited may possess some internal structure and are thus rendered not as "cities" but as clusters of cities, or megalopolises, which introduces a certain additional variation into the movements — the problem develops a two-layer hierarchy: layer one is the sequencing of megalopolises through a permutation of their indices — the route, and layer two is the choice of a "track" along the given route, i.e., the exact versions of movements through the megalopolises. The arguments leading to a two-layer optimization problem are stated in [13], and the applications of methods developed in [13] for problems of nuclear power generation are discussed in [14, 15].

Among the applications of theoretical constructions of [13], note the problems of machine engineering concerned with CNC cutting machines; see [16–20].

In this connection, note that in routing problems of appreciable dimension one could hardly avoid the use of heuristics, particularly in nonmetric statements (with nonmetric travel cost functions), which is the case for the problem considered in this paper. However, then, there appears the problem of testing the heuristics on problem instances of the corresponding class. This can be conducted (see [21]) on model problem instances of smaller dimension that nevertheless possess all features of the problem statement: constraints, cost functions, etc. In the mentioned model problems, we intend to use the DP apparatus to find the value (extremum). Then, by running the heuristics on the same problems, after a representative body of statistics is accumulated, one could analyze how close the heuristic results are to the mentioned extremum. After a heuristic is determined to be "sufficiently good" in the mentioned sense, we can later apply it to a "big" problem that can no longer be solved through DP. Thus, DP can be used to create a kind of proving ground for heuristics; still, it is possible that the success of a heuristic on model problem instances will not transfer to instances of greater dimension, whence a desire to expand the usability frame of DP as a testing instrument onto model problem instances of greater dimension, which could not be solved on a personal computer. It is deemed worthwhile to use supercomputers to consider the model problem instances the dimensions of which are closer to their real-life prototypes while retaining all complications and additional constraints typical in practice (expand the proving ground boundaries). To this end, we use a parallel version of dynamic programming aimed at supercomputers; in this paper, we describe the constructions aligned with the problem of minimizing nuclear power plant staff exposure to radiation during operations connected with dismantling radiation sources.

1. Problem Statement and Discussion

Let us start with general definitions, notions, and notations. The symbol $\triangleq$ denotes equality by definition. A set, all elements of which are themselves sets is called a family. The real line is denoted by $\mathbb{R}$; $\mathbb{R}_+ \triangleq \{\xi \in \mathbb{R} \,|\, 0 \leqslant \xi\}$, $\mathbb{N} \triangleq \{1; 2; \ldots\}$ and $\mathbb{N}_0 \triangleq \mathbb{N} \cup \{0\} = \{0; 1; 2; \ldots\}$, $\mathbb{N} \subset \mathbb{N}_0 \subset \mathbb{R}$. For $p \in \mathbb{N}_0$ and $q \in \mathbb{N}_0$, assume $\overline{p, q} \triangleq \{t \in \mathbb{N}_0 \,|\, (p \leqslant t) \,\&\, (t \leqslant q)\}$. For every ordered pair (OP) $z = (a, b)$ of arbitrary objects a and b, denote by $\mathrm{pr}_1(z)$ and $\mathrm{pr}_2(z)$, respectively, its first and second elements: $\mathrm{pr}_1(z) = a$, $\mathrm{pr}_2(z) = b$. For an object x, the corresponding singleton set is denoted by $\{x\}$. For arbitrary objects a, b, and c, by convention [22, p. 17], assume $(a, b, c) \triangleq ((a, b), c)$. For three arbitrary sets A, B, and C, following [22, p. 17], we assume $A \times B \times C \triangleq (A \times B) \times C$, thus, for $\mu \in A \times B$ and $\nu \in C$, we have $(\mu, \nu) \in A \times B \times C$. By $\mathcal{R}_+[S]$ we denote the set of all functions from a nonempty set S into $\mathbb{R}_+$. For a set H, denote by $\mathcal{P}(H)$ the set of all subsets of H, $\mathcal{P}'(H) \triangleq \mathcal{P}(H) \setminus \{\varnothing\}$; denote by $\mathrm{Fin}(H)$ the family of all (nonempty) finite sets of $\mathcal{P}'(H)$. If H is a finite set, then $\mathrm{Fin}(H) = \mathcal{P}'(H)$. To a nonempty finite set K, assign its cardinality $|K| \in \mathbb{N}$ and the (nonempty) set (bi)$[K]$ of all bijections [23, p. 87] of the set $\overline{1, |K|}$ onto K; in addition, $|\varnothing| \triangleq 0$. A permutation of a nonempty set T is [23, p. 87] a bijection of T onto itself; for every permutation α of T, there exists an inverse permutation α^{-1} of the set T: $\alpha(\alpha^{-1}(t)) = \alpha^{-1}(\alpha(t)) = t \ \ \forall t \in T$.

Fix a nonempty set X, a point $x^\circ \in X$, a number $N \in \mathbb{N}, N \geqslant 2$,(nonempty finite) sets – megalopolises $M_1 \in \mathrm{Fin}(X), \ldots, M_N \in \mathrm{Fin}(X)$, and relations $\mathbb{M}_1 \in \mathcal{P}'(M_1 \times M_1), \ldots, \mathbb{M}_N \in \mathcal{P}'(M_N \times M_N)$. Here and below, assume $(x^\circ \notin M_j \ \ \forall j \in \overline{1, N})\ \&\ (M_p \cap M_q = \varnothing \ \ \forall p \in \overline{1, N} \ \ \forall q \in \overline{1, N} \setminus \{p\})$. For each $j \in \overline{1, N}$, OPs from $\mathbb{M}_j$ determine the feasible interior jobs connected with visiting M_j: the first element of such an OP is the entry point, and the second element is the exit point. We consider the processes of the following [24, (3.3)] form:

$$x^\circ \to (\mathrm{pr}_1(z_1) \in M_{\alpha(1)} \rightsquigarrow \mathrm{pr}_2(z_1) \in M_{\alpha(1)}) \to \ldots \to (\mathrm{pr}_1(z_N) \in M_{\alpha(N)} \rightsquigarrow \mathrm{pr}_2(z_N) \in M_{\alpha(N)}), \quad (1.1)$$

where $z_1 \in \mathbb{M}_{\alpha(1)}, \ldots, z_N \in \mathbb{M}_{\alpha(N)}$, and α is a permutation of indices of $\overline{1, N}$, hereinafter referred to as route. Assume that in (1.1) the straight arrows denote the exterior movements and the wavy arrows denote the motions connected with (interior) jobs. The objects of our choice are $\alpha, z_1, \ldots, z_N$. Let $\mathbf{M}_j \triangleq \{\mathrm{pr}_2(z) : \ z \in \mathbb{M}_j\} \in \mathrm{Fin}(M_j) \ \ \forall j \in \overline{1, N}$; $\mathbb{X} \triangleq \{x^\circ\} \cup \left(\bigcup_{i=1}^{N} M_i\right) \in \mathrm{Fin}(X)$, $\mathbf{X} \triangleq \{x^\circ\} \cup \left(\bigcup_{i=1}^{N} \mathbf{M}_i\right) \in \mathrm{Fin}(\mathbb{X})$. In connection with (1.1), note that both exterior movements and interior jobs conducted as the megalopolises are visited are measured through the given functions, and the results of these measurements are aggregated additively, which is the natural way for many applications. Here and below, assume $\mathbb{P} \triangleq (\mathrm{bi})[\overline{1, N}]$, and also assume that each specific $\alpha \in \mathbb{P}$ (see (1.1)) must satisfy the precedence constraints, which are defined by means of the set $\mathbf{K} \in \mathcal{P}(\overline{1, N} \times \overline{1, N})$; OPs from $\mathbf{K}$ will be called address pairs; the feasibility of $\alpha \in \mathbb{P}$ can be reduced to the following requirement: for $z = (i, j) \in \mathbf{K}$, the set M_i must be visited before M_j. Here and below, assume [24, (3.11)] the following:

$$\forall \mathbf{K}_0 \in \mathcal{P}'(\mathbf{K}) \exists z_0 \in \mathbf{K}_0 : \ \mathrm{pr}_1(z_0) \neq \mathrm{pr}_2(z) \ \ \forall z \in \mathbf{K}_0. \quad (1.2)$$

Then, the set $\mathbf{A}$ of all feasible (with respect to precedence) routes from $\mathbb{P}$ has the following form [24, (3.12)]: $\mathbf{A} \triangleq \{\alpha \in \mathbb{P} | \alpha^{-1}(\mathrm{pr}_1(z)) < \alpha^{-1}(\mathrm{pr}_2(z)) \ \ \forall z \in \mathbf{K}\} \in \mathcal{P}'(\mathbb{P})$. In connection with (1.2), note that $\mathrm{pr}_1(z) \neq \mathrm{pr}_2(z) \ \ \forall z \in \mathbf{K}$. As evident from (1.1), the choice of some route $\alpha \in \mathbf{A}$ does not yet determine the whole process, it must be supplemented with some track $z_1, \ldots, z_N$. A track has to agree with the given route, and the solution is constructed in the form of OP consisting of a route and a track, where the route must be an element of $\mathbf{A}$. It is useful to bear in mind that such interpretation allows one to consider not only (1.1) but also the partial processes connected with visiting megalopolises $M_k, k \in K$, where $K \subset \overline{1, N}$. Such interpretation is also useful in connection with the dynamic programming (DP); it was laid out in [21,24,25]. In this paper, we omit its details and devote most of our attention to the computational concerns. We limit ourselves to a description of "complete" routes and "tracks" (trajectories), similar to (1.1). Thus, if $\alpha \in \mathbb{P}$, denote by $\mathcal{Z}_\alpha$ the set of all tuples $(z_i)_{i \in \overline{0, N}} : \overline{0, N} \to \mathbb{X} \times \mathbb{X}$ such that $(z_0 =$

$(x^0, x^0)) \,\&\, (\, z_t \in \mathbb{M}_{\alpha(t)} \quad \forall t \in \overline{1,N})$; $\mathcal{Z}_\alpha$ is a nonempty finite set. Feasible solutions will be represented by the OPs $(\alpha, (z_i)_{i \in \overline{0,N}})$, where $\alpha \in \mathbf{A}$ and $(z_i)_{i \in \overline{0,N}} \in \mathcal{Z}_\alpha$; $\mathfrak{N} \triangleq \mathcal{P}'(\overline{1,N})$.

Consider the following cost functions (they are assumed to be known): $\mathbf{c} \in \mathcal{R}_+[\mathbb{X} \times \mathbb{X} \times \mathfrak{N}]$, $c_1 \in \mathcal{R}_+[\mathbb{X} \times \mathbb{X} \times \mathfrak{N}], \ldots, c_N \in \mathcal{R}_+[\mathbb{X} \times \mathbb{X} \times \mathfrak{N}]$, $f \in \mathcal{R}_+[\mathbb{X}]$. In terms of these functions, let us define the additive criterion: for $\alpha \in \mathbf{A}$ and $(z_i)_{i \in \overline{0,N}} \in \mathcal{Z}_\alpha$, set

$$\begin{aligned} \mathfrak{C}_\alpha[(z_i)_{i \in \overline{0,N}}] \triangleq \sum_{t=1}^{N} [\mathbf{c}(\mathrm{pr}_2(z_{t-1}), \mathrm{pr}_1(z_t), \{\alpha(j) : j \in \overline{t,N}\}) + \\ + c_{\alpha(t)}(z_t, \{\alpha(j) : j \in \overline{t,N}\})] + f(\mathrm{pr}_2(z_N)). \end{aligned} \tag{1.3}$$

Our principal problem is as follows: $\mathfrak{C}_\alpha[(z_i)_{i \in \overline{0,N}}] \to \min,\ \alpha \in \mathbf{A}, (z_i)_{i \in \overline{0,N}} \in \mathcal{Z}_\alpha$; for this problem, there exists a nonempty set of optimal feasible solutions and the problem's value — the following extremum:

$$V \triangleq \min_{\alpha \in \mathbf{A}} \ \min_{(z_i)_{i \in \overline{0,N}} \in \mathcal{Z}_\alpha} \mathfrak{C}_\alpha[(z_i)_{i \in \overline{0,N}}] \in \mathbb{R}_+. \tag{1.4}$$

Our goal is to find V (1.4) and some optimal feasible solution $(\alpha^0, (z_i^0)_{i \in \overline{0,N}})$, where $\alpha^0 \in \mathbf{A}$ and $(z_i^0)_{i \in \overline{0,N}} \in \mathcal{Z}_{\alpha^0}$; evidently, they satisfy $\mathfrak{C}_{\alpha^0}[(z_i^0)_{i \in \overline{0,N}}] = V$.

2. Dynamic Programming. Layers of Bellman Function

Our problem is solved by means of a variety of DP [21, 24, 25], which we specify in a brief form. We use the construction from [21, 24, 25], which is based on the layer structure of the Bellman function; the algorithm of their construction is sketched below. Let us consider the construction of state space layers by means of the crossing-out operator (for tasks in the task list) [13, Pt. 2]: $\mathbf{I} : \mathfrak{N} \to \mathfrak{N}$, specifically, for $K \in \mathfrak{N}$, set $\Xi(K) \triangleq \{z \in \mathbf{K} \,|\, (\mathrm{pr}_1(z) \in K) \,\&\, (\mathrm{pr}_2(z) \in K)\}$ and $\mathbf{I}(K) \triangleq K \setminus \{\mathrm{pr}_2(z) : \ z \in \Xi(K)\}$. Operator $\mathbf{I}(K)$ is used to construct the layers of the state space. To this end, consider a (nonempty) set $\mathbf{G} \triangleq \{K \in \mathfrak{N} \,|\, \forall z \in \mathbf{K}\ (\mathrm{pr}_1(z) \in K) \Rightarrow (\mathrm{pr}_2(z) \in K)\}$, clearly, $\overline{1,N} \in \mathbf{G}$; we call its elements feasible task lists. We sort the mentioned lists by their cardinality, $\mathbf{G}_s \triangleq \{K \in \mathbf{G} \,|\, s = |K|\}\ \forall s \in \overline{1,N}$. Then, the family $\{\mathbf{G}_j : \ j \in \overline{1,N}\}$ is a partition of $\mathbf{G}$. In addition, $\mathbf{G}_1 \triangleq \{\{t\} : \ t \in \overline{1,N} \setminus \mathbf{K}_1\}$, where $\mathbf{K}_1 \triangleq \{\mathrm{pr}_1(z) : z \in \mathbf{K}\}$. Also [21, 24, 25], $\mathbf{G}_{s-1} = \{K \setminus \{j\} : \ K \in \mathbf{G}_s,\ j \in \mathbf{I}(K)\}\ \forall s \in \overline{2,N}$. Thus we obtain a recurrence procedure: $\mathbf{G}_N = \{\overline{1,N}\}$, $\mathbf{G}_N \to \mathbf{G}_{N-1} \to \ldots \to \mathbf{G}_1$. Based on this procedure, we construct the state space layers, which we denote $D_0, D_1, \ldots, D_N$. Specifically, $D_0 = \{(x, \varnothing) : \ x \in \mathfrak{M}\}$, where $\mathfrak{M}$ is by definition the union of all the sets $\mathbf{M}_i,\ i \in \overline{1,N} \setminus \mathbf{K}_1$. Next, $D_N \triangleq \{(x^0, \overline{1,N})\}$ (the singleton that contains the OP $(x^0, \overline{1,N})$). If $s \in \overline{1,N-1}$ and $K \in \mathbf{G}_s$, we have [25] the following sequence of definitions:

$$\begin{gathered} J_s(K) \triangleq \{t \in \overline{1,N} \setminus K \,|\, \{t\} \cup K \in \mathbf{G}_{s+1}\}, \mathcal{M}_s[K] \triangleq \bigcup_{j \in J_s(K)} \mathbf{M}_j, \mathbb{D}_s[K] \triangleq \\ \triangleq \{(x, K) : x \in \mathcal{M}_s[K]\} \in \mathcal{P}'(\mathbf{X} \times \mathbf{G}_s). \end{gathered}$$

Thus, for $s \in \overline{1,N-1}$, the layer D_s is defined by the rule

$$D_s \triangleq \bigcup_{K \in \mathcal{G}_s} \mathbb{D}_s[K]. \tag{2.1}$$

It is easy to see that $D_0 \neq \varnothing, D_1 \neq \varnothing, \dots, D_N \neq \varnothing$. If $s \in \overline{1,N}$, $(x,K) \in D_s, j \in \mathbf{I}(K)$, and $z \in \mathbb{M}_j$, then $(\mathrm{pr}_2(z), K \setminus \{j\}) \in D_{s-1}$.

Recurrence procedure for construction of the layers. Consider a system of funcions $v_0 \in \mathcal{R}_+[D_0], v_1 \in \mathcal{R}_+[D_1], \dots, v_N \in \mathcal{R}_+[D_N]$.

First, define $v_0 \in \mathcal{R}_+[D_0]$ by the condition $v_0(x, \varnothing) \triangleq f(x) \quad \forall x \in \mathfrak{M}$. Further constructions implement the following recurrence scheme: for $s \in \overline{1,N}$, if the function $v_{s-1} \in \mathcal{R}_+[D_{s-1}]$ is already constructed, then $v_s \in \mathcal{R}_+[D_s]$ can be determined by the rule

$$v_s(x,K) \triangleq \min_{j \in \mathbf{I}(K)} \min_{z \in \mathbb{M}_j} [\mathbf{c}(x, \mathrm{pr}_1(z), K) + c_j(z,K) + v_{s-1}(\mathrm{pr}_2(z), K \setminus \{j\})] \ \forall (x,K) \in D_s. \quad (2.2)$$

From the general constructions of [21, 24, 25], there follows the equality

$$V = v_N(x^0, \overline{1,N}), \quad (2.3)$$

which lets us determine the global extremum (see (1.4)). The procedure $v_0 \to v_1 \to \ldots \to V$, which is completely defined by virtue of (2.2), can be regarded as an algorithm for determining V, during which, only one layer of the Bellman function is retained in the computer's RAM; for details, refer to [26]. Similar idea was proposed in [27]. Construction of optimal solution corresponds to [21, 24, §4, §7].

3. Independent Computations of Bellman Function Layers

Further exposition follows a version of general construction [24, 28, 29] connected with conducting independent computations. Recall that (2.2) and (2.3) state the "final" formula for determining V through v_{N-1}. Thus, further constructions require one to determine v_{N-1}, which is conducted by the procedure below (it is clear how to compute V if v_{N-1} is known). The function v_{N-1} is defined on the set D_{N-1}, which is defined through $\mathbf{G}_{N-1}$. Here and below, assume $N \geqslant 3$; we have

$$\mathbf{G}_{N-1} = \{\overline{1,N} \setminus \{j\} : j \in \mathbf{I}(\overline{1,N})\}; \quad (3.1)$$

since $\mathbf{G}_N = \{\overline{1,N}\}$, it is possible to determine D_{N-1}. Assume all the layer-functions will be constructed on $\mathbf{n}$ nodes, where $\mathbf{n} \triangleq |\mathbf{G}_{N-1}| = |\mathbf{I}(\overline{1,N})| \in \mathbb{N}$. To facilitate these constructions, each layer $D_0, D_1, \dots, D_N$ will be interpreted as a union of $\mathbf{n}$ subsets; each such subset will be distributed to a specific node, which would construct its fragment of the layer. This construction will be conducted through auxiliary discrete dynamic systems (DDS), the trajectories of which will be determined through systems of inclusions. Thus, assume $K \in \mathbf{G}_{N-1}$, and let $\mathbb{T}[K]$ be the set of all the tuples $(K_t)_{t \in \overline{0,N-2}} : \overline{0,N-2} \to \mathbf{G}$ such that $(K_0 \triangleq K)$ & $(\forall \tau \in \overline{1,N-2} \ \exists s \in \mathbf{I}(K_{\tau-1}) : K_\tau = K_{\tau-1} \setminus \{s\})$. Given tuples – trajectories of the system — are not uniquely determined for each given K; thus, for each "time" $t \in \overline{0,N-2}$, we have a reachability set (RS) $\widetilde{\mathbb{T}}[K;t]$, defined to be the set of all the lists K_t such that $(K_\tau)_{\tau \in \overline{0,N-2}}$ are the trajectories from $\mathbb{T}[K]$:

$$\widetilde{\mathbb{T}}[K;t] \triangleq \{K_t : (K_i)_{i \in \overline{0,N-2}} \in \mathbb{T}[K]\}, \quad (3.2)$$

in addition [28, 29], $\widetilde{\mathbb{T}}[K;t] \in \mathcal{P}'(\mathbf{G}_{N-(t+1)})$. As it is easy to see, for every $K \in \mathbf{G}_{N-1}$, $(\widetilde{\mathbb{T}}[K;0] = \{K\})$ & $(\widetilde{\mathbb{T}}[K;N-2] \subset \mathbf{G}_1)$. Now, the families of feasible lists of fixed cardinality can be determined through these RSs, specifically,

$$\mathbf{G}_{N-(t+1)} = \bigcup_{K\in\mathbf{G}_{N-1}} \widetilde{\mathbb{T}}[K;t] \quad \forall t \in \overline{0, N-2}. \tag{3.3}$$

Let us also note that the mentioned RSs could [29, Proposition 16] be determined in a recurrent fashion for a fixed condition: if $K \in \mathbf{G}_{N-1}$ and $t \in \overline{0, N-3}$, then $\widetilde{\mathbb{T}}[K;t+1] = \{P \setminus \{h\} : P \in \widetilde{\mathbb{T}}[K;t],\ h \in \mathbf{I}(P)\}$. Through RSs, we construct the "individual" (to each computation node–its own) state space layers: for $K \in \mathbf{G}_{N-1}$ and $s \in \overline{1, N-1}$, in view of (2.1), set

$$\mathcal{D}_s[K] \overset{\triangle}{=} \bigcup_{P\in\widetilde{\mathbb{T}}[K;N-(s+1)]} \mathbb{D}_s[P] \in \mathcal{P}'(D_s). \tag{3.4}$$

These layers (3.4) possess the property $(\mathrm{pr}_2(z), Q \setminus \{s\}) \in \mathcal{D}_l[K] \ \ \forall l \in \overline{1, N-2} \ \ \forall (x, Q) \in \mathcal{D}_{l+1}[K] \ \ \forall s \in \mathbf{I}(Q) \ \ \forall z \in \mathbb{M}_s$. In view of (3.4), for $K \in \mathbf{G}_{N-1}$ and $s \in \overline{1, N-1}$, assume

$$\mathcal{W}_s[K] \overset{\triangle}{=} (v_s(x, P))_{(x,P)\in\mathcal{D}_s[K]} = (v(x, P))_{(x,P)\in\mathcal{D}_s[K]} \in \mathcal{R}_+[\mathcal{D}_s[K]]. \tag{3.5}$$

In particular, for $K \in \mathbf{G}_{N-1}$, we can determine $\mathcal{D}_1[K] \in \mathcal{P}'(D_1)$ and $\mathcal{W}_1[K] \in \mathcal{R}_+[\mathcal{D}_1[K]]$. To explicitly describe $\mathcal{D}_1[K]$ for $K \in \mathbf{G}_{N-1}$, let us first note, in view of (3.4), that $\widetilde{\mathbb{T}}[K;N-2] \subset \mathbf{G}_1$, whence $\forall P \in \widetilde{\mathbb{T}}[K;N-2] \ \ \exists t \in \overline{1, N} \setminus \mathbf{K}_1 : P = \{t\}$. Consequently [24, 10.4], we have the property $\forall K \in \mathbf{G}_{N-1} \ \ \forall (x, P) \in \mathcal{D}_1[K] \ \ \exists t \in \overline{1, N} \setminus \mathbf{K}_1 : P = \{t\}$. In view of (2.2), and (3.5), we find that, for $K \in \mathbf{G}_1$ and $(x, P) \in \mathcal{D}_1[K]$,

$$\mathcal{W}_1[K](x, P) = v_1(x, P) = \min_{j\in\mathbf{I}(P)} \min_{z\in\mathbb{M}_j} [\mathbf{c}(x, \mathrm{pr}_1(z), P) + c_j(z, P) + v_0(\mathrm{pr}_2(z), P \setminus \{j\})],$$

where $P = \{t\}$ for a certain $t \in \overline{1, N} \setminus \mathbf{K}_1$ and, consequently [24, Remark 3.2], $\mathbf{I}(P) = \mathbf{I}(\{t\}) = \{t\} = P$; then,

$$\mathcal{W}_1[K](x, P) = \min_{j\in P} \min_{z\in\mathbb{M}_j} [\mathbf{c}(x, \mathrm{pr}_1(z), P) + c_j(z, P) + f(\mathrm{pr}_2(z))] \tag{3.6}$$

(in (3.6), note that, evidently, $\mathbf{M}_t \subset \mathfrak{M}$ and $\mathrm{pr}_2(z) \in \mathbf{M}_t$ since $j \in P$ implies $j = t$). Since in (3.6) P is a singleton, this formula may be simplified: following [30, (5.7)], consider, for $K \in \mathbf{G}_{N-1}$ (see [29, (62), Proposition 7], a nonempty set $\mathbb{M}_0[K] \overset{\triangle}{=} \{h \in \overline{1, N} \setminus \mathbf{K}_1 | \{h\} \in \widetilde{\mathbb{T}}[K;N-2]\}$ for $K \in \mathbf{G}_{N-1}$. In view of (3.4), we obtain

$$\mathcal{D}_1[K] = \bigcup_{P\in\widetilde{\mathbb{T}}[K;N-2]} \mathbb{D}_1[P] = \bigcup_{P\in\widetilde{\mathbb{T}}[K;N-2]} \{(x, P) : x \in \mathcal{M}_1[P]\}, \tag{3.7}$$

where $\widetilde{\mathbb{T}}[K;N-2] = \{\{h\} : h \in \mathbb{M}_0[K]\}$ [27, Proposition 5.1]; thus, from (3.7), we have $\mathcal{D}_1[K] = \{(x, \{h\}) : h \in \mathbb{M}_0[K],\ x \in \mathcal{M}_1[\{h\}]\}$. Next, for $K \in \mathbf{G}_{N-1}$ and $h \in \mathbb{M}_0[K]$, we have $\{h\} \in \widetilde{\mathbb{T}}[K;N-2]$, where $h \in \overline{1, N} \setminus \mathbf{K}_1$; thus, $\{h\} \in \mathbf{G}_1$ and $J_1(\{h\}) = \{t \in \overline{1, N} \setminus \{h\} | \{t;h\} \in \mathbf{G}_2\}$,

$$\mathcal{M}_1[\{h\}] = \bigcup_{j\in J_1(\{h\})} \mathbf{M}_j. \tag{3.8}$$

In view of (3.6), for $K \in \mathbf{G}_{N-1}$, $h \in \mathbb{M}_0[K]$, and $x \in \mathcal{M}_1[\{h\}]$, we have the following property: $(x, \{h\}) \in \mathcal{D}_1[K]$ and

$$\mathcal{W}_1[K](x, \{h\}) = \min_{z \in \mathbb{M}_h}[\mathbf{c}(x, \mathrm{pr}_1(z), \{h\}) + c_h(z, \{h\}) + f(\mathrm{pr}_2(z))]. \tag{3.9}$$

Through (3.9), we can fully determine the function $\mathcal{W}_1[K]$, while, to use (3.9) we have to, knowing $K \in \mathbf{G}_{N-1}$ and $h \in \mathbb{M}_0[K]$, construct $J_1(\{h\})$, and then determine $\mathcal{M}_1[\{h\}]$ by means of (3.8), after which we can finally use formula (3.9) for $x \in \mathcal{M}_1[\{h\}]$. We determine all the functions $\mathcal{W}_1[K]$, $K \in \mathbf{G}_{N-1}$. Further constructions will concern one single computation node. A transformation of a function $\mathcal{W}_l[K]$ into $\mathcal{W}_{l+1}[K]$, where $K \in \mathbf{G}_{N-1}$ and $l \in \overline{1, N-2}$, is determined by a relation similar to (2.2) (indeed, by (3.5), for $(x, Q) \in \mathcal{D}_{l+1}[K]$, $s \in \mathbf{I}(Q)$, and $z \in \mathbb{M}_s$, the value $\mathcal{W}_l[K](\mathrm{pr}_2(z), Q \setminus \{s\}) \in \mathbb{R}_+$ is defined and can be used to compute $\mathcal{W}_{l+1}[K](x, Q)$). Thus (see (2.2) and (3.5)), for $K \in \mathbf{G}_{N-1}$, $l \in \overline{1, N-1}$, and $(x, Q) \in \mathcal{D}_{l+1}[K]$, we have [24, (10.17)]

$$\mathcal{W}_{l+1}[K](x, Q) = \min_{s \in \mathbf{I}(Q)} \min_{z \in \mathbb{M}_j}[\mathbf{c}(x, \mathrm{pr}_1(z), Q) + c_j(z, Q) + \mathcal{W}_l[K](\mathrm{pr}_2(z),\ Q \setminus \{s\})]. \tag{3.10}$$

Fix $\mathbb{K} \in \mathbf{G}_{N-1}$. For this set, we have the corresponding layers $\mathcal{D}_s[\mathbb{K}]$ (3.4), $s \in \overline{1, N-1}$, which form a nonempty set in the state space. For $s \in \overline{1, N-1}$, the function $\mathcal{W}_s[K] \in \mathcal{R}_+[\mathcal{D}_s[\mathbb{K}]]$ is defined. In particular, the function $\mathcal{W}_1[\mathbb{K}] \in \mathcal{R}_+[\mathcal{D}_1[\mathbb{K}]]$ is defined; to determine its values, (3.9) should be used in view of the representation of $\mathcal{D}_1[\mathbb{K}]$. Further construction of functions $\mathcal{W}_1[\mathbb{K}], \ldots, \mathcal{W}_{N-1}[\mathbb{K}]$ is conducted based on a recurrence procedure based on (3.10). Specifically, for $l \in \overline{1, N-2}$, the transformation of $\mathcal{W}_l[\mathbb{K}]$ into $\mathcal{W}_{l+1}[\mathbb{K}]$ is determined by the expression $\forall (x, Q) \in \mathcal{D}_{l+1}[\mathbb{K}]$:

$$\mathcal{W}_{l+1}[\mathbb{K}](x, Q) = \min_{s \in \mathbf{I}(Q)} \min_{z \in \mathbb{M}_j}[\mathbf{c}(x, \mathrm{pr}_1(z), Q) + c_j(z, Q) + \mathcal{W}_l[\mathbb{K}](\mathrm{pr}_2(z),\ Q \setminus \{s\})]. \tag{3.11}$$

Rule (3.11) is a specification of (3.10). Thus, (3.11) describes the transformation $\mathcal{W}_l[\mathbb{K}] \to \mathcal{W}_{l+1}[\mathbb{K}]$. Since the choice of l was arbitrary, we obtain the recurrence procedure

$$\mathcal{W}_1[\mathbb{K}] \to \mathcal{W}_2[\mathbb{K}] \to\ \ldots\ \to \mathcal{W}_{N-1}[\mathbb{K}]. \tag{3.12}$$

All computations connected with (3.12) are conducted by a single computation node independently of other nodes.

4. Construction of Layers of Bellman Function

We note positions [29, Section 7]. In particular, from [29, Proposition 17], we have the equalities

$$D_s = \bigcup_{K \in \mathbf{G}_{N-1}} \mathcal{D}_s[K] \quad \forall s \in \overline{1, N-1}. \tag{4.1}$$

Equalities (4.1) provide for a "union" of the processes conducted by the separate computation nodes. Our intention in this is to construct the layers $v_1, \ldots, v_{N-1}$. For every $s \in \overline{1, N-1}$, we have the functions $\mathcal{W}_s[K]$, $K \in \mathbf{G}_{N-1}$, the domains of which (that is, the sets $\mathcal{D}_s[K]$, $K \in \mathbf{G}_{N-1}$) form, in view of (4.1), a cover of D_s. So the function v_s is completely determined. The functions from this collection agree in view of (3.5): for $K_1 \in \mathbf{G}_{N-1}$, $K_2 \in \mathbf{G}_{N-1}, s \in \overline{1, N-1}$, $(x, P) \in \mathcal{D}_s[K_1] \cap \mathcal{D}_s[K_2]$, we have

$$\mathcal{W}_s[K_1](x, P) = v_s(x, P) = \mathcal{W}_s[K_2](x, P). \tag{4.2}$$

In view of (4.1) and (4.2), we obtain the following simple rule of construction of the function v_s for $s \in \overline{1, N-1}$. Having "particular" functions, to determine $v_s(x_0, K_0)$, where $(x_0, K_0) \in D_s$, first, in view of (4.1), obtain the set $\mathbb{K}_0 \in \mathbf{G}_{N-1}$ such that $(x_0, K_0) \in \mathcal{D}_s[\mathbb{K}_0]$. Then, in view of (3.5), $v_s(x_0, K_0) = \mathcal{W}_s[\mathbb{K}_0](x_0, K_0)$. Property (4.2) means that a specific choice of $\mathbb{K}_0 \in \mathbf{G}_{N-1}$ with the property $(x_0, K_0) \in \mathcal{D}_s[\mathbb{K}_0]$ can be arbitrary. Based on the given rule, all values of the function v_s can be determined and, therefore, the function itself. Thus, being in possession of functions (3.5), we obtain all the layers $v_0, v_1, \ldots, v_N$ of the Bellman function. Based on these layers, through procedures that solve the local problems, we determine the optimal feasible solution. To construct such a solution, the computer should retain all the layers $v_1, \ldots, v_N$ of the Bellman function in its memory.

Algorithm for determining the global extremum. Consider a procedure for determining V (2.3), during the implementation of which, in the memory of each computational node, there is only retained one "particular" function of the form (3.5), i.e., an "individual" Bellman function layer. Without loss of generality, consider the computational node that processes the set $\mathbb{K} \in \mathbf{G}_{N-1}$. The ultimate goal of the procedure running on this node is to construct the function $\mathcal{W}_{N-1}[\mathbb{K}] \in \mathcal{R}_+[\mathcal{D}_{N-1}[\mathbb{K}]]$.

Principal steps of the iteration procedure.

1) Determine $\mathcal{W}_1[\mathbb{K}] \in \mathcal{R}_+[\mathcal{D}_1[\mathbb{K}]]$ from (3.9).

2) For $s \in \overline{1, N-2}$, assume the "particular" function $\mathcal{W}_s[\mathbb{K}] \in \mathcal{R}_+[\mathcal{D}_s[\mathbb{K}]]$ is already constructed. Then, through (3.10), compute the values of the function $\mathcal{W}_{s+1}[\mathbb{K}]$, which only use the values of the function $\mathcal{W}_s[\mathbb{K}]$ (in (3.10), set $l = s$). This yields the "particular" function $\mathcal{W}_{s+1}[\mathbb{K}] \in \mathcal{R}_+[\mathcal{D}_{s+1}[\mathbb{K}]]$. Next, the memory holding the values $\mathcal{W}_s[\mathbb{K}]$ is released, and then filled with the values of $\mathcal{W}_{s+1}[\mathbb{K}]$: the "particular" functions are overwritten.

3) After consecutively conducting step 2), we obtain $\mathcal{W}_{N-1}[\mathbb{K}]$.

After steps 1)–3) are completed at each computation node, there are obtained $\mathcal{W}_{N-1}[K],\ K \in \mathbf{G}_{N-1}$. Now, construct v_{N-1} through the equality

$$D_{N-1} = \bigcup_{K \in \mathbf{G}_{N-1}} \mathcal{D}_{N-1}[K]. \tag{4.3}$$

For every state $(\hat{x}, \hat{P}) \in D_{N-1}$, determine, by means of (4.3), the set $\hat{K} \in \mathbf{G}_{N-1}$ such that $(\hat{x}, \hat{P}) \in \mathcal{D}_{N-1}[\hat{K}]$. Then, we have the value $\mathcal{W}_{N-1}[\hat{K}](\hat{x}, \hat{P}) \in \mathbb{R}_+$ for which, in view of (3.5), there holds the equality $v_{N-1}(\hat{x}, \hat{P}) = \mathcal{W}_{N-1}[\hat{K}](\hat{x}, \hat{P})$. Thus, we obtain all the values of the function v_{N-1}. Through the function v_{N-1} (the layer of the Bellman function), we determine the global extremum V. In connection with this procedure, note [26], where it was proved that one can determine the value of V without constructing the optimal feasible solution for problems of greater dimension. For constructing of the optimal solution, the above-mentioned algorithm is changed: it is required to determine not only v_{N-1} but all functions $v_s, s \in \overline{1, N-1}$. These functions are required to be kept in the memory of the computer. The concrete variant of such constructing is connected with (4.2). Under construction of all functions $v_s, s \in \overline{1, N-1}$, it is required to use procedure of Section 3.

5. Model and Computational Experiment

Let us consider a specific version of the general problem; assume $X = \mathbb{R} \times \mathbb{R}$ (the problem on a plane). We study the formulation where each megalopolis is a system of entries and exits for a certain zone of heightened intensity of certain harmful factors (in particular, radiation). The necessary activities are conducted by a point agent in the zone. The aim of activities inside a megalopolis is to go from the entry point to the radiation source, dismantle it, and then leave the megalopolis through an exit point. The radiation left after dismantling a source is assumed to be negligible. Thus, a megalopolis is, essentially, a "near zone" of the radiation source. Other (non-dismantled) radiation sources are also assumed to have a nonnegligible effect both during movement between megalopolises and during the near zone activities (interior jobs). The effect of radiation is assumed to be cumulative, whence the additive cost aggregation assumed in this paper. Another feature of the problem statement is the fact that when moving to a source to dismantle it, as the agent enters the near zone, the agent is assumed to be affected by it; however, after dismantlement, as the agent moves to exit the zone, the agent is not affected by the (dismantled) source anymore. Consequently, the degree of radiation exposure of the agent during the first stage depends on the length of the stage and, therefore, on the distance between the entry point and the radiation source. During exterior movements, the effect of each single source is not as pronounced, however, the cumulative effect of non-dismantled sources can not be ignored.

Passing by a radiation source s and dismantling it. Let us connect the system of megalopolises with a system of point "radiating" objects $(\mathbf{z}_i)_{i\in\overline{1,N}} : \overline{1,N} \to X$ with the following property: $\mathbf{z}_i \notin M_i \ \forall i \in \overline{1,N}$. Dismantlement of the objects $\mathbf{z}_1, \ldots, \mathbf{z}_N$ is the aim of visiting the megalopolises: specifically, during a visit to $M_{\alpha(i)}$, the agent has to enter some entry point $\mathrm{pr}_1(z_i)$, move to the point $\mathbf{z}_{\alpha(i)}$, dismantle the radiation source number $\alpha(i)$, and then reach an exit point $\mathrm{pr}_2(z_i)$. This implements scheme (2.1).

Let us sketch a description of the effect of a single source during the movement from the given point of the plane to another point (as mentioned before, the exposure suffered during these movements is summed up) in the "regular" case, when the trajectory does not pass through any "active" radiation sources. Thus, as the agent moves from a point i to a point j, the losses (in our model), or rather, the exposure suffered during the movement due to the effect of the (active) source s is as follows:

$$\begin{gathered} c_{i,j}[\{s\}] = \int\limits_0^T \frac{\gamma_s}{\rho^2_{s,t}(t)} dt = \\ = 2\rho_{i,j}\frac{\gamma_s}{v}\int\limits_0^{\rho_{i,j}} \frac{d\rho}{\left(2\rho_{i,j}\rho + \rho^2_{j,s} - \rho^2_{i,s} - \rho^2_{i,j}\right)^2 + \left(4\rho^2_{i,j}\rho^2_{i,s} - (\rho^2_{j,s} - \rho^2_{i,s} - \rho^2_{i,j})^2\right)}, \end{gathered} \tag{5.1}$$

where ρ denotes the Euclidean distance (where necessary, the mentioned distances are subscripted); γ_s is the intensity of the source s, and v is the movement speed. To calculate the integral use one of the following table formulas:

$$\int \frac{dR}{A^2 + R^2} = \frac{1}{A}\mathrm{arctg}\frac{R}{A} + C, \int \frac{dR}{A^2 - R^2} = \frac{1}{2A}\ln\left|\frac{A+R}{A-R}\right| + C,$$

depending on whether the sign is $4\rho^2_{i,j}\rho^2_{i,s} - (\rho^2_{j,s} - \rho^2_{i,s} - \rho^2_{i,j})^2$. If the source s lies on the trajectory of the motion from point i to point j, the cost $c_{i,j}[\{s\}]$ is assumed to be a very large number (roughly speaking, $c_{i,j}[\{s\}] = \infty$; in the actual computation, the number that is several times greater than the most "costly" motion is sufficient).

Approaching radiation source s for dismantlement (determining the cost of interior jobs). In this case, assume that the point j coincides with s and consider the exposure suffered by the agent during the approach to the source that is to be dismantled.

Thus, we still follow the model where the exposure is inversely proportional to the square of the distance to the source; however, to avoid the ill-posedness as the agent reaches the source, which could lead to a division by 0, the denominator of the integrand in (5.1) is increased by 1. To account for a more intense radiation in the near zone, we add factor 3 (which describes the case of radiation intensity not being weakened by obstacles, the latter being possible during exterior movements). So, in this case, the losses (the exposure) are calculated by the following formula:

$$c_{i,j}[\{s\}] = c_{i,s}[\{s\}] = 3\big(\frac{\gamma_s}{v}\big)\int\limits_0^{\rho_{i,j}} \frac{d\rho}{\rho^2+1} = 3\big(\frac{\gamma_s}{v}\big)\mathrm{arctg}(\rho_{i,j}). \qquad (5.2)$$

About constructing of cost functions. For constructing of $\mathbf{c}, c_1, \ldots, c_N$, it is required the values defined by (5.1) and (5.2) are summed from $s \in K$, where K is the corresponding list of tasks that are not implemented at the time of moving (we keep in mind the determination of values $\mathbf{c}(x, y, K)$ or $c_j(x, y, K)$, where $j \in \overline{1, N}$).

Computational experiment. Let us consider the model instances of the routing problem for dismantling radiation sources on a plane. The megalopolises, which imitate the possible entry and exit points of the chambers housing the radiation sources, are obtained by discretizing circles: at each circle, there are 12 points, equally spaced starting with the point with the angle coordinate of 0. Each megalopolis is assigned a point object, which imitates the radiation source of the chamber. Let the starting point of the dismantlement process coincide with the point of origin, i.e., $x^0 = (0, 0)$; after dismantling all radiation sources, the agent must return to the depot (the point of origin). Recall that function ρ is essentially the Euclidean distance. Assume the speed of the agent is 4 times greater outside the chambers than it is within — this models the intrinsic difficulty of moving inside the megalopolises, which is due to the presence of hardware, various structures, or mechanisms, which hamper the rapid movement inside. Here are some results concerning the calculations on supercomputers URAN. For the model with 30 megalopolises and 30 address pairs (51 pairs in transitive closure [31]) that determine the precedence constraints, the following results were obtained: total exposure (total losses): 222,9, total computation time: 17m 46s. For the model with 31 megalopolis and 34 address pairs (63 pairs in transitive closure [31]), the following results were obtained: total exposure (total losses): 226,5, total computation time: 15m 56s. Let us now consider the results of the accounts of the same tasks on a PC. For the model with 30 megalopolises and 30 address pairs, the following results were obtained: total exposure (total losses): 222,9, total computation time: 7h 26m 7s. For the model with 31 megalopolis and 34 address pairs, the following results were obtained: total exposure (total losses): 226,5, total computation time: 6h 29m 55s. A decrease in computation time for the problem with the greater number of megalopolises is due to the more "strict" precedence constraints, which significantly decrease both the volume of computations involved in obtaining the values of the Bellman function and the memory usage. For more information on the influence of precedence constraints on computation time and memory usage of dynamic programming solutions, refer to [32]. Calculating the optimal route for the passage of 45 megacities, with 20 cities each is realized. This setting is weakened by 40 conditions of precedence. The calculation was performed on the URAN supercomputer in the interval 1 hour 1 minute.

Acknowledgements. *This research was supported by Russian Foundation for Basic Research (projects no. 17-08-01385, 18-07-00637).*

References

1. Garey M.R., Johnson D.S. *Computers and Intractability: A Guide to the Theory of NP-Completeness*, N.Y., W.H. Freeman, 1979.
2. Gutin G., Punnen A.P. *The Traveling Salesman Problem and Its Variations*, N.Y., Springer, 2002.
3. Cook W.J. *In Pursuit of the Traveling Salesman. Mathematics at the Limits of Computation*, New Jersey, Princeton University Press, 2012.
4. Melamed I.I., Sergeev S.I., Sigal I. The Traveling Salesman Problem. Issues in Theory. *Automation and Remote Control*, 1989, vol. 50, no. 9, pp. 1147–1173.
5. Melamed I.I., Sergeev S.I., Sigal I. The Traveling Salesman Problem. Exact Methods. *Automation and Remote Control*, 1989, vol. 50, no. 10, pp. 1303–1324.
6. Melamed I.I., Sergeev S.I., Sigal I. The Traveling Salesman Problem. Approximate Algorithms. *Automation and Remote Control*, 1989, vol. 50, no. 11, pp. 1459–1479.
7. Little L.D.C., Murty K.G., Sweeney D.W., Karel C. An Algorithm for the Travelling Salesman Problem. *Operations Research*, 1963, vol. 11, no. 6, pp. 972–990. DOI: 10.1287/opre.11.6.972
8. Bellman R. Dynamic Programming Treatment of the Travelling Salesman Problem. *Journal of the Association for Computing Machinery*, 1962, vol. 9, pp. 61–63. DOI: 10.1145/321105.321111
9. Held M., Karp R.M. A Dynamic Programming Approach to Sequencing Problems. *Journal of the Society for Industrial and Applied Mathematics*, 1962, vol. 10, no. 1, pp. 196–210. DOI: 10.1137/0110015
10. Leon V.J., Peters B.A. Replanning and Analysis of Partial Setup Strategies in Printed Circuit Board Assembly Systems. *International Journal of Flexible Manufacturing Systems*, 1996, vol. 8, pp. 389–411. DOI: 10.1007/BF00170019
11. Alkaya A.F., Duman E. A New Generalization of the Traveling Salesman Problem. *Applied and Computational Mathematics*, 2010, vol. 9, no. 2, pp. 162–175.
12. Kinable J., Cire A., van Hoeve W.J. Hybrid Optimization Methods for Time-Dependent Sequencing Problems. *European Journal of Operational Research*, 2017, vol. 259, no. 3, pp. 887–897. DOI: 10.1016/j.ejor.2016.11.035
13. Chentsov A.G. *Ekstremal'nye zadachi marshrutizatsii i raspredeleniya zadaniy: voprosy teorii* [Extreme Problems of Routing and Tasks Distribution: Regular and Chaotic Dynamics]. Izhevsk, Izhevsk Institute of Computer Research, 2008. (in Russian)
14. Korobkin V.V., Sesekin A.N., Tashlykov O.L., Chentsov A.G. *Routing Methods and Their Applications to the Enhancement of Safety and Efficiency of Nuclear Plant Operation.* Moscow, Novye tekhnologii, 2012. (in Russian)
15. Tashlykov O.L. *Personnel Dose Costs in the Nuclear Industry. Analysis. Ways to Decrease. Optimization.* Saarbruke, LAP LAMBERT Academic Publishing GmbH & Co. RG., 2011.

16. Petunin A.A. About Some Strategies of the Tool Path Modelling at the Control Programs Generation for the Flame Cutting Machines. *Vestnik UGATU*, 2009, vol. 13, no. 2, pp. 280–286. (in Russian)

17. Petunin A.A., Chentsov A.G., Chentsov P.A. On Routing Tool Motion on the Sheet Cutting NPC Machines. *St. Petersburg State Polytechnical University Journal. Computer Science. Telecommunication and Control Systems*, 2013, no. 2, pp. 103–111. (in Russian)

18. Frolovskii V.D. Computer-Aided Design of the Control Programs for Thermal Metal Cutting on NPC Machines. *The scientific and technical journal "Information Technology of Cad/Cam/Cae" (ITDP)*, 2005, no. 4, pp. 63–66. (in Russian)

19. Wang G.G., Xie S.Q. Optimal Process Planning for a Combined Punch-and-Laser Cutting Machine Using ant Colony Optimization. *International Journal of Production Research*, 2005, vol. 43, no. 11, pp. 2195–2216. DOI: 10.1080/00207540500070376

20. Dewil R., Vansteenwegen P., Cattrysse D. Construction Heuristics for Generating Tool Paths for Laser Cutters. *International Journal of Production Research*, 2014, vol. 52, no. 20, pp. 1–20. DOI: 10.1080/00207543.2014.895064

21. Chentsov A.G., Chentsov P.A. Routing Under Constraints: Problem of Visit to Megalopolises. *Automation and Remote Control*, 2016, vol. 77, no. 11, pp. 1957–1974. DOI: 10.1134/S0005117916110060

22. Dieudonne J. *Foundations of Modern Analysis.* N.Y., Academic, 1960.

23. Cormen T.H., Leizerson C.E., Rivest R.L. *Introduction to Algorithms.* Cambridge, MIT Press, 1990.

24. Chentsov A.G. To Question of Routing of Works Complexes. *Vestnik Udmurtskogo universiteta. Matematika. Mekhanika. Kompyuternye nauki*, 2013, no. 1, pp. 59–82. (in Russian)

25. Chentsov A.G., Chentsov A.A. Route Problem with Constraints Depending on a List of Tasks. *Doklady Mathematics*, 2015, vol. 92, no. 3, pp. 685–688. DOI: 10.1134/S1064562415060083

26. Chentsov A.G., Chentsov A.A. On the Problem of Obtaining the Value of Routing Problem with Constraints. *Journal of Automation and Information Sciences*, 2016, vol. 6, pp. 41–54.

27. Lawler E.L. Efficient Implementation of Dynamic Programming Algorithms for Sequencing Problems. *CWI Technical report. Stichting Mathematisch Centrum. Mathematische Besliskunde-BW*, 1979, vol. 106, no. 79, pp. 1–16.

28. Chentsov A.G. One Parallel Procedure for the Construction of the Bellman Function in the Generalized Problem of the Courier with the Inner Workings. *Automation and Remote Control*, 2012, vol. 3, pp. 134–149.

29. Chentsov A.G. One Parallel Procedure for the Construction of the Bellman Function in the Generalized Problem of the Courier with the Inner Workings. *Bulletin of the South Ural State University. Series: Mathematical Modelling, Programming and Computer Software*, 2012, no. 18 (277), pp. 53–76. (in Russian)

30. Chentsov A.G., Grigoryev A.M. Dynamic Programming Method in the Route Problem: the Scheme of Independent Calculations. *Mekhatronika, avtomatizatsiya, upravlenie*, 2016, vol. 17, no. 12, pp. 834–846.

31. Schmidt G., Strohlein T. *Relations and Graphs: Discrete Mathematics for Computer Scientists.* London, EATCS Monographs on Theoretical Computer Science, Springer-Verlag, 1993.

32. Steiner G. On the Complexity of Dynamic Programming for Sequencing Problems with Precedence Constraints. *Annals of Operations Research*, 1990, vol. 26, no. 1, pp. 103–123. DOI: 10.1007/BF02248587

Received November 23, 2017

УДК 519.6 **DOI: 10.14529/mmp180106**

РЕШЕНИЕ ЗАДАЧИ МАРШРУТИЗАЦИИ С ИСПОЛЬЗОВАНИЕМ СХЕМЫ НЕЗАВИСИМЫХ ВЫЧИСЛЕНИЙ

А.Г. Ченцов[1,2]***, А.М. Григорьев***[1]***, А.А. Ченцов***[1]

[1]Институт математики и механики им. Н. Н. Красовского УрО РАН, г. Екатеринбург, Российская Федерация

[2]Уральский федеральный университет им. первого Президента России Б.Н. Ельцина, г. Екатеринбург, Российская Федерация

Статья посвящена вопросам построения и реализации параллельных алгоритмов для решения прикладных задач. Рассматривается задача маршрутизации перемещений с ограничениями и усложненными функциями стоимости. Предполагается, что объекты посещения — суть мегаполисы (непустые конечные множества), при посещении которых должны выполнятся некоторые работы, именуемые далее внутренними. По постановке задачи имеются ограничения в виде условий предшествования. Стоимости перемещений зависят от списка заданий, которые не выполнены на момент перемещения. Ситуация такого рода возникает, в частности, при аварийных ситуациях, связанных с работой АЭС и подобных происходящим в Чернобыле и Фукусиме. Речь идет об утилизации источников радиоактивного излучения, осуществляемой последовательно во времени; в этом случае исполнитель находится под воздействием источников, которые не были демонтированы на момент соответствующего перемещения. За счет этого в функциях стоимости, оценивающих воздействие радиации на исполнителя, возникает зависимость от списка невыполненных заданий. Последние состоят в том или ином варианте «выключения» соответствующего источника. В настоящем исследовании излагается подход к решению данной задачи параллельным алгоритмом, реализуемым на суперкомпьютере УРАН. Приведены результаты вычислительного эксперимента.

Ключевые слова: динамическое программирование; маршрут; условия предшествования; параллельный алгоритм.

Литература

1. Garey, M.R. Computers and Intractability: A Guide to the Theory of NP-Completeness / M.R. Garey, D.S. Johnson. – N.Y., W.H. Freeman, 1979.
2. Gutin, G. The Traveling Salesman Problem and Its Variations / G. Gutin, A.P. Punnen. – N.Y., Springer, 2002.
3. Cook, W.J. In Pursuit of the Traveling Salesman. Mathematics at the Limits of Computation / W.J. Cook. – New Jersey, Princeton University Press, 2012.
4. Меламед, И.И. Задача коммивояжера. Вопросы теории / И.И. Меламед, С.И. Сергеев, И.Х. Сигал // Автоматика и телемеханика. – 1989. – Т. 50, № 9. – С. 3–34.
5. Меламед, И.И. Задача коммивояжера. Точные алгоритмы / И.И. Меламед, С.И. Сергеев, И.Х. Сигал // Автоматика и телемеханика. – 1989. – Т. 50, № 10. – С. 3–29.

6. Меламед, И.И. Задача коммивояжера. Приближенные алгоритмы / И.И. Меламед, С.И. Сергеев, И.Х. Сигал // Автоматика и телемеханика. – 1989. – Т. 50, № 11. – С. 3–26.
7. Литл, Дж. Алгоритм для решения задачи о коммивояжере / Дж. Литл, К. Мурти, Д. Суини, К. Кэрел // Экономика и математические методы. – 1965. – Т. 1, вып. 1. – С. 94–107.
8. Беллман, Р. Применение динамического программирования к задаче о коммивояжере / Р. Беллман // Кибернетический сборник. – Т. 9.– М.: Мир, 1964. – С. 219–228.
9. Хелд, М. Применение динамического программирования к задачам упорядочения / М. Хелд, Р.М. Карп // Кибернетический сборник. – Т. 9. – М.: Мир, 1964. – С. 202–218.
10. Leon, V.J. Replanning and Analysis of Partial Setup Strategies in Printed Circuit Board Assembly Systems / V.J. Leon, B.A. Peters // International Journal of Flexible Manufacturing Systems. – 1996. – V. 8. – P. 389–411.
11. Alkaya, A.F. A New Generalization of the Traveling Salesman Problem / A.F. Alkaya, E. Duman // Applied and Computational Mathematics. – 2010. – V. 9, № 2. – P. 162–175.
12. Kinable, J. Hybrid Optimization Methods for Time-Dependent Sequencing Problems / J. Kinable, A. Cire, W.J. van Hoeve // European Journal of Operational Research. – 2017. – V. 259, № 3. – P. 887–897.
13. Ченцов, А.Г. Экстремальные задачи маршрутизации и распределения заданий: вопросы теории / А.Г. Ченцов. – Ижевск: НИЦ «Регулярная и хаотическая динамика», Ижевский институт компьютерных исследований, 2008.
14. Коробкин, В.В. Методы маршрутизации и их приложения в задачах повышения безопасности и эффективности эксплуатации атомных станций / В.В. Коробкин, А.Н. Сесекин, О.Л. Ташлыков, А.Г. Ченцов. – М.: Новые технологии, 2012.
15. Ташлыков, О.Л. Дозовые затраты персонала в атомной энергетике. Анализ. Пути снижения. Оптимизация: монография / О.Л. Ташлыков. – Saarbruke: LAP LAMBERT Academic Publishing GmbH & Co. RG., 2011.
16. Петунин, А.А. О некоторых стратегиях формирования маршрута инструмента при разработке управляющих программ для машин термической резки материала / А.А. Петунин // Вестник УГАТУ. Серия: Управление, вычислительная техника и информатика. – 2009. – Т. 13, № 2 (35). – С. 280–286.
17. Петунин, А.А. К вопросу о маршрутизации движения инструмента в машинах листовой резки с числовым программным управлением / А.А. Петунин, А.Г. Ченцов, П.А. Ченцов // Научно-технические ведомости СПбГПУ. Информатика. Телекоммуникации. Управление. – 2013. – № 2 (169). – С. 103–111.
18. Фроловский, В.Д. Автоматизация проектирования управляющих программ тепловой резки металла на оборудовании с ЧПУ / В.Д. Фроловский // Информационные технологии в проектировании и производстве. – 2005. – № 4. – С. 63–66.
19. Wang, G.G. Optimal Process Planning for a Combined Punch-and-Laser Cutting Machine Using and Colony Optimization / G.G. Wang, S.Q. Xie // International Journal of Production Research. – 2005. – V. 43, № 11. – P. 2195–2216.
20. Dewil, R. Construction Heuristics for Generating Tool Paths for Laser Cutters / R. Dewil, P. Vansteenwegen, D. Cattrysse // International Journal of Production Research. – 2014. – P. 1–20.
21. Ченцов, А.Г. Маршрутизация в условиях ограничений: задача о посещении мегаполисов / А.Г. Ченцов, П.А. Ченцов // Автоматика и телемеханика. – 2016. – № 11. – С. 96–117.

22. Dieudonne, J. Foundations of Modern Analysis / J. Dieudonne. – N.Y.: Academic, 1960.

23. Cormen, T.H. Introduction to Algorithms / T.H. Cormen, C.E. Leizerson, R.L. Rivest. – Cambridge: MIT Press, 1990.

24. Ченцов, А.Г. К вопросу о маршрутизации комплексов работ / А.Г. Ченцов // Вестник Удмуртского ун-та. Математика. Механика. Компьютерные науки. – 2013. – № 1. – С. 59–82.

25. Ченцов, А.Г. Задача маршрутизации с ограничениями, зависящими от списка заданий / А.Г. Ченцов, А.А. Ченцов // Доклады Академии наук. – 2015. – Т. 465, № 2. – С. 154–158.

26. Ченцов, А.А. К вопросу о нахождении значения маршрутной задачи с ограничениями / А.А. Ченцов, А.Г. Ченцов // Проблемы управления и информатики. – 2016. – № 1. – С. 41–54.

27. Lawler, E.L. Efficient Implementation of Dynamic Programming Algorithms for Sequencing Problems / E.L. Lawler // CWI Technical report. Stichting Mathematisch Centrum. Mathematische Besliskunde-BW. – 1979. – V. 106, № 79. – P. 1–16.

28. Ченцов, А.Г. Одна параллельная процедура построения функции Беллмана в обобщенной задаче курьера с внутренними работами / А.Г. Ченцов // Автоматика и телемеханика. – 2012. – № 3. – С. 134–149.

29. Ченцов, А.Г. Одна параллельная процедура построения функции Беллмана в обобщенной задаче курьера с внутренними работами / А.Г. Ченцов // Вестник ЮУрГУ. Серия: Математическое моделирование и программирование. – 2012. – № 18 (277), вып. 12. – С. 44–52.

30. Ченцов, А.Г. Динамическое программирование в задаче маршрутизации: схема независимых вычислений / А.Г. Ченцов, А.М. Григорьев // Мехатроника, автоматизация, управление. – 2016 – Т. 17, № 12. – С. 834–846.

31. Schmidt, G. Relations and Graphs: Discrete Mathematics for Computer Scientists / G. Schmidt, T. Strohlein // EATCS Monographs on Theoretical Computer Science. – N.Y.: Springer-Verlag, 1993.

32. Steiner, G. On the Complexity of Dynamic Programming for Sequencing Problems with Precedence Constraints / G. Steiner // Annals of Operations Research. – 1990. – V. 26, № 1. – P. 103–123.

Александр Георгиевич Ченцов, член-корреспондент РАН, доктор физико-математических наук, профессор, главный научный сотрудник, Институт математики и механики им. Н.Н. Красовского УрО РАН; Уральский федеральный университет им. первого Президента России Б.Н. Ельцина (г. Екатеринбург, Российская Федерация), chentsov@imm.uran.ru.

Алексей Михайлович Григорьев, заведующий отделом, Институт математики и механики им. Н.Н. Красовского УрО РАН (г. Екатеринбург, Российская Федерация), ag@uran.ru.

Алексей Александрович Ченцов, кандидат физико-математических наук, научный сотрудник, Институт математики и механики им. Н. Н. Красовского УрО РАН (г. Екатеринбург, Российская Федерация), chentsov_a_a@mail.ru.

Поступила в редакцию 23 ноября 2017 г.