Tool Routing Problem for CNC Plate Cutting Machines

Pavel A. Chentsov*. Alexander A. Petunin**

*Institute of Mechanics and Machine Building, Ural Federal University, Yekaterinburg, Russia, Institute of mathematics and mechanics UrB RAS, Yekaterinburg, Russia (Tel.: +7-343-362-81-62, e-mail: chentsov.p@uran.ru)

**Institute of Mechanics and Machine Building, Ural Federal University, Yekaterinburg, Russia (Tel: +7-343-375-97-14, e-mail: aapetunin@gmail.com).

The problem of sheet cutting optimization for CNC (Computer Numerical Control) plate cutting machines is considered. This problem includes restriction with engineering specifics. The heuristic method of the problem solving is offered. This is the algorithm of the generalized salesman problem solving with additional restrictions in form of precedence constraints and based on previous part of the route restrictions. The iterative method of algorithm using is given.

© 2016, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.

Keywords: routing (tool path) problem, route optimization, CNC cutting machines.

1. INTRODUCTION

The quality of producing on the CNC cutting machines parts highly depends on the sequence of parts cut. Moreover, the correct tool routing can decrease the total cutting time.

The general problem of cutting sheets of metal (see Petunin (2011)) consists of a nesting problem (features to be cut must be placed on the sheet), a problem of searching for feasible piercing points, and a problem of optimizing the tool path. This paper is devoted to the problem of tool path optimization that would take into account certain constraints on the order of visiting the features: precedence constraints arising from the need to cut the innermost contour first, geometrical, and thermal expansion restrictions.

The tool path problem is a complex problem of discrete and continuous optimization. Only a small number of cases of this problem was formalized in a mathematical way. The existing mathematical models and algorithms for solving of problem (see, for example, Castelino (2003), Dewil (2011), Yang (2010)) ignore many real technological constraints of the cutting process. In particular, the constraints connected with thermal deformation were not formalized to the best of the author's knowledge; the statements that we know were only concerned with minimizing the idle time (airtime) of the cutting tool. Some technological constraints on the cutting process and the solution of the corresponding tool path problem are described in Petunin (2009).

Our problem is essentially a generalized travelling salesman problem with additional constraints (the precedence constraints) that takes into account the direction of cutting the contours and uses a specific, complicated cost functions There are well-known exact methods of solving the generalized travelling salesman problem, see [Fischetti (1997), Chentsov (2013), Chentsov2 (2013)]. For heuristic algorithms, refer to Chentsov (2014), Petunin (2013), Snyder (2006), Silberholz (2007), and Gutin (2010).

Thus, our principal objective is to improve the quality of the products by complying with geometrical, thermal, and precedence constraints and minimize the time to finish the cutting. To this end, we construct a heuristic algorithm, which is an improved version of the heuristic algorithm from (Chentsov (2014)). An iterative scheme of this algorithm was proposed in Chentsov (2014). One of most important contributions of the present paper is the use of cost functions that *depend on the set of pending tasks*, which was never implemented before. This is the reason for only citing certain similar problem statements.

2. PROBLEM STATEMENT

2.1 Main definitions

Our workpiece is a steel sheet of fixed size, from which the features – our end products – are cut. Denote the start and finish points of tool motions by p^{st} and p^{fin} ($p^{st} \in R \times R$, $p^{fin} \in R \times R$, where R is the real line). The number of contours is N (some features consist of several contours). The cutting tool has two motion modes, the idling run with the speed V_{idling} and the cutting run with the speed V_{cut}.

The tool must be switched on (and the sheet must be pierced) outside the contour because the action of piercing the sheet involves strong thermal expansion. Moreover, it is possible for the radius of the bore that forms as the sheet is pierced to exceed the required cut width. Thus, the piercing point has to be located outside the contour, yet relatively close to it. After the tool is switched on, it moves to the point that starts the cutting of the contour; this *starting point* is in one-to-one correspondence with the *piercing point*. The contour may be cut in two directions, clockwise and counterclockwise. Often, the tool *switch-off point* must also be located outside the contour; in this case, after the contour is completely cut, the tool moves towards this switch-off point; however, the *switch-off point* may also coincide with the *starting point*. In the general case, we will consider three separate points: the

piercing point, the *starting point,* and the *switch-off* point. See Fig. 1 for an example of such a tool path.
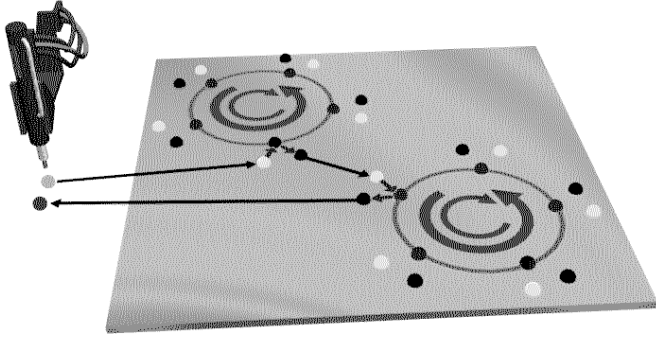

Fig. 1. Cutting tool motion.

There are also geometrical and thermal constraints; simply put, there must be enough metal near the end of the cutting process for a contour, see Fig. 2.
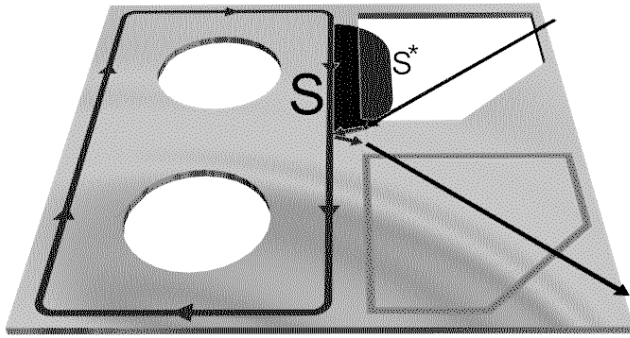

Fig. 2. Geometrical and thermal constraints.

The quality of cut for each contour depends on the value S_* / S^*.

Thus, the processing of every contour starts and ends with a triple of points, the *piercing point,* the *starting point,* and the *switch-off point.* There are $\tilde{N}$ such triples. Let M, $M = \overline{1, \tilde{N}}$, be the set of all indices of these triples.

Let $M_1, ..., M_N$, $N \geq 2$, be the sets of the indices of contour point triples for the contours. Each contour has its own number of the cutting point triples, $|M_i| = n_i$, $n_i \geq 1$. Note that

$$\bigcup_{i \in \overline{1,N}} M_i = M,$$

$$M_i \cap M_j = \varnothing \quad \forall i \in \overline{1, N}, \forall j \in \overline{1, N} \setminus \{i\}.$$

The cutting points are formalized through the following three functions:

$$p^{on} : M \to R \times R,$$

$$p^{cut} : M \to R \times R,$$

$$p^{off} : M \to R \times R.$$

Let us now consider the concept of route. Let A be a family of ordered sets. If $a \in A$, then

$$a = (a_i)_{i \in \overline{1,N}} : \overline{1, N} \to M,$$

$$\forall a_1 \in a, \forall a_2 \in a \setminus \{a_1\}, \neg \exists i \in \overline{1, N} :$$

$$(a_1 \in M_i) \,\&\, (a_2 \in M_i)$$

Thus, for $a \in A$, the route of the cutting tool (i.e., the tool path) is

$$p^{st} \to p^{on}(a_1) \to p^{cut}(a_1) \to p^{off}(a_1) \to ...$$

$$... \to p^{on}(a_N) \to p^{cut}(a_N) \to p^{off}(a_N) \to p^{fin}.$$

To define the cost of internal movements, we would need the direction sign d, $d \in \{0,1\}$. Let the value 0 corresponds to a clockwise cut process and 1 to counterclockwise cut.

Denote by D the ordered set of contour tool motion directions: if $d \in D$, then

$$d = (d_i)_{i \in \overline{1,N}} : \overline{1, N} \to \{0;1\}.$$

If $d_i = 1$, then the contour at the position i in the route is cut in the clockwise direction; if $d_i = 0$, the direction is counterclockwise. In addition, we need the concept of partial route and the corresponding directions set over a part of contours, described by a set K, $K \subset \overline{1, N}$. Denote them by $A(K)$ and $D(K)$, respectively; they are defined similarly to A and D.

2.2 Cost functions

The constants V_{idling} and V_{cut} are idling and work speeds, respectively. The cost of moving from one point to another is the time required to complete this motion. If x_1 and x_2 are start and finish points of the switched-off tool (p^{st}, p^{fin}, p^{on} or p^{off}), then the cost of this is calculated as follows:

$$C(x_1, x_2) = \frac{\rho(x_1, x_2)}{V_{idling}}.$$

Here, ρ is the Euclidean distance between the points x_1 and x_2.

Denote by $\widetilde{M}$ the family of all subsets of M (including the empty set). The total cost of a cut is obtained with the aid of the function $c_d : M \times \{0;1\} \times \widetilde{M} \to R$.

c_d is penalty for cut finishing by location of finish cut area with respect to cutted out contours at this time. Note that for final result estimation $c_d \equiv 0$, it is needed only when count.

c_d is the penalty describing how bad it will be to cut the contour as intended (through the given points, in the given direction) in view of workpiece rigidity constraints (i.e., with respect to the contours that are already cut). Note that after the algorithm finishes the calculation of the cost of the route, we set $c_d \equiv 0$ – this penalty is only used during the calculation.

Thus, the cost of cutting a $i, i \in \overline{1, N}$ – through the cutting points $m, m \in M_i$ in the direction d when the set of contours $K, K \subset \overline{1, N} \setminus \{i\}$, is still to be cut out – is calculated as follows:

$$c(m, d, K) = \frac{\rho(p^{on}(m), p^{cut}(m))}{V_{cut}} + \frac{\rho(p^{cut}(m), p^{off}(m))}{V_{cut}} + c_d(m, d, K) \quad (1)$$

Note the absence of the time actually required to cut the contour. This time does not depend on the sequence of the contours cut and it is not optimized.

The function c_d has the following form:

$$c_d(m, d, K) = \frac{S^*(m, d, K)}{S(m, d, K)} P \quad (2)$$

$S(m, d, K)$ is the checking area, the area of the part of the workpiece near the end of the contour that should not be cut prior to cutting the present contour; $S^*(m, d, K)$ is the area of the intersection of the checking area with any cuts in the (see the details in Fig. 2); P is the penalty coefficient.

The areas S and S^* are calculated by way of an area matrix. It is an approximate calculation; obviously, how fast and how precise is this calculation depends on the dimensionality of this matrix. However, precision needs not be too high.

The workpiece and the area around it is divided into squares, the area matrix cells. A cell outside the workpiece gets the value of -1. A cell inside the workpiece that is outside of every contour gets the value 0. If a cell is inside a contour, its value is the index of this contour; in case it is inside multiple contours, its value matches the index of the nearest contour inside which it still fits, as exemplified by Fig. 3.
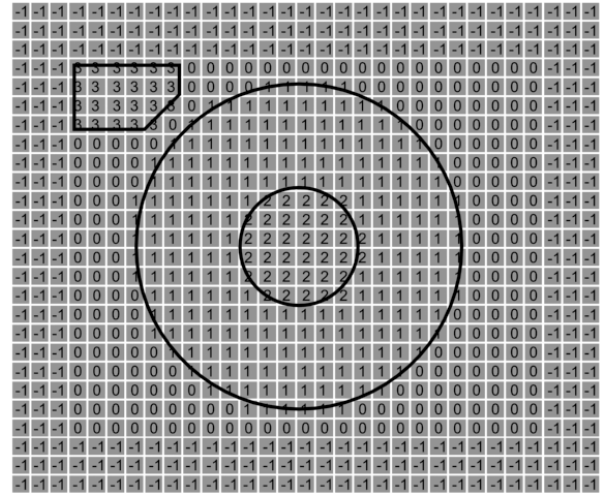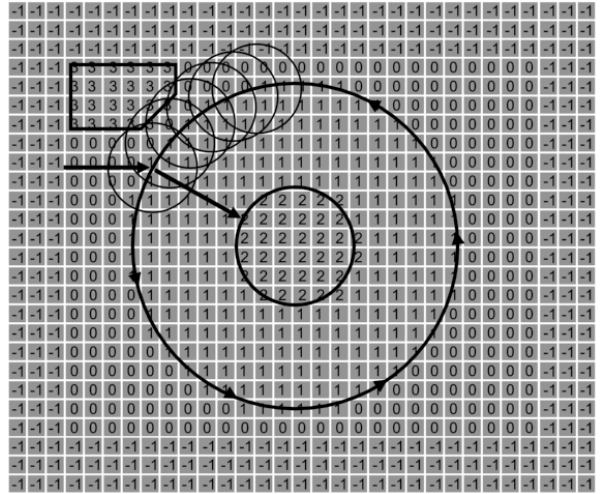

Fig. 3. Area matrix.

Fig. 4. Geometrical and thermal constraints.

Near the end of the cut, put N_L equally spaced points. Surround each of them with a circle, and find the intersection of this circle with the matrix's cells. The neighboring circles may intersect; the cells within those intersections are only counted once, which is done with the aid of a special Boolean, the size of which matches that of the area matrix; see the example on Fig. 4.

For every cell intersecting with the given areas, the following check is conducted: if a contour (the cut of which we analyze) is a parent contour and a cell's number corresponds to its number or to the one of its child contours, this cell is ignored. For a child contour, cells with the number that coincides neither with the number of the current nor of its child contours are ignored too. For a parent contour, the value -1 means there is no metal at the cell and the value 0 means there is. If the value is above 0, then the cell has metal if the contour with the index that matches the value of the cell is not yet cut out; otherwise, the cell is assumed to be empty. The ratio between S and S^* from (2) is calculated as the ratio between the number of filled cells and the number of empty cells.

Program optimizations make these computations fast enough.

2.3 Precedence constraints

The empty cells "fall down" and may not be cut again; if there are contours nested inside them, it will not be possible to cut those contours. Hence the demand to always cut child contours before the parent contours.

Let Z be the set of address pairs z for precedence constraints,

$$z = (z_1, z_2), z_1 \in \overline{1, N}, z_2 \in \overline{1, N}, z_1 \neq z_2,$$

(3) $$|Z| \geq 0.$$

For $z \in Z$, z_1 describes the index of the contour that must be visited before the contour z_2 (i.e., the indexes of the source and destination contours). Note that some $z \in Z$ can make the whole problem infeasible; however, such combinations of address pairs are precluded by the nature of the task. The influence of these constraints on the computational complexity of toolpath routing problems was studied in Salii (2013).

In view of the set Z, certain routes from A become infeasible (see (3)). The family of feasible routes is denoted by $\widetilde{A}$, $\widetilde{A} \subset A$: $\forall a \in \widetilde{A}$ $\forall z \in Z$ if $a_i \in M_{z_1}$ and $a_i \in M_{z_2}$ then $i < j$.

2.4 The main problem

The cost of a route $a \in \widetilde{A}$ with cut directions $d \in D$ is

$$T(a, d) = C(p^{st}, p^{on}(a_1)) + c(a_1, d_1, \varnothing) +$$

$$+ \sum_{i=2}^{N} \left(C(p^{off}(a_{i-1}), p^{on}(a_i)) + c\left(a_i, d_i, \bigcup_{j \in 1, i-1} a_j\right) \right) +$$

$$C(p^{off}(a_N), p^{fin}).$$

Now, we can finally state the main problem:

(4) $$T(a, d) \to \min, \quad a \in \widetilde{A}, d \in D.$$

3 Heuristic algorithm

3.1 Greedy routing algorithm

The calculation of areas in (2), which depends on the route traveled, is a computationally comple problem, which essentially precludes exact methods (such as the dynamic programming and branch and bound methods) – those are only feasible for small problem instances. Thus, the main means of solving problem (4) are heuristic algorithms.

In this section, we describe a greedy algorithm for problem (4) and an iterative method. Both algorithms make sure the results satisfy precedence constraints as well as geometrical and thermal constraints.

1. Find a contour with an index $i, i \in \overline{1, N}$, such that $i \neq z_2$ $\forall z \in Z$ and $m \in M_i$ and $d \in \{0,1\}$ that minimize the expression $C(p^{st}, p^{on}(m)) + c(m, d, \varnothing)$. Add to the route node the a_1, $a_1 = m$ and $d_1 = d$. Tag the contour with the index **i** as visited.

2. By $V, V \subset \overline{1, N}$, denote the current set of visited contours. Next, denote by W denote the set of child contours, the parent of which is not yet visited ($W = \{z_2 \mid z \in Z, z_1 \notin V\}$). The triple a_l is the last triple of the current route, its cut direction is d_l. Find the contour with the index $i, i \in \overline{1, N} \setminus \{V \cup W\}$ and $m \in M_i$ and $d \in \{0,1\}$ that minimize the expression $C(p^{off}(a_l), p^{on}(m)) + c(m, d, V)$. Add to the end of the route the node a_{l+1}: $a_{l+1} = m$ and $d_{l+1} = d$. Tag the contour with the index **i** as visited.

3. Carry out step 2 until the route complete.

This algorithm gives us a feasible route. Below, we state an iterative method that uses the result of the algorithm describe above and make its result better. For every iteration, we modify the cost function $c(m, d, K)$: for a given contour $i, i \in \overline{1, N}$, and its given place in the route $j, j \in \overline{1, N}$, set $c(m, d, K) = 1000000$ $\forall m \in M_i$ if the triplet **m** is placed at the position **j** in the route.

The matrix $D_{i,j} \in \{0;1\}$, $i \in \overline{1, N}$, $j \in \overline{1, N}$, describes the cost corrections. If $D_{i,j} = 1$, then $c(m, d, K) = 1000000$ $\forall m \in M_i$ $\forall d \in \{0,1\}$ and $\forall K \subset \overline{1, N} \setminus \{i\}$ if the triple **m** is placed at the position **j** in the route. If $D_{i,j} = 0$, then the cost is calculated in the ordinary way, see (1). Initially, $D_{i,j} = 0$ $\forall i \in \overline{1, N}$, $\forall j \in \overline{1, N}$. Cost corrections are used during the operation of the algorithm from (2). The final cost is calculated without these corrections (the actual operating time).

The parameter N_1 describes the total number of iterations. Every iteration consists of the following two parts: making a cost correction and rebuilding the route. Every N_2 iterations all corrections are set to zero.

Iterative method employing the greedy algorithm

1. Carry out the algorithm without any corrections. Store the cost value, route, and directions.

2. The current route is $a, a \in \widetilde{A}$ and the directions are $d, d \in D$. Choose a random route position $i, i \in \overline{1, N}$. There is $a_i \in M_j$, $j \in \overline{1, N}$. Set $D_{j,i} = 1$.

3. Carry out the algorithm. If the cost is improved, store the improved value, the route and the directions.

4. Repeat Steps 2 and 3 for $N_2 - 1$ times ($N_2 - 2$ for the first cycle of iteration because the first iteration was made at Step 1 without corrections). Every time we create a new cost correction and store the previous. 5. Reset the matrix $D_{i,j}$ to zero values.

6. Carry out steps 2–5 until the iterations count reaches N_1.

7. Retain the result with the best cost.

4. Experiment

The problem considered is brand new (to visit the sets subject to precedence constraints, with the cost depending on the sequence of visits). Therefore, it is not altogether valid to compare it to existing algorithms that solve routing problems, which is why we only cite several papers.

The calculations were made on a computer with the Intel i7-2630QM processor, 8GB of RAM in the environment of a Windows 7 (64-bit) operating system.

Example1. Parameters: $N = 15$, $|Z| = 6$, $N_1 = 100$, $N_2 = 10$.

The contours finish cut area length is 150 mm, the width is 50 mm. For small parts, these numbers may be reduced.

There is a lot of problem data concerning the coordinates of the points and the resulting route. We omit it for the sake of space.

The computation lasted 1 min. 35 sec, and the final cost was 52.51. The route is shown in the figure 5.
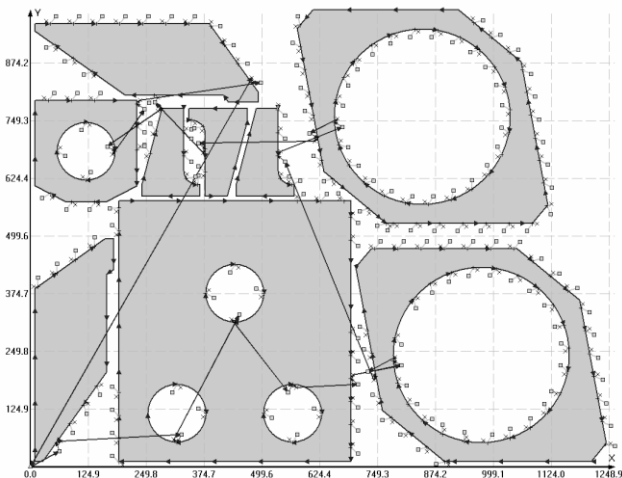

Fig. 5. Solution of Example 1

Example2. Parameters: $N = 121$, $|Z| = 106$, $N_1 = 100$, $N_2 = 10$.

The contours finish cut area length is 100 mm., the width is 25 mm.

The computation lasted 3 min. 15 sec. The cost was found to be 541.20. The route is shown in the Fig. 6.
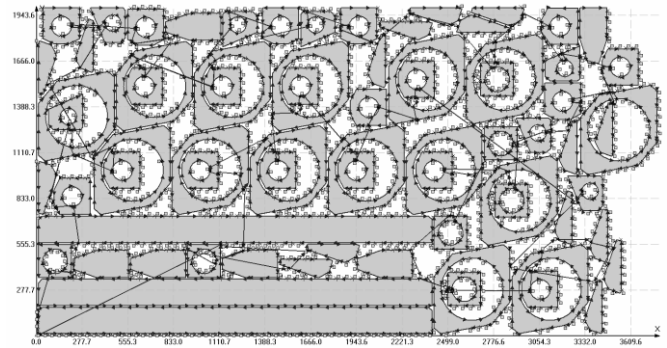

Fig. 6. Solution of Sample 2

5. Conclusions

We considered a routing problem with precedenc, thermal and geometrical constraints, which we solved with an iterative algorithm.

The obtained route looks good in terms of geometrical and thermal restrictions (with absolute compliance of precedence constraints).

Our solution method readily tackles the problem sizes characteristic of real-life industrial problems, which paves the way for applications in the industry.

REFERENCES

Castelino, K., D'Souza, R., Wright, P. (2003). Toolpath optimization for minimizing airtime during machining. *Journal of Manufacturing Systems* 22 (3), 173-180.

Chentsov, A.G. (2013) About routing of robots complexes. *UdGU Bulletin. Mathematics. Mechanics. Computer sciences*, Release 1, pp 59-82.

Chentsov, A.G., Chentsov, A.A. (2013) Dynamic programming in a problem of routing with restrictions and the costs depending on the list of tasks. *Reports of the Russian Academy of Sciences*, T.453, N 1, pp. 20-23.

Chentsov, P.A. (2014) About one application of greedy algorithm for the solution of the generalized salesman problem. *Proceedings of the 16h International Workshop on Computer Science and Information Technologies*. Sheffield. England. Vol.2. pp.148-151.

Dewil, R., Vansteenwegen, P., Cattrysse, D. (2011) Cutting Path Optimization Using Tabu Search. *Key Engineering Materials,* 473, 739-748.

Fischetti, M., Salazar-Gonzalez, J.J., and Toth, P. (1997) A Branch-and-Cut algorithm for the symmetric generalized traveling salesman problem. *Operations Research 45 (3)*, pp. 378-394.

Gutin, G., Karapetyan, D. (2010) A Memetic Algorithm for the Generalized Traveling Salesman Problem. *Natural Computing*, 9(1), Springer, pp. 47-60.

Petunin, A.A. (2009) About some strategy of formation of a route of the cutting tool by development of the controlling programs for the thermal sheet cutting machines. *The UGATU Bulletin. Series: Control, ADP equipment and informatics*, T. 13, No. 2 (35), pp. 280-286.

Petunin, A.A. (2011) Development of CAM-system for sheet cutting machines as an innovation example. *Innovative information technologies: Theory and practice. International scientific edition: materials of the International workshop* (Karlsruhe – Ufa – Dresden, April 8-13, 2011), Ufa, pp. 47-50.

Petunin, A.A. (2012) About optimization models and algorithms of the solution of tasks of design of controlling programs for the sheet cutting machines. *Information technologies and Systems. Scientific pub.: materials of the 1st international scientific ITIS-2012 conference*, Chelyabinsk: ChelGU.

Petunin, A.A., Chentsov A.G., Chentsov P.A. (2013) About routing of tool motion in the sheet cutting machines with numerical program control. *Scientific and technical sheets SPbGPU. Informatics. Telecommunications. Management*, N 2 (169), pp. 103-111.

Salii, Y.V. (2014) On the effect of precedence constraints on computational complexity of dynamic programming method for routing problems. *Vestnik Udmurtskogo Universiteta. Matematika. Mekhanika. Komp'yuternye Nauki,* N 1, pp. 76–86.Silberholz, J., Golden, B. (2007) The Generalized Traveling Salesman Problem: a new Genetic Algorithm approach. E*xtending the Horizons: Advances in Computing, Optimization, and Decision Technologies*, pp. 165−181.

Snyder, L.V., Daskin, M.S. (2006) A random-key genetic algorithm for the generalized traveling salesman problem. *European Journal of Operational Research*, 174, pp. 38−53.

Yang, W. B., Zhao, Y. W., Jie, J., Wang, W. L. (2010). An Effective Algorithm for Tool-Path Airtime. Optimization during Leather Cutting. *Advanced Materials Research*. 102, 373-377.