Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 52-13 (2019) 2320–2325

On the new Algorithm for Solving Continuous Cutting Problem

Petunin A.A *,** **Polishchuk E.G.** * **Ukolov S.S** *

* *Ural Federal University, Yekaterinburg, Russia*
** *Institute of Mathematics and Mechanics, UBr RAS, Yekaterinburg, Russia*

AbstractThe problem of tool path optimization for CNC (Computer Numerical Control) thermal cutting machines as well as its classification are considered. A new approach to solving Continuous Cutting Problem is proposed based on local optimization using Fermat principle and Variable Neighborhood Search approach, which automatically takes Precedence Constraint into account, that is crucial for generating correct CNC programs due to technological features of modern CNC cutting equipment.

Keywords: CNC machine; thermal cutting; tool path optimization; CCP; GTSP; Fermat principle; Variable Neighborhood Search

1. INTRODUCTION

The problem of cutting tool path optimization is one of the applied optimization problems arising in the design of control programs for CNC (Computer Numerical Control) plate cutting machines, refer to Dewil et al. (2016). The cost or time spent is typically used as objective function to optimize, see Tavaeva and Kurennov (2015). The NC program is generated by special software (Computer-Aided Manufacturing, CAM-system) just after another well-known optimization problem has been solved, i.e. the problem of nesting (optimal placement of parts on the plate), see Huang et al. (2009), Sherif et al. (2014). The task is to minimize the consumption of sheet material to produce the parts of known shapes, sizes and quantity.

The process of figure plate cutting with a CNC machine includes the following components:

- Pierce point (cutting start)
- Actual cutting
- Turning the cutter off (cutting end)
- Linear movement from end of cut to start of next cut (air move)

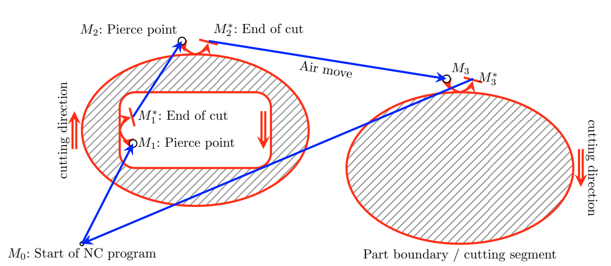

Figure 1. Example of cutting tool path for two parts with three pierce points

NC program usually starts with an air move from some starting point. Fig. 1 represents the scheme of cutting two parts with three pierce points. It shows tool path going along the boundary contours of parts, while in real life it should be offset to a half cut width in order to preserve parts geometry and sizes. From the other hand, most CAM software assume the cutting path goes along parts boundaries, and actual shift is calculated by CNC machine itself or special postprocessing software while converting tool path from CAM internal representation to specific CNC machine language. just before cutting process. We will further assume (if not specified otherwise) that cutting head path is programmed directly along bounding contours of the parts to cut.

2. FORMALIZATION

Let us introduce some notation for tool path components.

We denote $A_1, A_2, \dots A_n$ – two-dimensional shapes (i.e. the parts to cut) that are simple- or multiple-connected regions of Euclidean plane $\mathbb{R}^2$. They are bounded by one or several closed curves (bounding contours) $C_1, C_2, \dots C_N$, $A_i, C_j \subset \mathbb{R}^2, i \in \overline{1,n}, j \in \overline{1,N}, n \leqslant N$).

Also, define plane area $B \subset \mathbb{R}^2$, the model of sheet material, from which parts are to be cut out. We will suppose the nesting (position of the parts within B) somehow fixed already, meeting the condition of mutual non-intersection. Some other requirements may also arise due to specific technological features of CNC equipment used. Any way, fixed disposition of $A_1, A_2, \dots A_n \subset B$ is available to us.

Then in accordance with Petunin (2015) we denote $S = MM^*$ – a cutting segment, i.e. the path of cutting head from pierce point M to the corresponding cut end point M^*. In terms of geometry, cutting segment is a curve at Euclidean plane: $S \subset \mathbb{R}^2, M(x,y) \in \mathbb{R}^2, M^*(x^*,y^*) \in \mathbb{R}^2$.

10.1016/j.ifacol.2019.11.552

We suppose the cut direction is defined in every point of segment S.

Using the notion of cutting segment, all the CNC machine cutting technics can be classified as:

(1) *Closed contour cutting (standard)*: each segment contains exactly one part contour that is cut from start to end
(2) *Multi-segment cutting*: part contour consists of two (or more) cutting segments
(3) *Multi-contour cutting*: several part contours are cut at once within one cutting segment

Let's suppose that our parts $(A_1, A_2, \dots A_n)$ were cut using K cutting segments $S_k = M_k M_k^*, k \in \overline{1,K}$. Then we define cutting path as tuple

$$ROUTE = \langle M_0, M_1, S_1, M_1^*, M_2, S_2, M_2^*, \dots M_K, S_K, M_K^*, i_1, i_2, \dots, i_K \rangle \quad (1)$$

Here $i_1, i_2, \dots, i_K$ is the sequence of segments $S_1, S_2, \dots S_K$ processing, M_0 – starting point of NC program. Linear air moves are implicitly defined by the tuple. In terms of discrete mathematics, the sequence is a permutation, i.e. an ordered set of natural numbers $1 \dots K$, that unambiguously maps every $k \in \overline{1,K}$ to i_k. As noted before, we assume tool path to go along bounding contours of parts and cutting segments to contain all of them:

$$\bigcup_{j=1}^{N} C_j \subset \bigcup_{k=1}^{K} S_k$$

Modifying cutting path one can drastically change numerical parameters of cutting process. Therefore a number of optimization problems arise while developing NC programs for CNC cutting machines, see Makarovskikh et al. (2018). The most popular objective function is total cutting time. In case of thermal or hydrobarasive cutting the following formula applies:

$$T_{cut} = \frac{L_{on}}{V_{on}} + \frac{L_{off}}{V_{off}} + N_{pt} \cdot t_{pt} \quad (2)$$

Here, L_{off} is total length of air moves, L_{on} is total length of cutting segments, V_{off} is the speed of air move, V_{on} is the speed of cutting, N_{pt} is the number of pierce points, t_{pt} is the time spent in a pierce point, assuming it is located in the body of sheet. In some scenarios other kinds of pierce points can be used with their own t_{pt} constants. In that case, equation (2) transforms to

$$T_{cut} = \frac{L_{on}}{V_{on}} + \frac{L_{off}}{V_{off}} + \sum_{j=1}^{p} N_{pt}^j \cdot t_{pt}^j \quad (3)$$

Here, p is a number of pierce point kinds, N_{pt}^j – a number of pierce points of kind j, t_{pt}^j – time to pierce point of kind j. Both in (2) and (3), V_{off} is a constant dependent on CNC machine used. Value of V_{on} is to be determined along with NC program development concerning cutting technology to use and parameters of sheet material (including its thickness). Thus defined V_{on} is often assumed to be a constant value, but in practice actual cutting speed depends on many technological features as well as NC program itself. Additional research is required in this area, some results can be found in Tavaeva and Petunin (2017), Tavaeva and Petunin (2015), but this is beyond the scope of this article.

Total cost of cutting is another very important objective function, especially from economical point of view. This is rather complex indicator dependent on electric power consumption, expendables, CNC machine maintenance as well as other operational expenses.

Note that in general cutting cost is not proportional to cutting time, since it also depends on cutting modes. It can be estimated like in (2):

$$F_{cost} = L_{on} \cdot C_{on} + L_{off} \cdot C_{off} + N_{pt} \cdot C_{pt} \quad (4)$$

Where C_{on} is the unit cost of cutting, C_{off} is that of air move and C_{pt} is the cost of a piercing a point, while L_{on}, L_{off} and N_{pt} make the same sense as in (2). Exact values of unit costs C_{on}, C_{off} and C_{pt} depend on CNC equipment, cutting technology, sheet material and its thickness. This dependency is usually summarized in tabular form or even analytically.

In its turn, the task of finding appropriate unit costs C_{on}, C_{off} and C_{pt} for specific equipment and material is low-studied itself, refer to Tavaeva and Kurennov (2015), Tavaeva and Petunin (2014).

In any case, the value of above objective function depends entirely on the tuple (1) elements, i.e. the cutting path. In detail, the geometry of cutting segments $S_1, S_2, \dots S_K$ gives us total cutting length L_{on}, while points $M_0, M_1, M_1^*, \dots M_K, M_K^*$ with permutation $i_1, i_2, \dots i_K$ describe air moves hence their total length L_{off}.

Therefore, mentioned above problems of cutting path optimizations may be stated as minimization of some objective function F, defined on set G of admissible tuples (1):

$$F(ROUTE) \to \min_{ROUTE \in G} \quad (5)$$

As tuple contains both cutting sequence $i_1, i_2, \dots i_K$ and cutting start and end points $M_k M_k^*, k \in \overline{1,K}$, the latter located on the Euclidean plane $\mathbb{R}^2$, general optimization problem (5) is a very complex continuous-discrete optimization problem, even with significant restrictions imposed on admissible segments $S_1, S_2, \dots S_K$.

This leads to the fact that there are no general algorithms for solving the problem (5) described in the scientific literature. Nevertheless, some narrow classes of problems exist, that allow efficient optimization algorithms to be developed, see Dewil et al. (2012). We mention four such classes:

(1) **Continuous Cutting Problem (CCP)**: the cutter head visits each contour to be cut once. The tool can engage the contour at any point on its perimeter, but must cut the entire contour before it travels to the next contour. Accordingly, the same point must be used for entry and exit the contour. In terms of cutting techniques CCP implies the standard one. With no constraints on the pierce points, problem (2) reduces to finding minimal air move distance L_{off}, which is equivalent to the classical metric TSP (Travelling Salesman Problem). A significant part of the publications is devoted precisely to the solution of this particular optimization problem
(2) **Endpoint Cutting Problem (ECP)**: the tool can enter and exit contours only at some predefined points

on the boundary. However, it may cut the contour in sections, or stated otherwise: a contour can be pre-empted. This approach reduces the complexity of full problem by discretization of possible pierce point set and therefore contour entry points as well. Set of contour exit points is equal to that of entry points, however, contour exit point is not necessary a point of cutting end (where cutting head is switched off), because cutting can continue to another contour.

(3) **Generalized Traveling Salesman Problem (GTSP)**: the tool path visits each contour to be cut once and the tool can engage the contour only at some predefined points on the boundary. This is a special case for CCP and ECP problems.

(4) **Segment Continuous Cutting Problem (SCCP)**: cutter head visits each segment to be cut once. The tool can engage the segment at any point, but must cut the entire segment before it travels to the next segment, see Petunin (2015).

Note: $CCP \subset SCCP$.

This case uses the concept of basic cutting segment B^S, the part of full cutting segment $S = MM^*$ with lead-in and lead-out excluded. Unlike full cutting segments, basic ones doesn't assume cutting direction, containing only geometrical data. When standard cutting technique is used only, the set of basic segments $\bigcup B^S = \bigcup C_i$ (the set of bounding contours).

First classification of tool path routing problems was proposed in: Hoeft and Palekar (1997).

Most publications on the area explore discrete optimization problems (primarily, GTSP and ECP). In contrast to the well-known mathematical models of GTSP, these studies take into account additional technological constraints (see Petunin et al. (2016) for extensive discussion), in particular, the *Precedence Constraint*, refer to Chentsov and Chentsov (2013).

It imposes restrictions on the order of segment processing $I = (i_1, i_2, \ldots i_K)$. Modern CNC cutting equipment cannot guarantee positioning of cutting head inside the contour that is already cut, because the part can freely move (or even drop) after it is detached from the sheet. To cope with this, the following rules must be obeyed:

(1) If one or more inner contours (holes) are inside outer one, then they all must be cut before cutting of outer contour starts
(2) If some other part is placed inside inner contour of a part, the former one must be cut before the latter. Rule 1 applies either.

These rules are the precedence constraint for the permutation $I = (i_1, i_2, \ldots i_K)$. In other words:

(1) If permutation $I = (i_1, \ldots i_k, \ldots i_K)$ contains outer contour i_k, all its corresponding inner contours i_x must appear before i_k.
(2) If permutation $I = (i_1, \ldots i_k, \ldots i_K)$ contains inner contour i_k and there is some outer contour of another part $A_l, l \in \overline{1,n}$ inside it, that outer contour must precede i_k in permutation I

Both the precedence constraint and pierce point constraints are of static nature, that is they are completely defined by the nesting the parts on the sheet, CNC equipment and material used and are not affected by the elements of tuple itself. In terms of tool path (1), some of permutations $I = (i_1, i_2, \ldots i_K)$ are just forbidden. Of many recent papers on GTSP cutting optimization algorithms, we note Vicencio et al. (2014), Dewil et al. (2014), Yu and Lu (2014), Yang et al. (2010), DSouza and Wright (2003).

A number of technological features of modern CNC equipment impose a few other constraints on pierce points and segment cutting order. But unlike mentioned above, these constraints are dynamic, they depend on the preceding elements of tuple (1) and caused by thermal deformations of sheet material. It is rather difficult to get comprehensive equation for such constraints, see Chentsov and Chentsov (2013); Chentsov et al. (2015) for approaches. These constraints, however, are also beyond the scope of the paper.

3. TWO-STAGE RELAXATION FOR CCP

We will further investigate a problem that goes beyond discrete optimization only – Continuous Cutting Problem aka CCP.

As stated above, we assume cutting goes along part boundaries $S_i = C_i, \forall i = \overline{1,k}$ and piercing point M_i is always equal to cutter off point, so as both lay on the contour: $M_i \equiv M_i^* \in C_i, \forall i = \overline{1,k}$.

Tool path starting point M_0 is also fixed.

Finally, to suit most modern CNC cutting equipment, every closed contour $C_i \equiv S_i$ is assumed to be a finite set of circular arcs and straight line segments. It is tempting to reduce CCP to Touring Polygons Problem (see Dror (1999) and Qin and Yuan (2017)), but we consider more general problem.

In our notations Continuous Cutting Problem is to find:

(1) Positions of k piercing points $M_i \in C_i$
(2) Order of contours C_i processing, i.e. the permutation of k items $I = (i_1, i_2, \ldots, i_k), i_j \in \overline{1,k}, \forall j \in \overline{1,k}$

Objective function of cutting time (2) in our case can be simplified to total length of air movements

$$\mathcal{L} = \sum_{j=0}^{k} |\overrightarrow{M_{i_j} M_{i_{j+1}}}|$$
$$\mathcal{L} \to \min$$

where we suppose $M_{i_0} = M_{i_{k+1}} = M_0$ for the brevity sake.

Since Arkin and Hassin (1996) *CCP* is also referred to as *TSP-N* (Travelling Salesman Problem with Neighborhoods). This problem is often reduced to GTSP (Generalized Traveling Salesman Problem) by means of discretization of contours C_i with some step ε. Finding piercing point M_i becomes a discrete optimization problem of choosing among finite number of candidates, refer to Chentsov et al. (2016). But we are to solve problem of both discrete and continuous optimization, using two-step process (see Lee and Kwon (2006) for alternative application of these idea).

In addition, the *Precedence Constraint* is considered. For CCP we can state it in a simpler form: when some contour

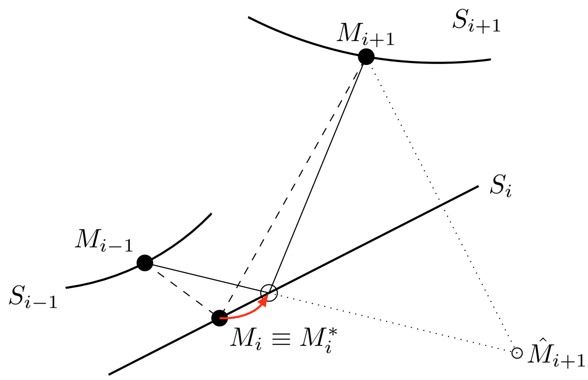

Figure 2. Applying Fermat principle to find piercing point

is inside another one, the former must be cut before the latter

$$\tilde{C}_a \subset \tilde{C}_b \Rightarrow i_a < i_b$$

where we denote $\tilde{C}_i$ a 2D figure, bounded by C_i.

Thus stated Precedence Constraint limits the feasible subset of permutations $I = (i_1, i_2, \ldots, i_k)$, that are the part of complete problem solution (1).

3.1 Continuous optimization

In practice, the shape of part bounding contours is limited to a finite set of straight line segments and circular arcs. This allows exact solution of subproblem of optimal positioning of single pierce point M_i, while others are fixed as well as processing order $I = (i_1, i_2, \ldots, i_k)$.

$$\mathcal{L} \to \min_{M_i}$$

which in turn means

$$\sqrt{\overrightarrow{M_i M_{i-1}}^2} + \sqrt{\overrightarrow{M_i M_{i+1}}^2} \to \min_{M_i \in C_i}$$

Simple consideration reveals, that if neighbor pierce point (assumed to be fixed) lay on both sides of cutting segment (i.e. line or circle containing the latter), optimal position is an intersection of line $M_{i-1}M_{i+1}$ with segment itself

$$M_i = M_{i-1}M_{i+1} \cap S_i$$

When neighbors are situated on the same side, famous *Fermat principle* should be applied, leading to angle of incidence be equal to the angle of reflection, as seen on Fig. 2.

Similar to *Alternating Least Squares* (ALS) method, the complete process for positioning all the pierce points $M_i, i \in \overline{1,k}$ could be called *Alternating Fermat Principle* and organized as follow:

- Randomly choose initial position $M_i \in C_i, \forall i$
- $\forall i \in \overline{1,k}$ find optimal M_i as stated above in constant time $O(1)$
- Repeat until all pierce point position $M_i, \forall i$ converge (with some predefined accuracy δ)

In practice this process also converges well in $O(k)$ time, that is why we use it as one step for complete procedure of continuous and discrete optimization.

3.2 Discrete optimization

Discrete optimization problem to be solved inside CCP is to find permutation $I = (i_1, i_2, \ldots, i_k)$, minimizing total air move length

$$\mathcal{L} \to \min_{i_1, i_2, \ldots i_k}$$

Variable Neighborhood Search (VNS) over full set of permutations is used, see Hansen et al. (2010).

The general scheme of VNS as applied to CCP:

(1) $I \leftarrow randomP_k$
(2) $k \leftarrow 1$
(3) **while** $k < k_{max}$
  (a) $I' \leftarrow \arg\min_{I' \in \mathcal{N}^k(I)} \mathcal{L}(I')$
  (b) **if** $\mathcal{L}(I') < \mathcal{L}(I)$
    • $I \leftarrow I'$
    • $k \leftarrow 1$
  (c) **else**
    • $k \leftarrow k + 1$
(4) **end**

Step (a) repeatedly applies above described continuous optimization, actually

$$\mathcal{L}(I') = \min_{M_1, M_2, \ldots M_k} \mathcal{L}(M_1, M_2, \ldots M_k | I')$$

To build neighborhoods $\mathcal{N}^k(I)$ of different structures, various techniques are used. Only a few of them can be expressed in terms of popular metrics.

- All pairwise permutations of the current one $I = (i_1, i_2, \ldots, i_k)$. This neighborhood can be generated using Levenshtein distance
- Permutations of three items (contours to cut). Enumeration of all triple permutations requires $O(k^3)$ time, so it was restricted by permutations with elements whose pairwise distance is below some value (algorithm parameter)
- In a similar manner, quadruple permutations are considered, limited to some pairwise distance
- Cyclic permutations inside a sequential block of arbitrary length
- Reversing sequential block of arbitrary length
- Permutations of two sequential blocks of the same length
- Cyclic shift of a chain of blocks of the same length
- About dozen other options for constructing a neighborhood of a permutation $\mathcal{N}(I)$

When some technique yields too large neighborhoods, it can be easily limited by simple heuristics, just as the one used for triple and quadruple permutations.

Besides, various options of Variable Neighborhood Search approach may be used, such as *First Improvement* (instead of above described *Best Improvement*) or stochastic point selection. Performance of these options is the subject of further research.

3.3 Precedence constraint

Described steps of the algorithm can be considered as one of the solutions for *Traveling Salesman Problem* (TSP) with quasi-Euclidean city distance.

But cutting tool path optimization must also take into account a number of additional constraints, caused by technological features of CNC equipment. Most popular of them is *Precedence Constraint*, see Veeramani and Kumar (1998) for example.

Luckily, it can be easily accounted inside proposed algorithm. To do so, we start from discarding the contours, that contains others inside:

$$\{C_i | \forall j \neq i : C_j \cap \tilde{C}_i = \varnothing\} \quad (6)$$

For that new set of contours we solve the unconstrained CCP as described above. Note, that resulting air move path

$$\bigcup_{i=0}^{k'} M_i M_{i+1} \quad (7)$$

intersect **all** source contours $C_i, i \in \overline{1,k}$.

Indeed, it intersects the contours in (6) by construction, and the others (outer ones), because its start and finish point $M_0 \equiv M_{k+1}$ is outside all the contours, and every outer contour contains at least one contour (6) inside.

Now, we proceed to the previously skipped contour C_i without pierce point M_i (i.e. it contains some contours in (6) inside). We find all the intersection points $C_i \cap$ (7) and since there are several such points, we select *last* one of them (according to path (7) direction).

Next, we add all new points M_i found to path (7). This is the solution of original CCP, considering the *Precedence Constraint*. Its length obviously remains the same during extra point insertion.

The resulting path contains all k piercing points $M_i, \forall i \in \overline{1,k}$ and point for outer contour is guaranteed to pass after the points for its inner ones.

Further, when solution for partial problem is optimal (according to length of air move), the final solution is optimal either. Indeed, if shorter full path existed, then one could build solution for partial problem by discarding all the contours not in (6) with their piercing points, preserving total path length. But it is supposed impossible for such a solution to exist.

The technique described thus effectively takes *Precedence Constraint* into account while solving CCP. Moreover, it reduces the problem space dimension, leading to lower time requirements.

4. NUMERICAL EXPERIMENTS

From a practical point of view, the proposed algorithm turns out to be very successful and finds easily good solutions despite the lack of mathematical guarantees for this.

In order to confirm this claim a series of experiments were conducted. The results of the proposed algorithm were compared with the ones of another algorithm, which is proved to find the exact solution to the GTSP problem with the number of contours (*megalopolises*) no more than 32, see Chentsov et al. (2018), Petunin et al. (2014).

The comparison results are in the table. 1.

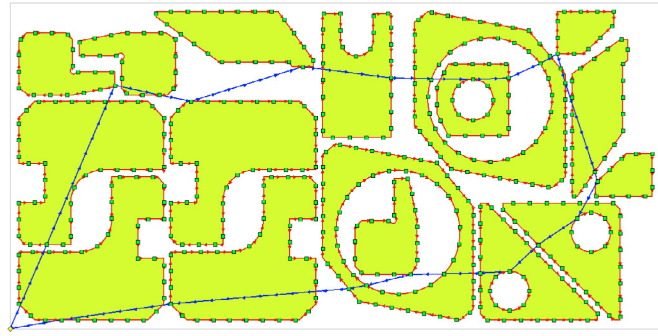

Figure 3. Exact solution of GTSP, Job #3211

Job	#229	#464	#3211
# of parts	11	14	17
# of contours	12	21	22
L_{on}, m	24.609	21.717	25.051
# of GTSP points	491	429	493
GTSP's L_{off}, m	7.729	4.743	4.557
Our L_{off}, m	7.727	4.706	4.536

Table 1. Solution quality comparison

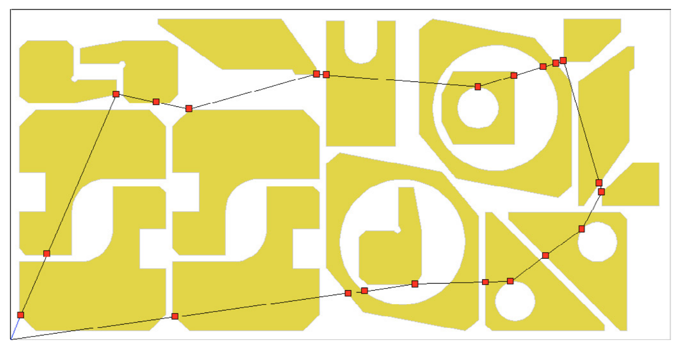

Figure 4. Solution of CCP, Job #3211

The quality of the constructed cutting path can be visually compared on Fig. 3 and Fig. 4.

Values of L_{off} are in good compliance for two algorithms used as well as shapes of the paths themselves.

Two-stage relaxation finds even slightly shorter tool path, because it can route cutting head along straight lines whereas GTSP's tool path is always polygonal due to preliminary discretization involved. In other words, optimal solution for CCP is always a bit shorter than that of GTSP.

5. CONCLUSION

The new heuristic algorithm for solving Continuous Cutting Problem for CNC cutting machine is described and evaluated.

In the future, the developed algorithm is planned to be modified so that all the piercing points meet the technological requirements of thermal cutting, described in Petunin and Stylios (2016).

ACKNOWLEDGEMENTS

The work was supported by Act 211 Government of the Russian Federation, contract №02.A03.21.0006, and by Ministry of Education and Science of the Russian Federation, project №2.9563.2017/8.9

REFERENCES

Arkin, E.M. and Hassin, R. (1996). Approximation algorithms for the geometric covering salesman problem. *Location Science*, 4(4), 281.

Chentsov, A.G., Chentsov, P.A., Petunin, A.A., and Sesekin, A.N. (2018). Model of megalopolises in the tool path optimisation for cnc plate cutting machines. *International Journal of Production Research*, 56(14), 4819–4830. doi:10.1080/00207543.2017.1421784.

Chentsov, A.G., Khachai, M.Yu., and Khachai, D.M. (2016). An exact algorithm with linear complexity for a problem of visiting megalopolises. *Proceedings of the Steklov Institute of Mathematics*, 295(1), 38–46. doi: 10.1134/S0081543816090054.

Chentsov, A. and Chentsov, A. (2013). Dynamic programming in the routing problem with constraints and costs depending on a list of tasks. In *Doklady Mathematics*, volume 88, 637–640. Springer.

Chentsov, A.G. et al. (2015). A model of "nonadditive" routing problem where the costs depend on the set of pending tasks. *Vestnik YuUrGU*, 8(1).

Dewil, R., Vansteenwegen, P., and Cattrysse, D. (2012). Heuristics for laser cutting tool path generation. In *Applied Mathematical Optimization and Modelling*, 10.

Dewil, R., Vansteenwegen, P., and Cattrysse, D. (2014). Construction heuristics for generating tool paths for laser cutters. *International Journal of Production Research*, 52(20), 5965–5984.

Dewil, R., Vansteenwegen, P., and Cattrysse, D. (2016). A review of cutting path algorithms for laser cutters. *International Journal of Advanced Manufacturing Technology*, 87(5), 1865–1884. doi:10.1007/s00170-016-8609-1.

Dror, M. (1999). Polygon plate-cutting with a given order. *IIE Transactions*, 31(3), 271–274. doi: 10.1023/A:1007606007369.

DSouza, K.C.R. and Wright, P.K. (2003). Tool-path optimization for minimizing airtime during machining. *Journal of Manufacturing Systems*, 22, 173–180.

Hansen, P., Mladenović, N., and Moreno Pérez, J.A. (2010). Variable neighbourhood search: methods and applications. *Annals of Operations Research*, 175(1), 367–407. doi:10.1007/s10479-009-0657-6.

Hoeft, J. and Palekar, U.S. (1997). Heuristics for the plate-cutting traveling salesman problem. *IIE Transactions*, 29(9), 719–731. doi:10.1023/A:1018582320737.

Huang, X., Xi, F., Li, J., and Zhong, Z. (2009). Optimal layout and path planning for flame cutting of sheet metals. *International Journal of Computer Integrated Manufacturing*, 22(1), 30–41.

Lee, M.K. and Kwon, K.B. (2006). Cutting path optimization in CNC cutting processes using a two-step genetic algorithm. *International Journal of Production Research*, 44(24), 5307–5326. doi: 10.1080/00207540600579615.

Makarovskikh, T.A., Panyukov, A.V., and Savitskiy, E.A. (2018). Mathematical models and routing algorithms for economical cutting tool paths. *International Journal of Production Research*, 56(3), 1171–1188. doi: 10.1080/00207543.2017.1401746.

Petunin, A.A., Polishuk, E.G., Chentsov, A.G., Chentsov, P.A., and Ukolov, S.S. (2016). About some types of constraints in problems of routing. *AIP Conference Proceedings*, 1789. doi:10.1063/1.4968494.

Petunin, A.A. (2015). Modeling of tool path for the cnc sheet cutting machines. *AIP Conference Proceedings*, 1690(1), 060002. doi:10.1063/1.4936740.

Petunin, A.A., Chentsov, A.G., and Chentsov, P.A. (2014). Local dynamic programming incuts in routing problems with restrictions. *Vestnik Udmurtskogo Universiteta. Matematika. Mekhanika. Komp'yuternye Nauki*, (2), 56–75.

Petunin, A.A. and Stylios, C. (2016). Optimization models of tool path problem for cnc sheet metal cutting machines. *IFAC-PapersOnLine*, 49(12), 23–28.

Qin, Y. and Yuan, B. (2017). Aco-irba: A hybrid approach to tspn with overlapping neighborhoods. *SpringerLink*, 87–96. doi:10.1007/978-3-319-68759-9_8.

Sherif, S.U., Jawahar, N., and Balamurali, M. (2014). Sequential optimization approach for nesting and cutting sequence in laser cutting. *Journal of Manufacturing Systems*, 33(4), 624–638.

Tavaeva, A. and Petunin, A. (2014). The issue of cutting cost parameters optimization for cnc laser cutting machines. In *Proc. Information Technologies for Intelligent Decision Making Support*, volume 2, 75–78.

Tavaeva, A. and Petunin, A. (2015). The definition of correction coefficients for speed of working tool path in objective function for problems of tool path optimization. In *Proc. Information Technologies and Systems*, 45–47.

Tavaeva, A.F. and Petunin, A.A. (2017). Investigation of cutting speed influence on optimality of the tool path route for cnc laser cutting machines. *2017 International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM)*, 1–5. doi: 10.1109/ICIEAM.2017.8076452.

Tavaeva, A. and Kurennov, D. (2015). Cost minimizing of cutting process for cnc thermal and water-jet machines. *AIP Conference Proceedings*, 1690(1), 020003. doi: 10.1063/1.4936681.

Veeramani, D. and Kumar, S. (1998). Optimization of the nibbling operation on an NC turret punch press. *International Journal of Production Research*, 36(7), 1901–1916. doi:10.1080/002075498193020.

Vicencio, K., Davis, B., and Gentilini, I. (2014). Multi-goal path planning based on the generalized traveling salesman problem with neighborhoods. *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2985–2990. doi:10.1109/IROS.2014.6942974.

Yang, W.B., Zhao, Y.W., Jie, J., and Wang, W.L. (2010). An Effective Algorithm for Tool-Path Airtime Optimization during Leather Cutting. *Advanced Materials Research*, 102-104, 373–377. doi: 10.4028/www.scientific.net/AMR.102-104.373.

Yu, W. and Lu, L. (2014). A route planning strategy for the automatic garment cutter based on genetic algorithm. In *Evolutionary Computation (CEC), 2014 IEEE Congress on*, 379–386. IEEE.