

Using PCGTSP Algorithm for Solving Generalized Segment Continuous Cutting Problem

Alexander Petunin^{***}, Michael Khachay^{**}, Stanislav Ukolov^{*}, Pavel Chentsov^{**}

^{*}Ural Federal University, Ekaterinburg, Russia

^{**}Krasovsky Institute of Mathematics and Mechanics, Ekaterinburg, Russia

Abstract: The problem of optimal tool routing for CNC sheet cutting machines (referred to as Cutting Path Problem or Tool Path Problem) is considered. The general formulation is used – Generalized Segment Continuous Cutting Problem (GSCCP). The new algorithm developed by the authors to solve generalized traveling salesman problem with precedence constraints (PCGTSP) is shown to effectively tackle this problem. This branch-and-bound algorithm, combined with the use of dynamic programming and a specialized heuristic solver, makes it possible to get optimal solutions for problems of small dimension in a relatively short time compared to known exact algorithms, as well as to find effective lower and upper bounds for the optimal solutions for large-scale problems. The conclusions are illustrated by solving several model examples.

Copyright © 2022 The Authors. This is an open access article under the CC BY-NC-ND license (<https://creativecommons.org/licenses/by-nc-nd/4.0/>)

Keywords: Cutting Path Problem, Optimization, Branch-and-Bound, Dynamic Programming, Generalized Traveling Salesman Problem, Precedence Constraints

1. INTRODUCTION

Let us consider the problem of routing the tool of NC sheet cutting machines, known as the Cutting Path Problem or Tool Path Problem, see Dewil et al. (2015a). This problem occurs during the development of control programs for the NC machine, which specify the tool path and several commands to set the parameters for cutting sheet material to obtain parts of known shapes and sizes. The input data for the tool path design for NC machine is calculated from the information about positions of the parts to cut which in its turn is generated during the solution of the problem of irregular figure cutting of sheet material (aka "nesting"), see Sherif et al. (2014). From the point of view of geometric optimization, this problem belongs to the class of Cutting & Packing problems, see Alvarez-Valdes et al. (2018). Polynomial complexity solving algorithms are not known to exist for those problems, as well as for route optimization ones.

Among modern researchers of the Cutting Path Problem, one should single out R. Dewil and his colleagues, see Dewil et al. (2015a, 2016, 2015b). In their writings, an attempt to link the features of laser cutting with routing algorithms is made. In Dewil et al. (2016) an overview of routing algorithms related to figure sheet cutting on NC machines is given. The authors classify the existing routing literature into six classes of problems: the continuous cutting problem (CCP), the endpoint cutting problem (ECP), the intermittent cutting problem (ICP), the polygon traversal problem (TPP), the traveling salesman problem (TSP), and the generalized traveling salesman problem (GTSP). All the listed classes of problems, except CCP, use discrete mathematical models. The most general case of routing problem is in fact ICP. However, the literature on the latter is very scarce, and most scientific articles are devoted to problems of other classes.

2. GSCCP model

In Petunin (2019), based on the introduced concepts of "cutting segment" and "basic cutting segment", the opportunity was presented to distinguish a fairly wide subclass of tasks in the ICP class, which are reduced to the CCP and GTSP classes. This new class was called Generalized Segment Continuous Cutting Problem (GSCCP). New concept made it possible, in particular, to solve problems of different classes, applying different cutting techniques within the same route. The cutting segment here means the tool path between the pierce point and the corresponding tool off point, and the base segment is the part of the cutting segment without the initial part of the path between the pierce point and the tool entry point into the contour (lead-in), and the end part between the exit point from the contour and the tool turning off point (lead-out). Fig. 1 shows an example of a cutting segment (curve in red) containing tool movement along the boundary contours of three circles.

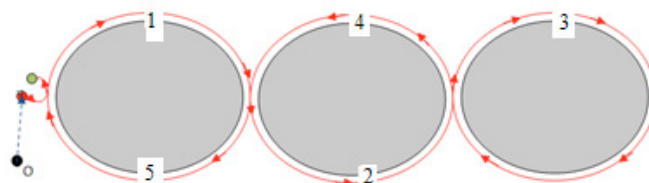


Figure 1. Example of a cutting segment

To cut these circles, you can use the standard cutting technique. It means each contour is always cut in its entirety and the number of pierce points is equal to the number of contours. In this case, the set of basic segments coincides with the set of boundary contours and the toolpath problem belongs to the CCP class. If the number of the possible pierce points for each contour is finite (for example, by discretizing the

contour with some step), then the GTSP model can be used for the CCP problem. Unlike CCP, the GSCCP model assumes that more than one set of base segments is specified for the same nesting result, which does not match the set of boundary contours for the parts.

Note that the problem of toolpath routing for sheet cutting machines involve the imposition of mandatory constraints on the feasible solution, so-called precedence constraints, which are caused by the technological features of sheet cutting. These constraints very often make it possible to significantly reduce the computational complexity of the problem to solve.

2.1. Algorithms used

As far as problems of small dimension are considered, it is possible to use exact algorithms. The algorithm, hereinafter referred to as DP, see Chentsov et al. (2018), uses the Bellman scheme of dynamic programming. The tool path is optimized in reverse order, starting from the end, which allows, among other things, to select the optimal position of the starting point of the tool path. This algorithm usually finds the optimal solution for problem instances with less than 33 contours in a reasonable amount of time. Conventionally, real-life large-scale instances of the GTSP stemming from the industry are solved using heuristic and metaheuristic algorithms, e.g. local-search and memetic algorithms [Karapeyan, Gutin, 2012], the well-known Lin-Kernigan-Helsgaun algorithm [Helsgaun, 2015], efficient implementations of the Adaptive Large Neighborhood Search [Smith and Imeson, 2017], and bio-inspired algorithms [Pintea, 2017]. However, such algorithms do not allow calculating effective estimates for the approximate solution. Along with numerous heuristics proposed for the GTSP, we could recall recent results in the field of polynomial-time algorithms with theoretical accuracy guarantees [Bhattacharya et al, 2015] and approximation schemes [Khachay, Neznakhina, 2017, 2020]. The papers Khachay et al. (2020, 2021) describe new algorithm focused on discrete optimization problems in the form of a generalized traveling salesman problem with precedence constraints (PCGTSP). These algorithm are based on some ideas from Salman et al. (2020), the branch and bound method in combination with the use of the dynamic programming apparatus using (in contrast with DP algorithm above) Held-Karp scheme.

The route is optimized in the forward direction, starting from a fixed initial point. The route is split into prefix and the tail. The traversal order of the former is fixed, so optimal path along the prefix can be easily found with regular dynamic programming approach. The tail is used to construct the instance of PCGTSP problem of the smaller size, and the lower bound for its solution is estimated by triple relaxation: 1) partial elimination of precedence constraint, 2) converting GTSP to regular TSP and 3) converting TSP to Assignment Problem or Minimal Spanning Arborescence Problem, since both can be solved in polynomial time. In some rare cases TSP problem can be solved directly on the step 3.

Having two estimates for the prefix path and all the paths induced by the tail, their sum gives us a lower bound for solution of original PCGTSP problem. This score gradually

improves as the prefix lengthens, so the search tree is traversed in the depth-first order.

For each prefix, its calculated lower bound is compared with a previously known upper bound, and if the former exceeds the latter, the prefix is subjected to a cut and does not contribute to further calculations. For upper bound we use the length / weight of solution of original PCGTSP problem instance with specialized PCGLNS heuristics, see Khachay et al. (2020). It gives high quality solutions in a short time (a few seconds for tens of contours), which allows to discard from 50% to 90% of prefixes.

This new algorithm allows to find optimal solutions of larger scale problems, for example, Fig. 2 shows solution of PCGTSP problem instance of 47 parts, 100 contours and 718 nodes, see Petunin et al. (2021).

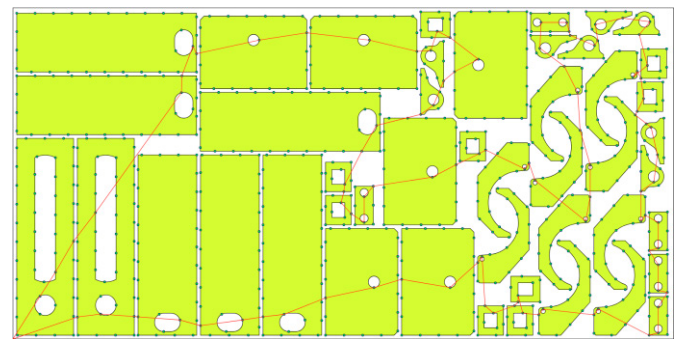


Figure 2. Solution of large scale PCGTSP problem instance

2.2. Reduction of GSCCP problem to PCGTSP instance

These algorithms were intentionally designed to solve GTSP class problems, but a specialized converter developed made it possible to use these algorithms to solve GSCCP segment cutting problems.

The original task of segment cutting is subject to both discrete (sequence of contour processing) and continuous optimization (selection of pierce point on each contour). For algorithms of the Section 2.1 to apply, we reduce continuous optimization to discrete one by choosing a finite number of feasible pierce points on the contour with some predetermined distance. Thus, each cutting segment in the original GSCCP problem is converted into a cluster for the GTSP. Precedence constraints on the order of segment cutting are preserved.

To create alternative sets of cutting segments, we use the bridge technique in this work. Two or more adjacent contours can be combined into one by adding a small jumper traversed by the cutting tool in both directions. This procedure decreases the number of contours / cutting segments and slightly increase length of the cutting path.

2.3. Objective functions

Both algorithms described in Section 2.1 optimize only air move length of tool path L_{off} , that is the length of the broken line connecting all the contours. When applying them to GSCCP problem we must use, however, the full objective function, either full cutting time

$$T_{cut} = \frac{L_{on}}{V_{on}} + \frac{L_{off}}{V_{off}} + T_{pt} \cdot N_{pt}$$

or cutting cost

$$F_{cut} = C_{on} \cdot L_{on} + C_{off} \cdot L_{off} + C_{pt} \cdot N_{pt}$$

where L_{on} is full length of all the contours to cut and N_{pt} is the number of the piercing points, that is the number of contours. Parameters V_{on} , V_{off} , C_{on} , C_{off} , T_{pt} , C_{pt} depend on the CNC equipment used and cutting process selected, see Tavaeva et al. (2019).

Thus, we optimize not just $L_{off} \rightarrow \min$, but linear combination $\alpha \cdot L_{off} + \beta \cdot L_{on} + \gamma \cdot N_{pt} \rightarrow \min$, since L_{on} and N_{pt} are not constants any longer. In fact, L_{on} changes slightly in our experiments, while N_{pt} contributes significantly into the choice of the optimal solution of the GSCCP problem.

In the next section, using a few model examples, we will demonstrate that the new algorithms make it possible to obtain optimal solutions for problems of small dimension in a relatively short time compared to the DP algorithm, and also to find effective lower and upper bounds for the optimal solution for problems of large dimension.

3. NUMERICAL EXPERIMENTS

3.1. Experimental setup

For the purpose of evaluating algorithms performance, the special nesting plan was used, containing 19 plain parts and 24 contours, see Fig. 3. Position of 224 feasible pierce points are also depicted along the contours. They were spaced in 50 mm increments.

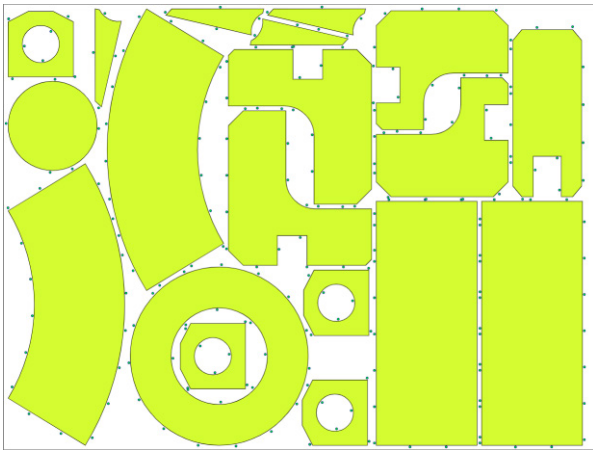


Figure 3. Original nesting plan.

To modify set of cutting segments for the nesting plan, two bridges were added, as seen at Fig. 3, yielding 17 parts bounded by 22 contours. Thus, two new complex parts appeared, marked with figures 1 and 2 at Fig. 4.

Further, another four bridges were created at the original nesting plan, see Fig. 5, yielding 15 parts and 20 contours. Three new complex parts are also marked with figures 1, 2, and 3, where part 1 is identical to that of Fig. 4, while part 2 is slightly bigger.

All three instances were solved to optimality with two algorithms: 1) DP (Dynamic programming) scheme, see Chentsov et al. (2018); 2) new problem-specific Branch-and-Bound algorithm, see. Khachay et al. (2021).

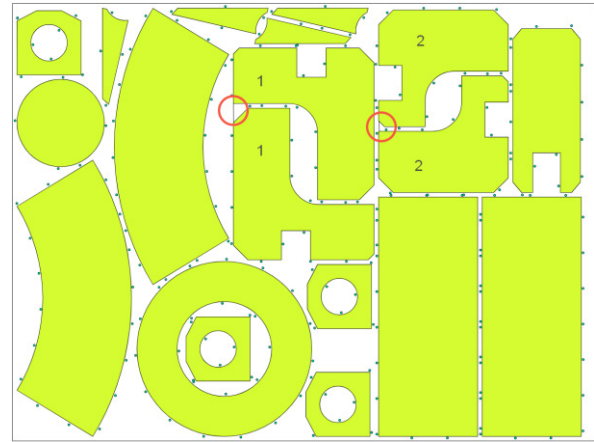


Figure 4. Position of two bridges

Numerical experiments were conducted on the ordinary workstation with Intel Core i5 CPU at 1.60 GHz with 8 Gb of RAM.

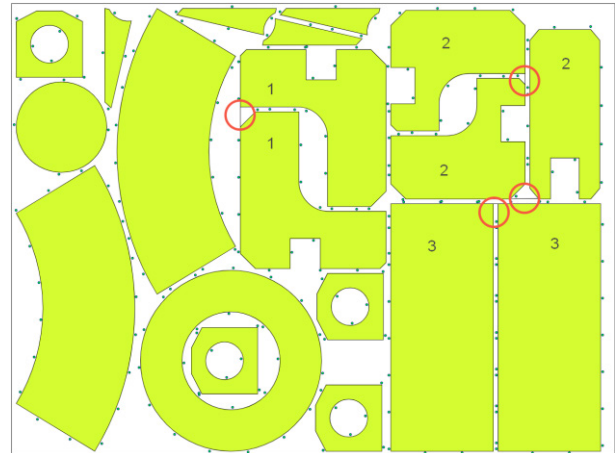


Figure 5. Position of four bridges.

3.2. Results

All the problem instances were successfully solved by both algorithms. Solution of original instance without bridges are on Fig. 6 for DP and Fig. 7 for B-n-B.

Solutions of the second instance of 22 contours and 17 parts are on Fig. 8 and Fig. 9 respectively for DP and B-n-B algorithms. Note the two paths are almost identical in this case.

And finally, solutions of smallest 20 contours instance are at Fig. 10 and Fig. 11. They look even more similar.

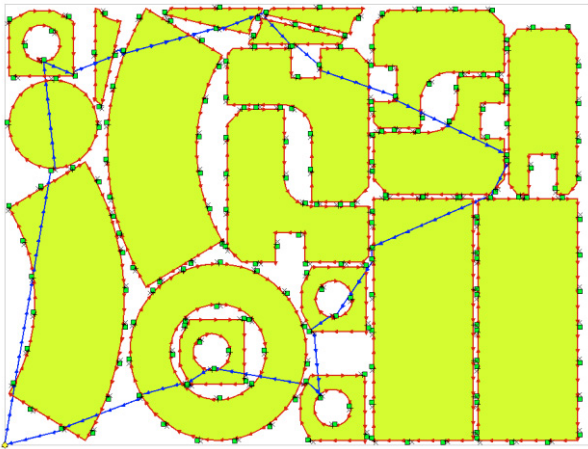


Figure 6. Solution of 24 contours instance with DP.

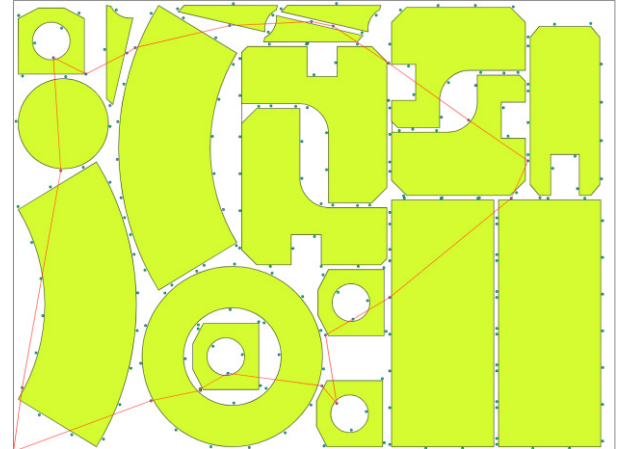


Figure 9. Solution of 22 contours instance with B-n-B.

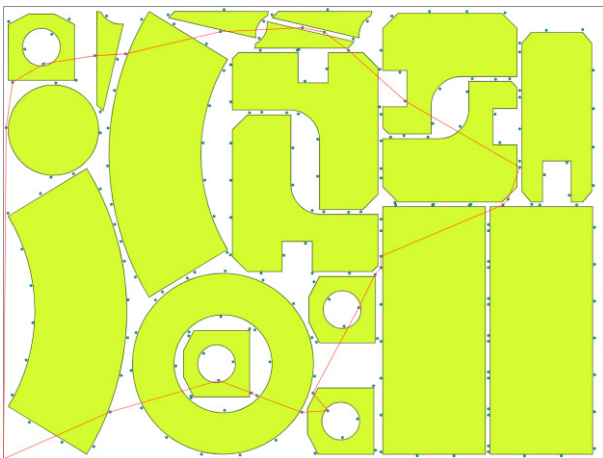


Figure 7. Solution of 24 contours instance with B-n-B.

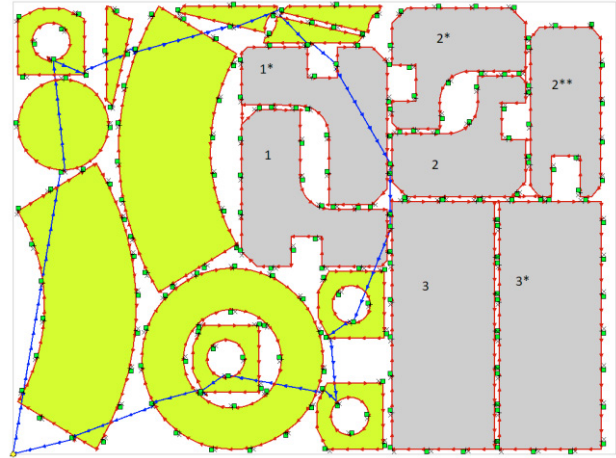


Figure 10. Solution of 20 contours instance with DP.

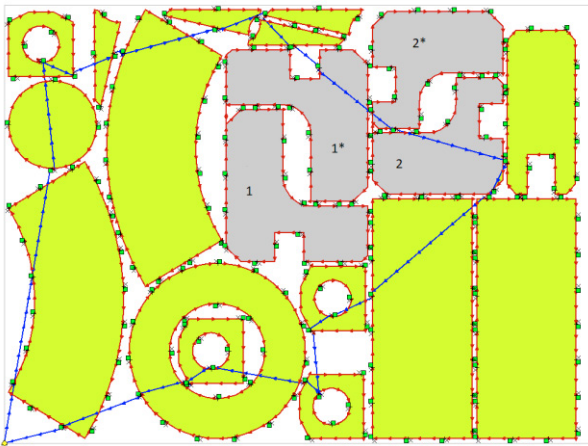


Figure 8. Solution of 22 contours instance with DP.

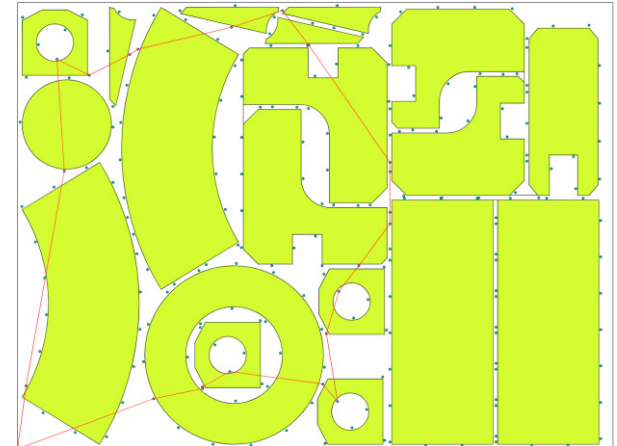


Figure 11. Solution of 20 contours instance with B-n-B.

All the solutions are summarized in Table 1.

Table 1. Solutions

Problem instance		DP			B-n-B	
#C	#P	Calc time (min:s)	Route (mm)	Cut time (s)	Calc time (s)	Route (mm)
24	19	42:12	5340.12 5820.12	168	4.5	5411 5411.16
22	17	4:01	5254.13 5634.13	133	4.1	5380 5382.02
20	15	2:14	4709.12 5109.12	140	3.6	4782 4786.55

For each algorithm and problem instance time is specified in minutes and seconds. For DP solution length two values provided: air move length and full route length without contours, i.e., air move length with lead-in and lead-out. Estimated cutting time in seconds is also calculated. For B-n-B solution, two lengths are integer one (since current implementation of algorithms uses integer calculus) and its exact floating-point value.

3.3. Discussion

It can be easily seen, that both algorithms give solutions very similar to each other, both visually and numerically. The main reason for the slight difference in solutions is that DP algorithm due to its maturity considers the technological constraints of thermal cutting and distinguishes piercing points and tool off points, while B-n-B consider them as one point. This leads to slight decrease in air move length during optimization while simultaneously adding constant lead-in and lead-out distances to resulting toolpath.

Another reason for difference is that current implementation of PCGLNS and B-n-B algorithms uses integer arithmetic, so they often allow several optimal solutions with the same integer length. For example, during numerical experiments another solution for 22 contours case was obtained, slightly different from the one on the Fig. 9.

From the other hand, PCGLNS heuristic offers the great performance, giving high quality solutions in literally seconds. Even in case of hundreds of contours, high quality solutions can be obtained in minutes or tens of minutes, which make it useful in practical application, including development of control program for CNC cutting machines.

From that point of view, an idea of *Segment Cutting* seems very promising. Comparing 20 contours case against original 24 contours one we see not only 11% decrease of route length, but also lower (by 17%) number of piercing points. Both changes reduce time and cost of cutting process.

In contrast with lightning speed of obtaining solution, estimation of its lower bound is rather slow due to exponential time complexity. For example, solution at Fig. 11 (4782 mm) was proven to be optimal in almost 5 hours. This time can be improved both with parallel calculation as well as by

reimplementing B-n-B algorithm using more performant programming language, C++ for instance.

Thus, as can be seen from Table 1, the minimum total cutting time is achieved by optimal tool routing for the set of segments shown in Fig. 8., while the optimal trajectory for the set of segments in Fig. 6 was obtained almost 10 times faster than by the DP method. We also note that solutions for all 3 subtasks of the GSCCP problem (24, 22 and 20 cutting segments), were proven to be optimal ones.

ACKNOWLEDGEMENTS

The work was performed as part of research conducted in the Ural Mathematical Center with the financial support of the Ministry of Science and Higher Education of the Russian Federation (Agreement number 075-02-2022-874).

REFERENCES

- Alvarez-Valdes, R., Carravilla, M.A., and Oliveira, J.F. (2018). Cutting and packing. doi:10.1007/978-3-319-07153-4_43-1.
- Bhattacharya, B., Ćustić, A., Rafiey, A., Rafiey, A., Sokol, V. (2015). Approximation Algorithms for Generalized MST and TSP in Grid Clusters, *Lecture Notes in Computer Science*, vol. 9486, pp. 110–125. Springer International Publishing, Cham. https://doi.org/10.1007/978-3-319-26626-8_9
- Chentsov, A.G., Chentsov, P.A., Petunin, A.A., and Seseikin, 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.
- Dewil, R., Vansteenwegen, P., and Cattrysse, D. (2015a). *Sheet Metal Laser Cutting Tool Path Generation: Dealing with Overlooked Problem Aspects*, volume 639. Trans Tech Publications Ltd. doi: 10.4028/www.scientific.net/KEM.639.517.
- 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.
- Dewil, R., Vansteenwegen, P., Cattrysse, D., Laguna, M., and Vossen, T. (2015b). An improvement heuristic framework for the laser cutting tool path problem. *International Journal of Production Research*, 53(6), 1761–1776. doi:10.1080/00207543.2014.959268.
- Helsgaun, K. (2015). Solving the equality Generalized Traveling Salesman Problem using the Lin–Kernighan–Helsgaun algorithm. *Mathematical Programming Computation*, 7, 269–287, <https://doi.org/10.1007/s12532-015-0080-8>
- Karapetyan, D., & Gutin, G. (2012). Efficient local search algorithms for known and new neighborhoods for the generalized traveling salesman problem. *European*

- Journal of Operational Research*, 219, 234–251. <https://doi.org/10.1016/j.ejor.2012.01.011>
- Khachai, M. Y., & Neznakhina, E. D. (2017). Approximation schemes for the Generalized Traveling Salesman Problem. *Proc. of the Steklov Institute of Mathematics*, 299, 97–105. <https://doi.org/10.1134/S0081543817090127>
- Khachay, M., & Neznakhina, K. (2020). Complexity and approximability of the Euclidean Generalized Traveling Salesman Problem in grid clusters. *Annals of Mathematics and Artificial Intelligence*, 88, 53–69. <http://doi.org/10.1007/s10472-019-09626-w>.
- Khachay, M., Kudriavtsev, A., and Petunin, A. (2020). PCGLNS: A heuristic solver for the Precedence Constrained Generalized Traveling Salesman Problem. In N. Olenov, Y. Evtushenko, M. Khachay, and V. Malkova (eds.), *Optimization and Applications*, volume 12422 of *Lecture Notes in Computer Science*, 196–208. Springer International Publishing, Cham. doi:10.1007/978-3-030-62867-3_15.
- Khachay, M., Ukolov, S., and Petunin, A. (2021). Problem-Specific Branch-and-Bound Algorithms for the Precedence Constrained Generalized Traveling Salesman Problem. In N. Olenov et al. (eds.), *Optimization and Applications*, volume 13078 of *Lecture Notes in Computer Science*, 136–148. Springer Nature Switzerland AG, Cham, Switzerland. doi:10.1007/978-3-030-91059-4_10.
- Petunin, A. (2019). General Model of Tool Path Problem for the CNC Sheet Cutting Machines. *IFAC-PapersOnLine*, 52(13), 2662–2667. doi: 10.1016/j.ifacol.2019.11.609.
- Petunin, A., Khalyavka, A., Khachay, M., Kudriavtsev, A., Chentsov, P., Polishchuk, E., and Ukolov, S. (2021). Library of Sample Image Instances for the Cutting Path Problem. In *Pattern Recognition. ICPR International Workshops and Challenges*, 227–233. Springer, Cham, Switzerland. doi:10.1007/978-3-030-68821-9_21.
- Pintea, CM., Pop, P.C. & Chira, C. (2017). The generalized traveling salesman problem solved with ant algorithms. *Complex Adapt Syst Model* 5, 8. <https://doi.org/10.1186/s40294-017-0048-9>
- Salman, R., Ekstedt, F., and Damaschke, P. (2020). Branch-and-bound for the Precedence Constrained Generalized Traveling Salesman Problem. *Operations Research Letters*, 48(2), 163–166. doi: 10.1016/j.orl.2020.01.009.
- Sherif, S.U., Jawahar, N., & Balamurali, M.N. (2014). Sequential optimization approach for nesting and cutting sequence in laser cutting. *Journal of Manufacturing Systems*, 33, 624–638. doi:10.1016/J.JMSY.2014.05.011.
- Smith, S. L., & Imeson, F. (2017). GLNS: An effective large neighborhood search heuristic for the Generalized Traveling Salesman Problem. *Computers & Operations Research*, 87, 1–19. <https://doi.org/10.1016/j.cor.2017.05.010>
- Tavaeva, A., Petunin, A., Ukolov, S., and Krotov, V. (2019). A Cost Minimizing at Laser Cutting of Sheet Parts on CNC Machines. In *Mathematical Optimization Theory and Operations Research*, 422–437. Springer, Cham, Switzerland. doi:10.1007/978-3-030-33394-2_33