Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 44 (2016) 138 – 143

www.elsevier.com/locate/procedia

6th CIRP Conference on Assembly Technologies and Systems (CATS)

An industrially validated CMM inspection process with sequence constraints

Raad Salman[a,*], Johan S. Carlson[a], Fredrik Ekstedt[a], Domenico Spensieri[a], Johan Torstensson[a], Rikard Söderberg[b]

[a]*Fraunhofer-Chalmers Centre, Chalmers Science Park, SE-412 88 Göteborg, Sweden*
[b]*Wingquist Laboratory, Chalmers University of Technology, 412 96 Gothenburg, Sweden*

[*] Corresponding author. Raad Salman, Tel.: +46-739-837573; *E-mail address:* raad.salman@fcc.chalmers.se

Abstract

An efficient CMM inspection process implemented in industry gives significant productivity improvements. A key part of this improvement is the optimization of the inspection sequences. To ensure quality of the inspection the sequences are often constrained with respect to the order of the measurements. This gives rise to so called precedence constraints when modelling the inspection sequence as a variation of the travelling salesperson problem (TSP). Two heuristic solution approaches and a generic optimizing algorithm are considered. A generation based stochastic algorithm is found to reduce cycle time by as much as 12% in comparison to the currently used algorithm.

Peer-review under responsibility of the organizing committee of the 6th CIRP Conference on Assembly Technologies and Systems (CATS)
Keywords: CMM; inspection; automation; task sequences; precedence constraints; generalized travelling salesman problem; sequential ordering problem

1. Introduction

Many products such as car and truck bodies, engines, medical prosthesis, mobile phones, and lumbering equipment depend visually and functionally on its geometry. Since variation is inherent in all production processes, consistent efforts in styling, design, verification and production aiming at less geometrical variation in assembled products, is a key to shortening development time of new products, as well as for choosing an efficient and resource-economic production process. The activities aiming at controlling geometrical variation throughout the whole product realization process are called the geometry assurance process. Figure 1 shows a general model for product realization consisting of a concept phase, a verification phase and a production phase.

The geometry assurance process, as defined in [1], relies on inspection data in all phases. Product concepts are analyzed and optimized to withstand the effect of manufacturing variation and tested virtually against available production data often based on carry over type of inspection. In the verification and pre-production phase the product and the production system is physically tested and verified. Adjustments are made to both product and production system based on inspection data. In full production the focus is to control the process and to detect and correct errors by analyzing inspection data. These inspection data are often collected before, during and after important assembly steps. In this way, important assembly issues as part, fixture and joining errors can be detected and corrected in an efficient manner.

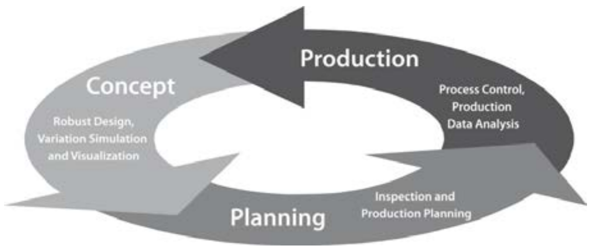

Fig. 1. A general model for product realization and the main activities of the geometry assurance process.

Therefore, the inspection preparation and measuring is an important activity and this paper presents an industrial validated closed loop from inspection preparation to automatic efficient off-line programming of automated measurement equipment. Then the focus is on improving the sequence optimization part of it by solving precedence constrained generalized travelling salesperson problem.

Peer-review under responsibility of the organizing committee of the 6th CIRP Conference on Assembly Technologies and Systems (CATS)
doi:10.1016/j.procir.2016.02.136

2. An Efficient Process for Inspection Preparation and Programming

The efficient inspection process implemented to support programming of automated inspection devices is built up by five main steps; (i) define the inspection task by breaking down product and process requirements to geometrical inspection features, e.g. a hole or a slot, on part and subassembly level (Figure 2), (ii) create parameterized inspection rules that define how a feature should be measured, i.e. number of points, distribution, coordinate system, and probe cones, (iii) perform feature accessibility analysis to find a set of probe configurations of minimum size that can reach all inspection points with collision free CMM configurations (Figure 3), (iv) plan by math based algorithms for motion planning and combinatorial optimization the collision free motions and sequence of the measurement equipment to visit each feature, and (v) generate the control code, e.g. DMIS to instruct the equipment to perform the actual measurement.

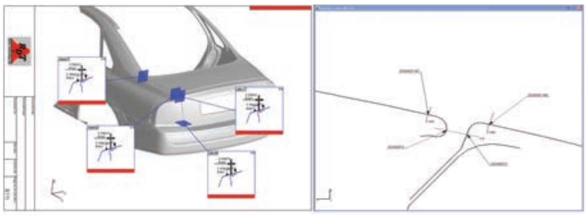

Fig. 2. An inspection task is defined by breaking down the product quality appearance requirement (right picture) on gap and flushes to boot and rear fender part inspection points (right picture).

This process has been industrially evaluated and used by e.g. Volvo Cars to program all automated inspection devices since 2011. The results show an improvement in inspection preparation time of 75% and productive increase in equipment utilization of 25%. The experience is also that the inspection preparation process becomes more structured and thereby reusable to a larger extent than previously.

2.1. Parameterized Inspection Rules

As mentioned, part of the process is to create parameterized inspection rules for the most commonly used inspection features in practice, i.e. surface point, edge point, circular hole, oval hole, rectangular hole, sphere, and cylinder [2,3]. The parameterization describes the inspection rule in terms of number of points, positions and probe configurations, and the allowed deviation from the ideal/default rule [4]. Today, it is common that the CMM embedded software contains the inspection rules and decides the motion patterns and sequence during feature inspection. However, the proposed approach with parameterized inspection features has four key advantages: (i) it makes the inspection preparation flexible, structured and repeatable, (ii) the same control code can be used with CMMs of different brands with more consistent results, (iii) the inspection sequence inside and between features can be optimized together to minimize cycle time, (iv) if the default inspection rule is not feasible due to collisions then the conflict can automatically be resolved by using the allowed deviation from the default rules. In Figure 4, as an example, the parameterized inspection rule for a circle is defined and illustrated.

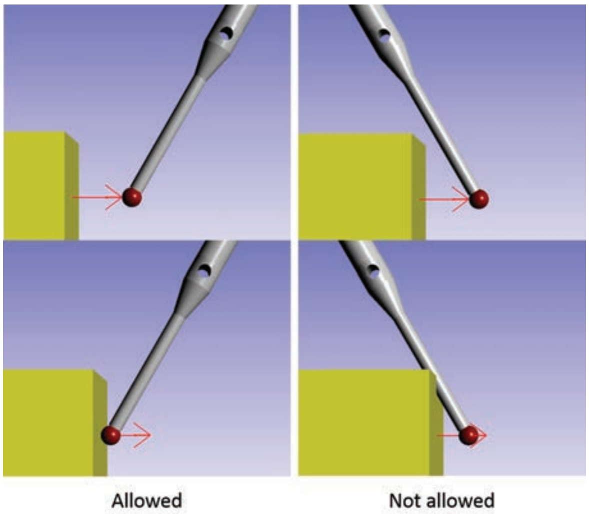

Fig. 3. Approachability illustrated; It should be possible to perform a linear motion along the inspection direction from a specified approach point and that the probe sphere/tip should make contact with the inspection point during that motion without any further collisions. The red arrow represents the normal of the inspection point.

Inspection Feature	Feature Definition	Inspection Rule
Circle	P: Position vector D: Diameter T: Thickness	t1: Measurement depth α1: Start angle (points on) plane α2: Start angle (points in) circle n1: #Points on plane n2: #Points in circle d1: Diameter on circle in plane

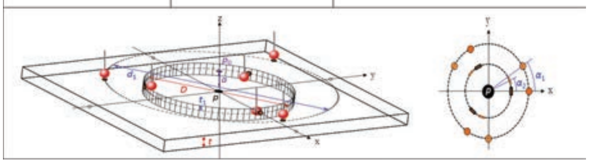

Fig. 4. A parameterized inspection rule of circle feature.

2.2. Automatic Path Planning

The next technology used is path planning where the collision free CMM motions are generated by automatically finding via points and probe reorientations between the inspection features [5,6,15]. Complete path planning algorithms, which always find a solution or determine that none exist, are of little industrial relevance since they are too slow. In fact, the complexity of the problem has proven to be PSPACE-hard for polyhedral object with polyhedral obstacles [7]. Therefore, sampling based techniques trading completeness for speed and simplicity is the choice. Common for these methods are the needs for efficient collision detection, nearest neighbor searching, graph searching and graph representation. The two most popular methods are; Probabilistic Roadmap Methods (PRM) [8] and Rapidly-Exploring Random Trees (RRT) [9]. These methods have been extended and tailored in several ways, for example in [10]. Inspired by these probabilistic methods FCC has developed a deterministic path planner that adaptively adjusts a grid in the configuration space.

2.3. Inspection Sequence Optimization

Data generated by the inspection rule analysis and path planning can then be used to optimize task sequences for robot stations, such as automated welding or measuring. Such optimization can reduce cycle time by as much as 25% and thereby greatly increase efficiency of production [11]. Task sequences can be discretized and modelled as a travelling salesperson problem (TSP) or some variation of it [12]. Introducing increasingly complex attributes to the problem such as different ways to complete each task, precedence constraints and/or several robot arms working on the same object requires the TSP model to be more advanced.

The precedence constraints are introduced by hierarchical relations between features since some features are required to be measured in relation to other features. Typically, to be able to measure some features there is a need for a local alignment. The alignment is a coordinate system calculated from group of measured/actual features. This type of local measurement creates hierarchical relations between features and thus imposes precedence constraints. However, this should not be confused with evaluating features in relation to each other. Sequence constraints are only introduced when features are physically measured in relations to other features. Spitz and Requicha [14] solved a constraint satisfaction problem to handle this. This paper will instead incorporate this directly in the TSP solution.

Therefore, this paper consider the case of optimizing the precedence constrained task sequence of a single arm CMM robot station where each task can be performed in several different ways. Since a CMM has five degrees of freedom, each point can be approached from many different angles and thus be evaluated in a multitude of ways. To model these characteristics, one can discretize a subset of the different ways in which a point can be measured and constrain the order of the points being evaluated. Given such a discretization and set of precedence constraints one can model the problem of minimizing the total cycle time as a precedence constrained generalized travelling salesperson problem (PCGTSP).

Since the PCGTSP is an extension of the GTSP it is also an NP-hard problem [17]. So as with many other NP-hard problems, using exact optimizing algorithms for solving larger problem instances are often impractical and heuristic algorithms are implemented instead [23]. The PCGTSP is similar to two other well-studied variations of the TSP, the sequential ordering problem (SOP) [21–26] and the generalized TSP (GTSP) [16–20], but the PCGTSP has not been extensively studied itself. Therefore, there is a need to develop and evaluate heuristic algorithms for the PCGTSP and their effectiveness on real industrial applications which is what this paper aims to do.

2.4. Results from Volvo Cars

At Volvo Cars a new vehicle program is inspected with typically 700 inspection programs containing up to 25 000 features. By implementing this efficient process for inspection the preparation and programming time have been estimated to be reduced by 75% and the equipment utilization has been improved by 25% more efficient programs. Some examples from the inspection process implementation at Volvo Cars can be seen in Figures 5-7.

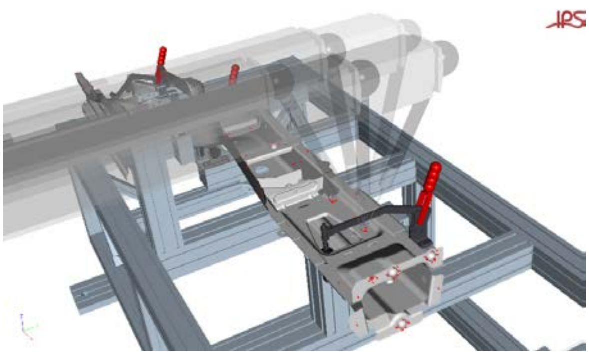

Fig. 5. Feature accessibility analysis resulting in five different collision free probe configuration inspection alternatives (courtesy of Volvo Cars).

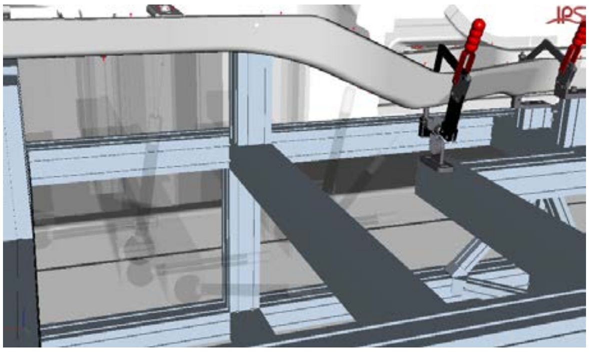

Fig. 6. An automatic generated collision free path between two features containing a non-trivial necessary probe change in the middle. Movement shown by transparent probe states (courtesy of Volvo Cars).

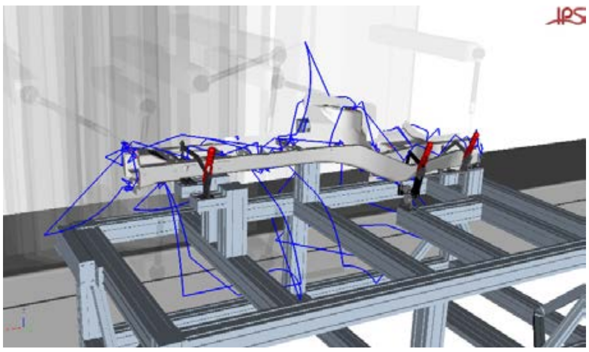

Fig. 7. An optimized collision free inspection sequence (blue trajectory) for 20 features containing 115 points, calculated by the system (courtesy of Volvo Cars).

The rest of the paper will proceed as follows. In Section

3 the PCGTSP is described. Section 4 describes the different solving methods which are evaluated in this paper and Section 5 presents the results when testing these methods on some real industrial cases. Finally, Section 6 presents some final conclusions and suggestions for future research.

3. Problem description

The PCGTSP is a variation of the TSP where the node set is partitioned into groups and then precedence constraints are enforced on a group level, i.e. such that the groups are required to precede each other (but not necessarily directly) in a solution. Because the PCGTSP solution represents a sequence of tasks (modelled as groups) where each task can be performed in different ways (modelled as nodes) it is natural to have the precedence constraints enforced on a group level, since the tasks are required to precede each other.

Let n be the number of nodes in a problem instance and let $V := \{1, \ldots, n\}$ denote the set of all nodes. Let $A := \{(i, j) : i, j \in V, i \neq j\}$ denote the set of all (directed) arcs between all nodes and let c_{ij}, $i, j \in V$, denote the cost associated with the arc from node i to node j. Let $M := \{1, \ldots, m\}$ denote the set of all group indices and let $V_1, \ldots, V_m$ be a partition of V where V_p, $p \in M$, is called a *group*. The partition of V must satisfy $V_p \neq \emptyset$, $V = \cup_{p \in M} V_p$ and $V_p \cap V_q = \emptyset$ when $p \neq q$. Let the precedence constraints be defined by sets which are denoted as $\mathrm{PG}_q := \{p \in M : \textit{group } p \textit{ must precede group } q \textit{ in the tour}\}$, $q \in M$. For these applications a start group, p_{start}, which consists of a single node is specified as the starting position of the robot as well. The PCGTSP is then to find a tour starting from p_{start} such that one node in every group is visited exactly once, the precedence constraints are satisfied and the sum of the cost associated with the traversed arcs is minimized.

When attempting to solve the PCGTSP one can view it as two subproblems: group sequence and node choice, i.e. the order in which the groups are visited and the choice of the node that is to be visited in each group. The group sequence subproblem requires a fixed selection of which node that is to be visited within each group while the node selection subproblem requires a fixed order of the groups to be solved. While there is a clear dependency between these subproblems, heuristic solving algorithms which separate or combine them to different degrees have, however, been shown to be efficient for the GTSP without precedence constraints [19,20].

4. Solution approaches

In this paper two different approaches for solving the PCGTSP are presented. The first approach is a deterministic algorithm which successively expands the set of groups as their precedence constraints are satisfied and uses a high performance heuristic algorithm designed for the GTSP as a lower level solver. The second approach is a stochastic algorithm based on an Ant Colony System (ACS) metaheuristic hybridized with a special purpose local search. This algorithm has been very successful for the SOP [23,24] and was shown to perform quite well for larger problem instances when generalized to the PCGTSP [13]. The generic optimizing software CPLEX is also considered as a solution approach and as a method for obtaining lower bounds.

4.1. CPLEX software

The CPLEX solver uses an advanced but generic method of branch-and-cut to optimize mixed integer linear programming (MILP) formulations of optimization problems. The MILP formulation of the PCGTSP first proposed by Salman in [13] is used to study the effectiveness of such a generic optimizing method in the industrial cases considered in this paper. CPLEX is used for completely solving the PCGTSP to optimality as well as solving the linear programming (LP) relaxed PCGTSP where the integrality constraints are relaxed and a lower bound on the minimal tour length is obtained.

4.2. Sequentially Expanding GTSP (SEG) solver

The general algorithm for the SEG solver is as follows:

Algorithm 1 Sequentially Expanding GTSP

1. Set $k = 1$ and initialize a path $P^1 = \{p_{\text{start}}\}$.
2. Set $U = \{p \in M :$ group p is allowed to be visited given the path $P^k\}$.
3. Let the GTSP solver expand the path $P^k = \{P^k_1, P^k_2, \cdots, P^k_l\}$, $l \leq m$ using the groups in U.
4. If P^k visits all groups in M then add a final arc between P^k_m and P^k_1 to the path P^k, reoptimize the node selection and exit.
5. For each $j = 1, \cdots, l$ check if any groups in M are allowed to be visited given the path $P^{k_j} = \{P^k_1, \cdots, P^k_j\}$. As soon as one or several groups in M are allowed to be visited for some P^{k_j} then set $P^{k+1} = P^{k_j}$, set $k = k + 1$ and go to step 2.

The SEG solver approach handles the precedence constraints implicitly and is also constructive in its nature, meaning that it is deterministic and does not iteratively improve the solution. The benefit of the SEG algorithm is that any GTSP solver can be used in conjunction with this general strategy and one can therefore utilize the many effective solving algorithms which have been developed for the GTSP. A potential drawback is the short-sightedness of the algorithm since it only considers the groups allowed to be visited in the graph given a current path constructed by the GTSP solver.

4.3. Hybridized Ant Colony System (HACS)

The idea for the ACS algorithm is to model a fixed number of ants, N, that iteratively generate feasible solutions to the PCGTSP by traversing arcs, $(i, j) \in A$, in a non-deterministic manner. In each iteration the generation of paths is guided by the depositing of "pheromones", which are denoted $\tau_{ij} \in [0, 1]$, along the arcs that have been traversed by the ant which has produced the shortest tour. The higher the value of τ_{ij}, the higher the probability that arc (i, j) is chosen during the process of generating paths. For each arc $(i, j) \in A$ a fixed parameter $\eta_{ij} \in [0, 1]$ is initialized as $\eta_{ij} = 1/c_{ij}$. This parameter is called the visibility parameter and provides a fixed measurement of how attractive the corresponding arc is for the ants.

However, to avoid getting stuck at locally optimal solutions and to promote diverse solutions the ACS algorithm incorporates a so-called evaporation rate parameter $\rho \in [0, 1]$. Let $T^k = (T^k_1, \ldots, T^k_m)$ be the shortest tour in iteration k. At the end of each iteration k the pheromone levels are updated as $\tau_{ij} = (1 - \rho)\tau_{ij} + \rho/C_{T^k}$ where C_{T^k} is the total cost of tour T^k. Furthermore, during the path generation process, if an ant chooses to traverse an arc (i, j), the pheromone level of that arc is updated as $\tau_{ij} = (1 - \rho)\tau_{ij} + \rho\tau_0$ where τ_0 is the initial pheromone level parameter. The ACS algorithm also introduces a probability $d_0 \in [0, 1]$ that the arc chosen by an ant during the path generation is the arc which is the most attractive. Otherwise, i.e. with probability $(1 - d_0)$, an arc (i, j) is chosen with probability

$$p^a_{i,j} = \begin{cases} \frac{[\tau_{ij}]^\alpha[\eta_{ij}]^\beta}{\sum_{l \in V(T^a)}[\tau_{il}]^\alpha[\eta_{il}]^\beta} & \text{if } j \in V(T^a) \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where α and β be parameters that control the relative importance of the pheromone level and the visibility parameter and $V(T^a)$ is the set of allowed nodes given a tour T^a of ant a.

4.3.1. Local search

After each tour generated by the ACS metaheuristic a local search procedure is executed. First, the node selection of the tour is fully optimized given a fixed order of the groups through a dynamic programming algorithm [13,16,19].

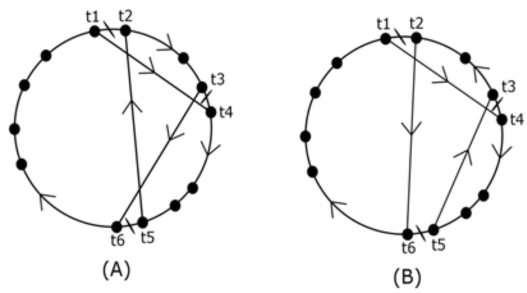

Fig. 8. (A) is an example of a path preserving 3-opt move. (B) is an example of a path inverting 3-opt move.

Then a highly efficient 3-opt local search [23] is performed. The k-opt local search heuristic removes k arcs from an existing tour and adds k arcs such that the tour becomes improved. This 3-opt local search was specifically developed to handle precedence constraints by excluding certain 3-opt moves from the search and was found to perform better than many other k-opt local search heuristics when generalized to the PCGTSP [13]. By excluding so-called path inverting 3-opt moves, i.e. moves that inverts the orientation of one or several segments of the tour (see Figure 8), the algorithm reduces the time spent on verifying that the precedence constraints are satisfied and verifying the improvement condition of a 3-exchange. Furthermore, the 3-opt local search employs a special labelling procedure which makes the verification of the precedence constraints even more efficient.

When a tour which can not be improved further by the 3-opt local search is found, the node selection is fully optimized again.

5. Computational experiments and results

Five problem instances derived from CMM inspection cases of various sizes are studied. Each problem instance is evaluated using the three solution approaches described in Section 4.

The CPLEX software was run with a 24 hour time limit and was run for the LP relaxed problem as well as the original MILP problem for each instance.

The HACS algorithm was run 10 times with 10 ants and 100 iterations per run. The parameters were set to $\rho = 0.1$, $\alpha = 1$, $\beta = 2$, $d_0 = 0.9$ and $\tau_0 = 1/(mC_u)$ where C_u is an upper bound on the minimal tour length. Also, the local search is only run for a generated tour if the cost is within 20% of the best one found so far. This heuristic rule as been found to be benificial in [24].

Let z be the sum of costs c_{ij} for the arcs (i, j) traversed in a solution. An optimal solution is then the shortest possible tour given the graph of a problem instance.

Table 1. Results from CPLEX. z^*_{LP} is the minimal solution for the LP relaxed problem and z^*_{MILP} is the minimal tour length for the original MILP problem. T_{LP} is the time for the LP relaxed problem and T_{MILP} is the time for the original MILP problem.

Instance	m	n	z^*_{LP}	T_{LP} (s)	z^*_{MILP}	T_{MILP} (s)
cmm001	13	15	48.85	0.03	49.12	0.03
cmm002	16	25	7.60	0.03	20.26	1.86
cmm003	18	36	11.43	0.19	20.04	0.41
cmm004	91	216	23.00	3936.43	-	>86400
cmm005	174	405	-	>86400	-	>86400

Table 1 shows the tour lengths when running the problem instances in the generic optimizing software CPLEX. For the two larger problems, cmm004 and cmm005, CPLEX was not able to find an optimal solution within the time limit of 24 hours and for cmm005 CPLEX was not able to solve the LP relaxation to optimality within the time limit either.

Table 2. Tour lengths and average running times for the heuristic algorithms. $z^{\mathrm{best}}_{\mathrm{HACS}}$ is the best (shortest) tour length out of 10 runs. T_{HACS} and T_{SEG} is the average time for completing a run.

Instance	m	n	z_{SEG}	T_{SEG} (s)	$z^{\mathrm{best}}_{\mathrm{HACS}}$	T_{HACS} (s)
cmm001	13	15	49.12	0.01	49.12	6.93
cmm002	16	25	20.48	0.02	20.73	9.56
cmm003	18	36	20.46	0.02	20.04	6.69
cmm004	91	216	48.31	2.80	46.07	286.91
cmm005	174	405	212.23	22.03	185.83	698.52

Table 2 shows the results from the heuristic algorithms. For the smaller instances, cmm001-003, the difference in solution quality is marginal. For cmm004 the HACS algorithm performs a bit better than the SEG solver and for cmm005 the solution produced by the HACS algorithm is significantly better. While the HACS algorithm is much slower than the SEG solver, Table 3 suggests that the number of iterations can probably be

Table 3. More detailed results for the HACS algorithm. $z^{\text{avg}}_{\text{HACS}}$ is the average solution found for an instance over 10 runs. T_{bavg} is the average running time and I_{bavg} is the average number of iterations elapsed before the best solution is found by the HACS algorithm in each run.

Instance	m	n	$z^{\text{best}}_{\text{HACS}}$	$z^{\text{avg}}_{\text{HACS}}$	T_{bavg} (s)	I_{bavg}
cmm001	13	15	49.12	49.12	0.02	1
cmm002	16	25	20.73	20.73	0.20	2
cmm003	18	36	20.04	20.11	1.36	13
cmm004	91	216	46.07	46.98	176.56	54
cmm005	174	405	185.83	187.61	237.63	33

lowered by almost 50% without any significant loss of solution quality for these problem instances.

6. Conclusions and future research

The productivity of the CMM inspection process and equipment is significantly improved by a structured inspection preparation process combined with automatic path planning. Inspection sequence optimization is an important part of the improvement. In this paper, the optimization part related to inspection sequence precedence constraints is further improved.

The presented HACS algorithm is able to reduce cycle time of the largest case by more than 10% on average in comparison to the now used SEG solver and while it is much slower, the number of iterations can probably be significantly tightened for the studied cases without losing much in terms of solution quality. The results from the CPLEX software shows the need for developing heuristic algorithms and special purpose optimizing algorithms for the PCGTSP.

Further development of the MILP model in conjunction with the optimizing algorithms might enable optimization of small to medium sized problem instances within reasonable computation times. For some industrial cases there arises a need for multiple CMMs evaluating features on the same object which corresponds to expanding the PCGTSP to a precedence constrained generalized multiple travelling salesperson problem (PCGmTSP).

Acknowledgements

This work was carried out within the Wingquist Laboratory VINN Excellence Centre, supported by the Swedish Governmental Agency for Innovation Systems (VINNOVA). It is also part of the Sustainable Production Initiative and the Production Area of Advance at Chalmers University of Technology.

References

[1] R. Söderberg, L. Lindkvist, J.S. Carlson, Virtual Geometry Assurance for Effective Product Realization, 1st Nordic Conference on Product Lifecycle Man- agement – NordPLM'06; 2006, Göteborg, Sweden.
[2] M.R. Henderson, D.C. Anderson, Computer Recognition and Extraction of Form Features: A CAD/CAM Link, Computers in Industry; 1984, vol. 5, pp. 329-339.
[3] H.C. Lee, W.C. Jhee, H. S. Park, Generative CAPP Through Projective Feature Recognition, Computers and Industerial Engineering; 2007, vol. 53, pp. 241-246.
[4] F. Zhao, X. Xu, S.Q. Xie, Computer Aided Inspection Planning: The state of the art, Computer Aided design; 2009, vol. 60(2), pp. 453-466.
[5] C.W. Ziemian, D.J. Medeiros, Automating probe selection and part setup planning for inspection on a coordinate measuring machine, International Journal of Computer Integrated Manufacturing; 1998, vol. 11(5), pp. 448-460.
[6] C.W. Ziemian, D.J. Medeiros, Automated feature accessibility algorithm for inspection on a coordinate measuring machine, International Journal of Production Research; 1997, vol. (35)10, pp. 2839-2856.
[7] J.F. Canny, The Complexity of Robot Motion Planning, MIT Press; 1988.
[8] R. Bohlin, L.E. Kavraki, Path Planning Using Lazy PRM, In: Proc. IEEE Int. Conf. On Robotics and Automation; 2000.
[9] S.M. LaValle, J.J. Kuffner, Randomized Kinodynamic Planning, In: Proc. IEEE Int. Conf. on Robotics and Automation; 1999.
[10] S. Karaman, E. Frazzoli, Sampling-based Algorithms for Optimal Motion Planning, International Journal of Robotics Research; 2011, vol. 30, iss. 7, pp. 846-894.
[11] J. Segeborn, D. Segerdahl, F. Ekstedt, J.S. Carlson, M. Andersson, R. Söderberg, "An Industrially Validated Method for Weld Load Balancing in Multi Station Sheet Metal Assembly Lines", ASME Journal of Manufacturing Science and Engineering; 2013.
[12] S. Björkenstam, D. Spensieri, J.S. Carlson, R. Bohlin, D. Gleeson, Efficient Sequencing of Industrial Robots through Optimal Control, Procedia CIRP 2014, vol 23, pp. 194-199.
[13] R. Salman (2015), Algorithms for the precedence constrained generalized travelling salesperson problem (Master's thesis), Retrieved from http://www.chalmers.se/en/departments/math/research/research-groups/optimization/Pages/master-thesis-projects.aspx.
[14] S.N. Spitz, A.A.G. Requicha, Hierarchical Constraint Satisfaction for Dimensional Inspection Planning, Proceedings of the 1999 IEEE International Symposium on Assembly and Task Planning, Porto, Portugal, July 1999.
[15] A.J. Spyridi, A.G. Requicha, Accessibility analysis for the automatic inspection of mechanical parts using coordinate measuring machines, International conference on Robotics and Automation; 1990, pp.1284-1289.
[16] M. Fischetti, J.J. Salazar Gonsalez, P. Toth, A Branch-And-Cut Algorithm for the Symmetric Generalized Traveling Salesman Problem, Operations Research; 1997, vol 45, pp. 378–394.
[17] I. Kara, H. Guden, O.N. Koc, New Formulations for the Generalized Traveling Salesman Problem, In Proceedings of the 6th International Conference on Applied Mathematics, Simulation, Modelling, ASM'12, pp. 60–65, Stevens Point, Wisconsin, USA, 2012. World Scientific and Engineering Academy and Society (WSEAS).
[18] D. Karapetyan, G. Gutin, Lin-Kernighan Heuristic Adaptation for the Generalized Trav- eling Salesman Problem, European Journal of Operational Research; 2011, vol 208, pp. 221–232.
[19] D. Karapetyan, G. Gutin, Local Search Algorithms for the Generalized Traveling Salesman Problem. European Journal of Operational Research; 2012, vol 219, pp. 234–251.
[20] G. Gutin, D. Karapetyan, N. Krasnogor, Memetic Algorithm for the Generalized Asymmetric Traveling Salesman Problem, In Nature Inspired Cooperative Strategies for Optimization (NICSO 2007), vol 129 of Studies in Computational Intelligence, pp. 199– 210, Springer Berlin Heidelberg; 2008.
[21] S.C. Sarin, H.D. Sherali, A. Bhootra, New tighter polynomial length formulations for the asymmetric traveling salesman problem with and without precedence constraints, Operations Research Letters; 2005, vol 33, pp. 62–70.
[22] N. Ascheuer, M. Jünger, G. Reinelt, A Branch & Cut Algorithm for the Asymmetric Traveling Salesman Problem with Precedence Constraints, Computational Optimization and Applications; 2000, vol 17, pp. 61–84.
[23] L.M. Gambardella, M. Dorigo, An Ant Colony System Hybridized with a New Local Search for the Sequential Ordering Problem, INFORMS Journal on Computing; 2000, vol 12, pp. 237–255.
[24] L.M. Gambardella, R. Montemanni, D. Weyland, An Enhanced Ant Colony System for the Sequential Ordering Problem, In Operations Research Proceedings 2011, Operations Research Proceedings, pp. 355–360, Springer Berlin Heidelberg; 2012.
[25] D. Anghinolfi, R. Montemanni, M. Paolucci, L.M. Gambardella, A hybrid particle swarm optimization approach for the sequential ordering problem, Computers & Operations Research; 2011, vol 38, pp. 1076–1085.
[26] J. Sung, B. Jeong. An Adaptive Evolutionary Algorithm for Traveling Salesman Problem with Precedence Constraints. The Scientific World Journal, 2014, 2014.