Proceedings Paper:

Duro, J.A. orcid.org/0000-0002-7684-4707, Ward, R., Rooke, J. orcid.org/0000-0003-0720-7891 et al. (5 more authors) (2024) Digital framework for metallic subtractive process planning: Liger optimisation case study. In: Putnik, Goran, (ed.) Procedia CIRP. 57th CIRP Conference on Manufacturing Systems 2024 (CMS 2024), 29-31 May 2024, Póvoa de Varzim, Portugal. Elsevier BV , pp. 1358-1363.

https://doi.org/10.1016/j.procir.2024.10.252

Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 130 (2024) 1358–1363

www.elsevier.com/locate/procedia

57th CIRP Conference on Manufacturing Systems 2024 (CMS 2024)

Digital framework for metallic subtractive process planning: Liger optimisation case study

João A. Duro[a], Rob Ward[a,b,*], Jack Rooke[b], Emily Pickford[b], Rich Bonnell[b], Adam Brown[b], Robin C. Purshouse[a], David Curtis[b]

[a]Department of Automatic Control and Systems Engineering, University of Sheffield, Sheffield, S1 3JD, UK
[b]University of Sheffield Advanced Manufacturing Research Centre, University of Sheffield, Rotherham, S60 5TZ, UK

* Corresponding author. *E-mail address:* r.ward@amrc.co.uk

Abstract

Metallic subtractive process planning is challenging. It requires a blend of expert knowledge, experience, and established manufacturing methods. Decision-making in this domain is multifaceted and complex, and this includes the need to satisfy multiple conflicting demands faced by the manufacturing sector today. However, with the evolving landscape of manufacturing, combined with increased adoptions of digital technologies, the traditional rules of thumb are becoming obsolete. In response, this research investigated these decision-making processes within a subset of the digital framework for subtractive process planning. In particular, it has demonstrated that an advanced open-source optimisation tool (Liger) could be integrated with commercial industrial software and applied to a machining case study. The integrated software intelligently executed and automated simulations using advanced optimisation techniques. The outputs are a range of optimised solutions that support manufacturing engineers' decision-making. One of the major benefits of the open-source software is that it provides an intuitive to use interface suitable for the non-expert in optimisation, offers a varied range of state-of-the-art multi-objective optimisation algorithms, and is capable of incorporating many different third-party types of software, models and simulations. In this paper, the Liger software has been applied to a simple case study to find the best feedrate and depths of cut that simultaneously minimise the cutting force and the time it takes to complete the machining process. The case study demonstrates proof-of-principle results that the optimisation software can be implemented in a wider process planning context with additional digital manufacturing simulation models.

Keywords: Machining; Process Planning; Multi-Objective Optimisation

1. Introduction

Process planning in metallic subtractive manufacturing is complex. The processes rely on expert knowledge and experience of engineers backed up with many tried and tested methods of manufacture. The number of decisions during process planning is vast, some are based on experience, some on computational optimisation and some on general "rules of thumb". Recently, these rules have come under scrutiny due to the increased challenges balancing productivity, cost, quality and sustainability. Traditional rules no longer apply. Therefore, to address this, the authors undertook research to capture these decision points and propose a digital framework to support a robust and adaptable future process planning system for subtractive processes.

As a subset problem of the wider digital framework, a research stream focused on incorporating advanced optimisation techniques from academia with commercial industrial software, the results are presented herewith. This paper is written as follows, first, optimisation methods and the open-source optimisation software [1] are introduced, then integration and application of the Liger optimisation is presented, followed by the results and concluding remarks.

2. Optimisation and Decision-Making

The study and research of optimisation and decision-making is rooted in mathematical theories and computational techniques. To give a general appreciation of the field, the main topics are briefly described here.

2212-8271
10.1016/j.procir.2024.10.252

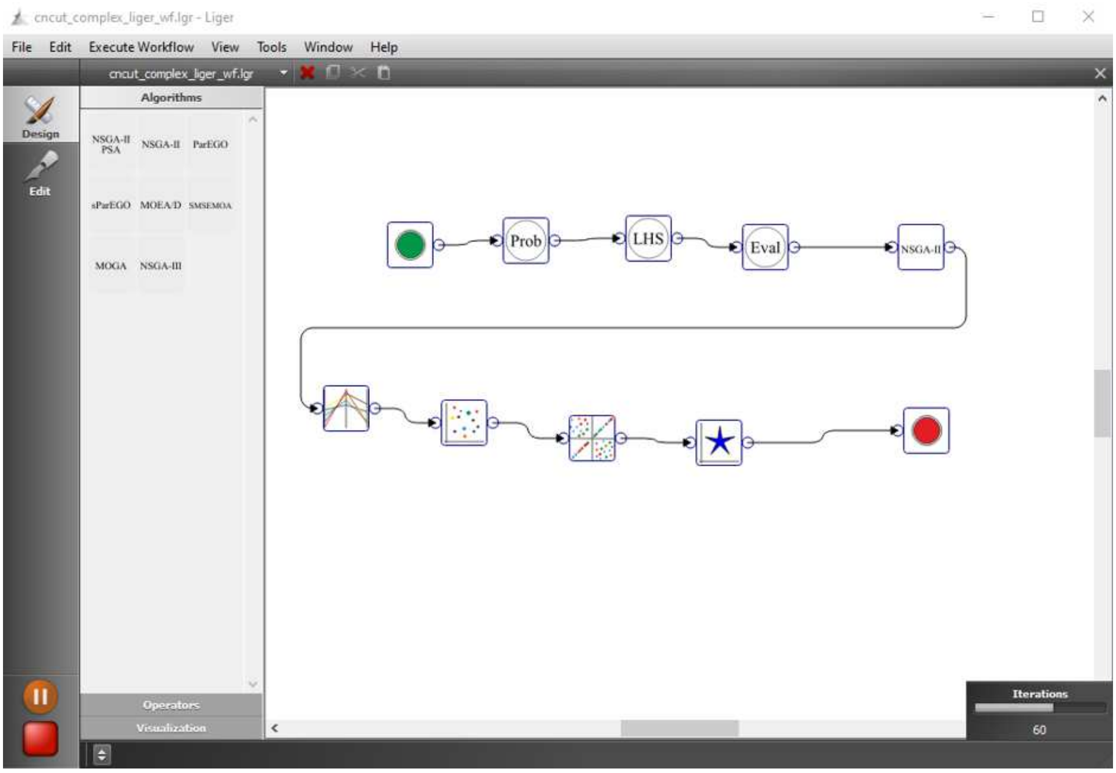


Fig. 1: Liger graphical user interface running an optimization workflow with the subtractive process planning problem.

1. **Integer and Mixed-Integer Programming**. These techniques are used when decision variables are required to be integers, which is common in problems like scheduling and allocation where fractional results don't make practical sense. In a process planning context, this is the method used in sequencing.
2. **Linear and Non-linear Optimisation**. Linear optimisation deals with problems where the objective function and constraints are linear, meaning they can be expressed as simple inequalities, e.g., in machining, feedrate should never exceed a given threshold in a part program. Non-linear optimisation, on the other hand, handles problems where at least one constraint or the objective function is non-linear—this could relate to expressions for toolwear for example.
3. **Multi-objective Optimisation**. In problems with a single objective there is only one optimal solution, corresponding to the minimum of the objective function to be optimised (assuming minimisation). When there are multiple conflicting objectives, there is an optimal solution per objective, and there is also a set of optimal trade-off solutions where a gain in one objective is obtained at the expense of another objective. In many engineering contexts, there's a need to optimise multiple conflicting objectives simultaneously, like maximising performance while minimising cost. This generally is what an experienced engineer will do without realising, i.e., they will trade-off multiple objectives such as cost, cycle time and part quality. Pareto optimality is a commonly used concept in this area to identify the best trade-offs. Consider two solutions u, v of a given multi-objective optimisation problem, then u is said to dominate v if and only if: (i) u is not worse than v in all objectives, and (ii) u is strictly better than v in at least one objective. In case neither solution dominates the other then u and v are said to be non-dominated. A solution is said to be Pareto optimal if it is not possible to find another solution in the entire decision space that dominates it. The set of all Pareto optimal solutions is referred to as the Pareto set and the image of this set in the objective space is the Pareto front.
4. **Stochastic Optimisation**. For problems with uncertainty in input data or inherent randomness, stochastic optimisation is employed. It considers various possible scenarios, often integrating techniques from probability and statistics. The aim is to find solutions that provide robust performance under the uncertainties.
5. **Evolutionary and Genetic Algorithms**. Inspired by biological evolution concepts like "survival of the fittest", these algorithms use mechanisms like mutation, crossover, and selection to drive the search towards the optimal solution(s).
6. **Simulation-based Optimisation**. Often, direct mathematical modelling is challenging. In such cases, optimisation is achieved by combining simulation models with optimisation techniques, commonly seen in complex systems or processes. For example, combining virtual machining models or CAM software such as CATIA V5.

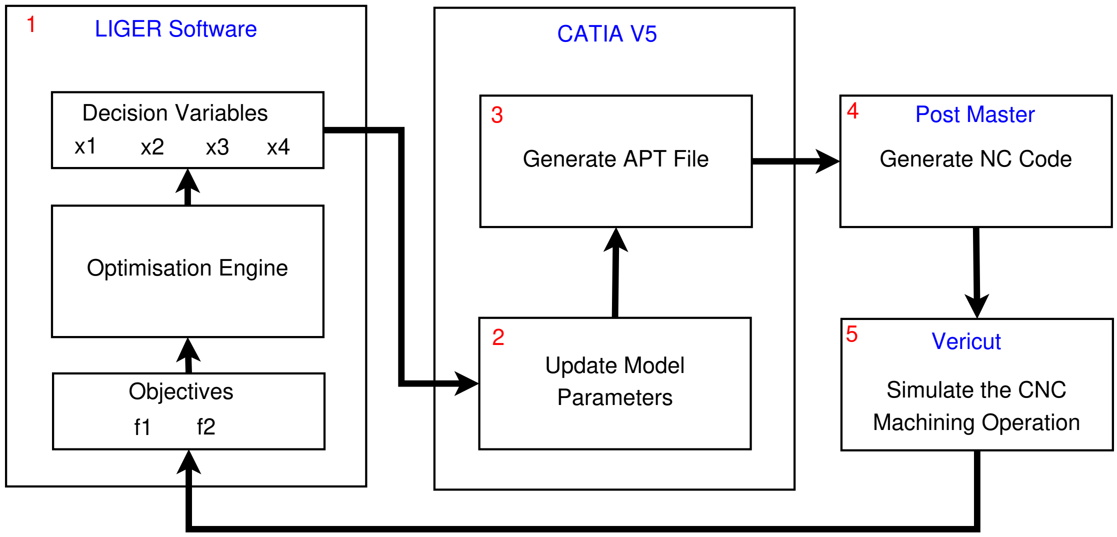


Fig. 2: Process flowchart for machining case study.

Optimisation in real-world settings is complex, often involves multiple conflicting objectives, and it is typically best addressed by using multi-objective optimisation methods. This approach offers a range of balanced solutions for decision-makers. However, the complex nature of multi-objective optimisation algorithms can hinder the optimisation process. Determining the best algorithm for a particular problem, or selecting the right set of parameters, typically demands a good level of expertise. The absence of accessible, user-friendly, and transparent software can result in prolonged time it takes to complete the design of the product and incur in higher costs. To address these challenges, one major focus of the research in [2, 3, 4] within the University of Sheffield's department of Automatic Control and Systems Engineering has been the development of a software tool called Liger. Liger is a cross-platform open-source integrated optimisation and decision-making environment. The next section will describe the software in more detail.

2.1. Liger Software

Liger is an open-source optimisation platform developed with both scalability and user accessibility in mind. This makes it ideal for industrial applications, even for those professionals not deeply versed in optimisation. Central to Liger is a visual programming interface, streamlining the process of setting up optimisation workflows (see Figure 1). While the market has various optimisation solutions, the majority addressing genuine industrial challenges are often proprietary and not open-source. These solutions tend to be intricate libraries, necessitating significant time investment for effective deployment. In contrast, Liger aims to reduce this time overhead, offering engineers a straightforward and efficient optimisation tool.

To demonstrate the feasibility of using the Liger software in process planning a simple framework was established. This framework enables the automation of Computer Aided Manufacture (CAM) parameter selection with subsequent virtual machining processes. In this initial framework, the Liger software connected the commercial CAM package CATIA V5 with VERICUT via a set of macros. The system was able to automate changing machining parameters such as axial and radial depth of cut, feedrate and spindle speed. The updated part program was then automatically sent to the VERICUT software to utilise the VERICUT Force plugin for the cutting force prediction. The output of the VERICUT Force simulations was sent back to the Liger software.

Figure 1 shows the Liger interface and the setup of the Liger workflow. The workflow is formed by different components as shown in the figure, described as follows:

1. The optimisation problem is defined inside the Prob component. This is where the decision variables, objectives or even constraints (if any) are defined. This is also where the integration with CATIA and VERICUT happens.
2. LHS stands for Latin Hypercube Sampling, a popular space filling approach. This component is responsible for creating the initial population (a set of candidate solutions), which forms the starting point for the optimisation algorithm.
3. The Eval component is responsible for evaluating the solutions, and in this case the solutions are evaluated one after the other in a sequence. There are other more advanced components in Liger that can evaluate the solutions in parallel, thus saving time.
4. NSGA-II [5] is the multi-objective evolutionary optimisation algorithm (MOEA) used for this case study. This is a widely used MOEA for dealing with real-world problems, including for process planning optimisation.
5. The components in the second row of the workflow are state-of-the-art visualisation plots, suitable for visualising the solutions of problems containing many variables and objectives. Examples of such visualisation plots are scatter plot, parallel coordinate plots and glyphs plots.

3. Method

3.1. Machining Case Study

The case study was a closed pocket milling operation of a 7075-T6 aluminium aerostructure part using a 3 fluted 16 mm solid carbide end mill (SGS S-Carb APR tool EDP No. 44688), as shown in Figure 4.

The first task in the optimisation process is to determine the objectives. In this multi-objective case study, the objectives were to minimise the cycle time (f_1) and minimise the maximum cutting force (f_2). The decision variables were chosen as axial depth of cut (x_1), radial depth of cut (x_2), feedrate (x_3) and spindle speed (x_4). The ranges are given below:
Discrete decision variables:

- x_1: Axial depth of cut: 1,2,...,15 (1 mm increments)
- x_2: Radial depth of cut: 5,10,...,65 (% of cutter diameter)

Continuous decision variables:

- x_3: Feed rate: [100, 12000] (mm/min)
- x_4: Spindle speed: [800, 20000] (RPM)

The first two decision variables are discrete, implying that their allowed values are restricted to a set of integer values as shown. The last two decision variables are continuous, meaning that they can take any values from within the interval shown.

Figure 2 shows the evaluation of a solution by Liger, which requires calling each software in the given sequence:

1. Initially, Liger decides the decision variable values for the candidate solution that needs to be evaluated.
2. The new decision variable values are sent to CATIA V5 software, where the model parameters are updated, and new machining paths are generated accordingly.
3. These new machining paths are translated into APT source format by using CATIA V5. This APT file contains a set of instructions used by the machine to machine the part.
4. The APT file needs to be converted to a controller specific format that is compatible with the CNC machine. In this case the code is converted to G-code, which is stored in an NC (numerical control) file with extension MPF. The conversion process is conducted by the in-house post-processing software Post Master.
5. The MPF file created by the Post Master is then sent to the VERICUT software (screenshot shown in Figure 3), which simulates the machining operation (cutting forces), and as an output calculates the cycle time (f_1) and maximum forces (f_2) involved in the given operation.
6. These two outputs are sent to Liger and this completes the solution evaluation process. This sequence of events is repeated for each solution that needs to be evaluated.

The CAD model of the workpiece and the machined part from this study is shown in Figure 4. The optimization conducted in this study focused on only one part program, namely the closed pocket F3 in Figure 4 and the parameters of the remaining programs were kept constant.

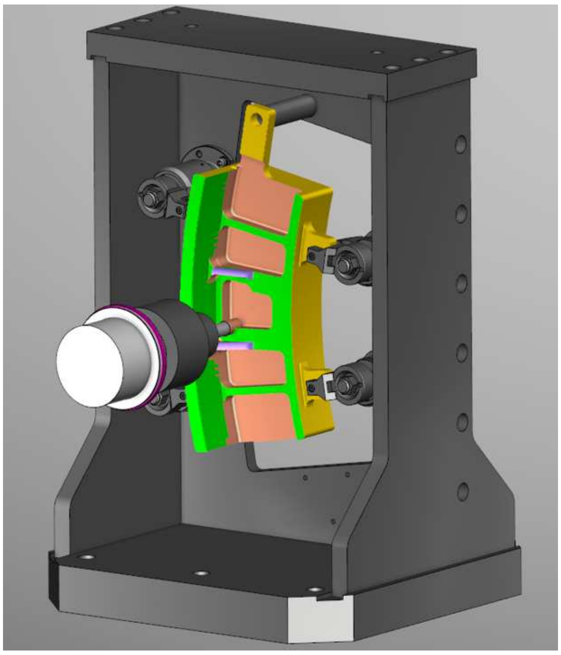


Fig. 3: VERICUT screenshot showing the case study machining operation.

3.2. Experimental Settings

In this study a comparison is conducted between an optimisation algorithm and a Design of Experiments (DoE) technique known as Latin Hypercube Sampling (LHS). It is common to use a DoE to study the behaviour of a computer simulation model or metamodel, allowing us to see the effect of changing the parameters on the model performance. There are different DoE techniques in the literature and some examples are: factorial experiments, D-optimal or I-optimal designs, and LHS. The latter is suitable for computer experiments when no prior knowledge exists about the form of the model, and that interesting phenomena are likely to be found in different regions of the experimental space. This is a type of space-filling design that does not contain any replicated runs and it is therefore suitable for a deterministic computer model. The optimisation algorithm used is known as NSGA-II. This is by far the most well-known multi-objective optimisation algorithm and has been applied to a large range of manufacturing problems in the literature. For the algorithm to handle this specific case study, the authors have enhanced NSGA-II with mixed-integer capabilities. This implies that depending on the type of decision variable (either continuous or discrete), different types of genetic operators (in this case variation operators) are used. For all cases the same probability of crossover and mutation was used which is set to 90% and 10%, respectively. Specifically for each case:

- **Continuous variables**: the crossover and mutation operators are simulated binary crossover (SBX) [6] and poly-

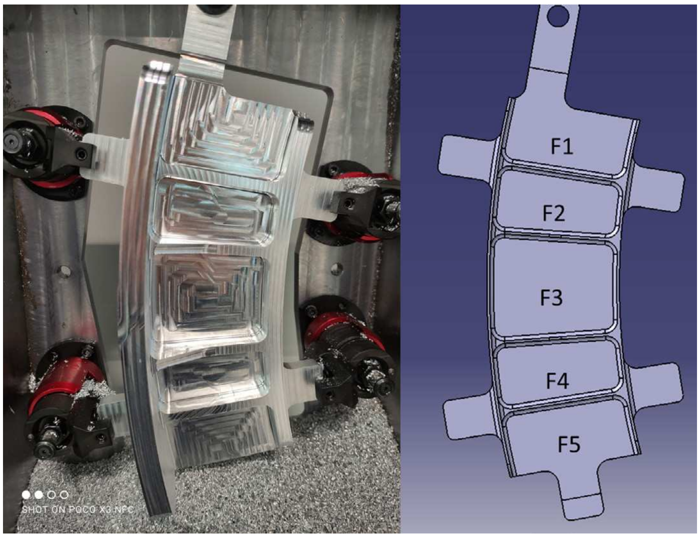


Fig. 4: Machined workpiece (left) and CATIA model (right). For scale the part is approx 400x150 mm.

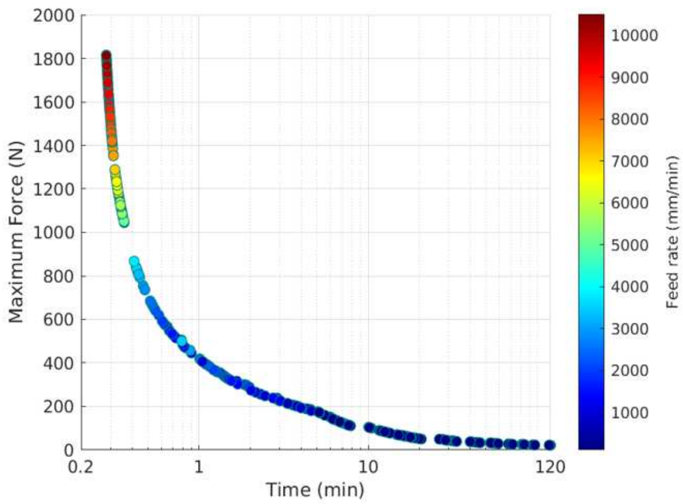


Fig. 5: Non-dominated solutions obtained by NSGA-II that show the effect of feed rate on the objectives.

nomial mutation [7]. The distribution index for SBX and polynomial mutation is set to 15 and 20, respectively.

- **Discrete variables**: The crossover operator creates new offsprings by swapping the values of the parents. The integer mutation operator is taken from [8], where the step size parameter is set to 10% of the length defined by the upper and lower bound of each decision variable.

For a fair comparison, both approaches have been given the same computational budget, that is, $N = 2000$ function evaluations. NSGA-II initially uses a DoE technique for creating the initial population. This population has been initialised by using the LHS technique with a total of $Ni = 100$ solutions. The size of the parent and offspring populations in NSGA-II have been set to $Np = Ni/2 = 50$ each, and this implies that NSGA-II conducts a total of $(N - Ni)/Np = 38$ generations (or iterations) in order to use the entire computational budget.

Initially 10 evaluations were conducted with different candidate solutions and on average it took 3 minutes and 33 seconds for each evaluation to complete. This implies that it is estimated for the optimization algorithm to take approximately 22 hours and 10 minutes to complete an optimization run.

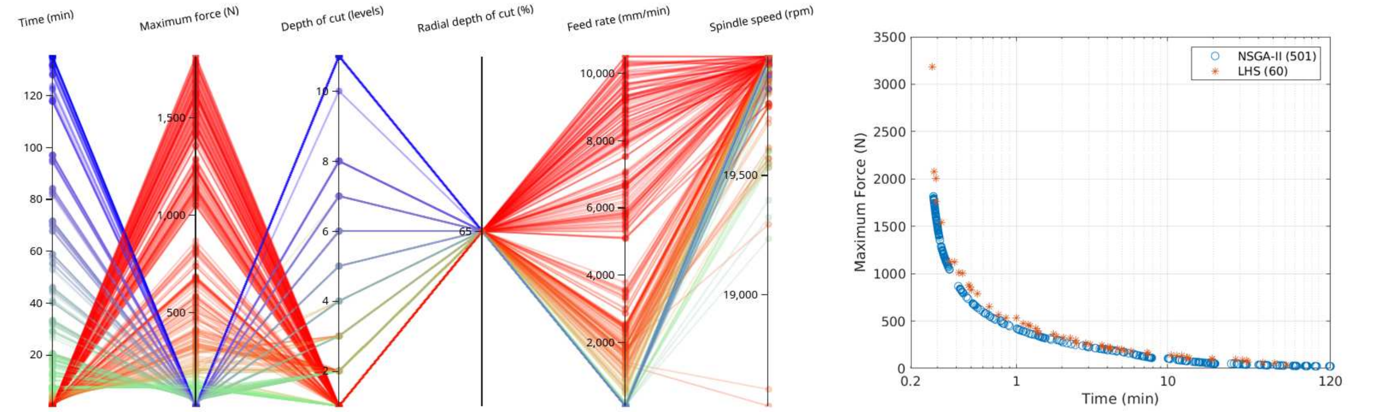
4. Experimental Results

First, the results obtained by NSGA-II are discussed, and these are shown by the Liger visualisation tools in Figures 5 and 6. Figure 5 shows the non-dominated solutions in a scatter plot obtained at the end of the optimization run forming the Pareto front. Figure 6a shows the same non-dominated solutions on a parallel coordinates plot. This non-dominated set is comprised of 501 solutions taken from the total of 2000 function evaluations obtained during this run. In this type of plots each solution is represented by a line formed by connecting the points across all the dimensions. The slides on each vertical bar can set the parameter range and highlight the solutions inside that range. The solutions have been colour coded with respect to the cycle time, where red and blue colour correspond to low and high cycle time across all dimensions, respectively. In this problem the two objectives are in conflict as shown in Figure 5, since an improvement to one objective leads to a loss of performance in the other objective. This conflict between the objectives is also captured by the parallel coordinates plot, which is shown by the crossing lines between cycle time and maximum force in Figure 6a. The selection of the most appropriate solution from amongst the available options across the Pareto front will be determined by the manufacturing priority. The results shown in Figure 6a indicate that for Pareto optimality the radial depth of cut should be kept constant at 65% while the axial depth of cut can vary between 1 and 10 levels.

It is also possible to visualise all solutions that have been evaluated, and not just those that are non-dominated in case this is desirable, this is a useful feature for defining a region of interest, both in objective and decision space. This is likely to aid the decision-maker in selecting the most desirable solution, for example, by restricting the results to a particular radial depth of cut or feedrate.

Another useful feature in Liger is the ability to use the scatter plot to visualise multiple dimensions at once as shown in Figure 5. In this case the third dimension is represented by the colour of the markers. In Figure 5 the two objectives are shown alongside the feedrate, visually it shows that decreasing the feed rate implies a reduction in the maximum force whilst the cycle time increases.

In Figure 6b, comparison is shown between the non-dominated solutions obtained by NSGA-II and the non-dominated solutions obtained by LHS. The set obtained by NSGA-II offers better convergence across most of the Pareto-front. There is only one solution found by the LHS (top solution in Figure 10), which is non-dominated with respect to the solutions obtained by NSGA-II. A quarter of solutions (501) found



(a) Parallel coordinates plot showing the NSGA-II non-dominated solutions colour coded with respect to cycle time

(b) Comparison between the non-dominated solutions obtained by NSGA-II and the Latin Hypercube Sampling (LHS) technique

Fig. 6: Non-dominated solutions obtained by the Liger software.

by NSGA-II of the available are non-dominated, and from those found by LHS only 60 are non-dominated. This is expected given that NSGA-II relies on the dominance relation to decide which solutions are worth evaluating next, while LHS only relies on a space filling metric.

5. Conclusions

The main benefit of the Liger software is the ability to generate solutions for the manufacturing engineer to use as part of their toolbox alongside their experience to support machining strategy and parameter selection. The optimisation engine in Liger identifies a richer set of trade-off options to choose from (over 8 times as many as the DoE) and also offers performance improvements within those trade-offs. For example, for a cycle time of less than 1 minute, the maximum force for Liger's solution is 50% less than for the DoE. The Liger software is essentially an adaptive design of experiments software such that it chooses the next set of parameters to sample (and run simulations) based on the previous set of results. This optimisation of the sample space reduces the required computations exponentially saving significant time compared to standard brute force methods.

The project successfully demonstrated that the open-source Liger optimisation software can be applied to subtractive process problems. It was demonstrated that it can form a subset of a larger digital framework for subtractive processes. It is envisaged that the software and future research will play a key part in the strategy to form the wider digital framework for autonomous subtractive manufacturing.

Acknowledgements

The authors acknowledge the collaboration funding provided by the department of Automatic Control and Systems Engineering and the follow-on funding provided by the High Value Manufacturing Catapult.

References

[1] João A. Duro, Yiming Yan, Ioannis Giagkiozis, Stefanos Giagkiozis, Shaul Salomon, Daniel C. Oara, Ambuj K. Sriwastava, Jacqui Morison, Claire M. Freeman, Robert J. Lygoe, Robin C. Purshouse, and Peter J. Fleming. Liger: A cross-platform open-source integrated optimization and decision-making environment. *Applied Soft Computing*, 98, 2021.

[2] João A. Duro, Yiming Yan, Robin C. Purshouse, and Peter J. Fleming. Collaborative multi-objective optimization for distributed design of complex products. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '18, pages 625—632, New York, NY, USA, 2018. ACM.

[3] Victoria Johnson, João A. Duro, Visakan Kadirkamanathan, and Robin C. Purshouse. A scalable test suite for bi-objective multidisciplinary optimization. In *Evolutionary Multi-Criterion Optimization*, pages 319–332. Springer Nature Switzerland, 2023.

[4] João A. Duro, Umud Esat Ozturk, Daniel C. Oara, Shaul Salomon, Robert J. Lygoe, Richard Burke, and Robin C. Purshouse. Methods for constrained optimization of expensive mixed-integer multi-objective problems, with application to an internal combustion engine design problem. *European Journal of Operational Research*, 307(1):421–446, 2023.

[5] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.

[6] Kalyanmoy Deb and Amarendea Kumar. Real-coded genetic algorithms with simulated binary crossover : Studies on multimodal and multiobjective problems. *Complex Systems*, 9, 1995.

[7] Kalyanmoy Deb and Mayank Goyal. A combined genetic adaptive search (GeneAS) for engineering design. *Computer Science and Informatics*, 26:30–45, 1996.

[8] Günter Rudolph. An evolutionary algorithm for integer programming. In Yuval Davidor, Hans-Paul Schwefel, and Reinhard Männer, editors, *Parallel Problem Solving from Nature — PPSN III*, pages 139–148, Berlin, Heidelberg, 1994. Springer Berlin Heidelberg.