Evolutionary and Deterministic Methods for Design,
Optimization and Control with Applications to Industrial and Societal Problems
EUROGEN 2005
R. Schilling, W.Haase, J. Periaux, H. Baier, G. Bugeda (Eds)
© FLM, Munich, 2005

MULTI-OBJECTIVE DESIGN OF OPTIMAL SCULPTURED SURFACE ROUGH MACHINING THROUGH PARETO AND NASH TECHNIQUES

Agathocles A. Krimpenis, Panagiotis I. K. Liakopoulos, Kyriakos C. Giannakoglou and George-C. Vosniakos

School of Mechanical Engineering
National Technical University of Athens, Greece
E-mail: agathocles@central.ntua.gr

Key words: Evolutionary Algorithms, Pareto front, Nash game, CAM, CNC rough machining, sculptured surface.

Abstract. *Sculptured surface parts machining is a demanding task even though commercially available CAM software offers a variety of possible strategies and a number of machining planning functions. However, process optimization routines are seldom embedded in CAM software. The present work aims at proposing an optimization method that computes optimal process parameter values for sculptured surface rough machining. The objectives are minimum machining time and maximum material removal, subject to technological constraints imposed by the rough machining process. Optimization is based on multi-objective evolutionary algorithms (EAs), assisted by surrogate evaluation models. In addition, a two-player Nash game optimization was programmed and tested. Software was developed for the interaction between the Application Program Interface (API) of a given CAM system and the optimization tool as well as for the calculation of objective function values. The proposed optimization process was applied to the machining of three parts consisting almost entirely of sculptured surfaces; a part in the form of a hip prosthesis and two turbomachinery blades. Emphasis is put on the quality of the optimal solution as well as its contribution to decision making.*

1 INTRODUCTION

Cost reduction in machining processes requires means (software or hardware) that can lead to products having improved accuracy and quality, in continuously reduced machining time. Most low-cost procedures involve empirical knowledge and machine intelligence after properly modeling the process. The relevant literature shows that optimization methods can be coupled with manufacturing processes, either in production preparation or during manufacturing. Evolutionary algorithms, artificial neural networks, expert systems, fuzzy logic, etc. have been implemented in order to reduce decision related burden as both processes and parts become more complicated and experience does not necessarily lead to concrete solutions that optimize production.

New trends in machining operations optimization involve EAs used to compute optimal milling process parameters, for either analytical machining models or process-oriented models. In some cases, they are used for automatic G-code extraction [8], machining cost calculation stemming from various analytical cost functions [13], [2], [1], [7], optimal cutting conditions based on numerical and experimental data [3], etc. These applications were based on either custom EAs or ready off-the-shelf EAs, after determining the appropriate EA parameter values.

The optimization tool used in the present work, namely code *EASY* (**http://velos0.ltt.mech.ntua.gr/EASY/**), is based on generalized Evolutionary Algorithms (EAs), and it has been developed and brought to market by NTUA. It handles three population sets, with μ parents, λ offspring and ε elites and employs various recombination, mutation and elitistic operators, for single- and multi-objective optimization problems. The latter are treated through the Pareto front concept. *EASY* includes and optionally employs on-line trained surrogate models or metamodels which, in the context of the so-called *Inexact Pre-Evaluation* (IPE) phase, screen out badly performing population members and avoid a great amount of unnecessary exact evaluations leading to a considerable economy to computing cost. Constraints are handled through penalizing the objective function values. A detailed overview of the algorithmic features of *EASY* and its additional capabilities can be found in [4]and [5]. In the present work, the use of *EASY* was also extended in order to form a two-player Nash game. Each player is assigned half of the optimization variables, performs its own evolution while regularly communicating good solutions with the other player and the games comes up with a simple optimal solution, i.e. the so-called Nash equilibrium [12].

2 ROUGH MACHINING PROBLEM MODELING

Machining processes involve many parameters, each having a different impact on the product and the quality of the process. Though previous work has resulted in defining statistically significant process parameters with respect to the proposed quality characteristics [9], in this paper both significant and unimportant design variables have been accounted for in the model. This work addresses a few members of the family of sculptured surface parts, particularly those that are commonly manufactured through direct machining, as opposed to parts produced by using machined dies and processes such as casting, forging etc. The herein studied parts include a hip prosthesis, similar to those used in orthopedic surgery and two turbomachinery blades (Fig. 1); the first is an industrial compressor blade and the second is an extruded form of a 2D compressor airfoil.

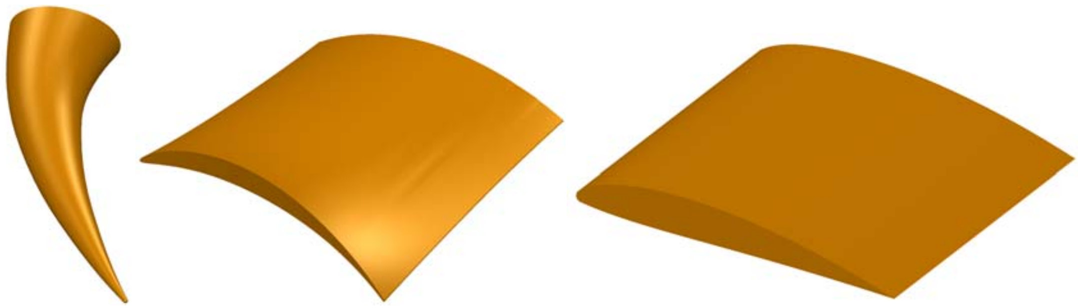

Figure 1: The three parts under study: a hip prosthesis and two turbomachinery blades.

For these parts 3-axis rough machining was implemented, using a commercially available CAM system. In many cases, sculptured surfaces are machined using 5-axis CNC milling machine tools, due to the requirement for constant normal (or near normal) positioning of the cutter axis and the surface tangent plane at the cutter contact point. However, 3-axis milling was considered sufficient, since 3-axis rough machining is sufficiently effective and does not exhibit orientation and collision problems often associated with 5-axis machining. Besides, surface finish is not an objective in roughing, as it aims at substantial material volume removal from an original raw material block at a reasonably high removal rate.

The machining parameters (corresponding to the optimization variables) used in the present work are:

1. Tool machining the sculptured surface part. For the present study, flat-end mills were employed due to their efficiency and increased productivity in 3-axis milling.
2. Stepover: the distance in the x-y plane between two successive passes of the tool.
3. Thickness: the mean distance between the part and the cutting edge of the tool.
4. Stepdown: the distance between two successive passes of the tool in the z direction.
5. Profiling: tool pass around the profile that cleans a specific z-slice.
6. Raster angle: the angle of the raster pattern relative to x-axis.
7. Allowance: offset surface of the machined part that the tool never penetrates.
8. Infinite range: the distance that a tool travels between successive passes. “Infinite” means that any distance is acceptable for joining successive passes and these are produced automatically by the software.
9. Feedrate: the relative (cutting) velocity between the tool and the part.
10. Spindle speed: the rotating speed of the machine tool spindle in rpm.

A tool database of flat-end mills was built, consisting of eight different tools with diameters ranging from 20 to 100 mm. These tools were selected from the SECO® tools catalogue [15]. The use of a specific tool database aims at simulating industrial practice, where a certain number of tools are available. In addition, any set of available tools can be used, as long as they have properly been registered in the database. This particular tool diameter range was chosen according to the parts' dimensions and operations the tools are to perform. Apart from tool geometry and dimensions, the database includes technical characteristics, such as insert types, number of inserts, power constraint, and a recommended value ranges for feedrate and cutting speeds etc. These characteristics are associated with each corresponding machining tool and used in the milling strategy definition.

In order to accurately model the rough machining problem, it is essential to define all the physical and technological constraints. A typical example concerns machining power demand being bounded by the available machine tool motor. These are referred to as technological constraints and implemented before the evaluation. The technological constraints are modeled through equations and value limits in the application and, thus, a solution is penalized if it lies outside the predefined field. Quality constraints refer to the quality of an objective with respect to the pre-set limits, e.g. total machining time is assigned upper and lower bounds, as a result of experience. Quality constraint checking is performed after each evaluation.

Equation (1) expresses cutting power demand, P_c (in KW), [15], as function of some of the machining parameters. Part material for the present study was Al belonging to material group 16 in [15], leading to the choice of tool insert. Cutting conditions have to be chosen so as not to conflict with tool characteristics, e.g. tool depth of cut, maximum rpm, etc.

$$P_c = \frac{a_p \times a_e \times v_f}{60000000 \times \eta} \times k_c \tag{1}$$

where

a_p = depth of cut (mm)

a_e = tool engagement (mm)

v_f = cutting speed (mm/min)

η = efficiency of machine spindle motor

k_c = cutting force per surface unit (N/mm²) depending on the part material type.

In general, in machining processes, it is desired to obtain a final machined product with certain properties. Main desired properties comprise surface finish, dimensional tolerances in critical parts of the product, etc., provided that machining time or tool lifts are as small or as few as possible, etc. Thus, apart from results related to the product itself, quality requirements for machining processes involve process characteristics and performance, because of productivity demands imposed by industrial manufacturing. Nowadays, CAM systems offer a variety of functionalities and advanced features that improve performance of CNC machine tools in production. However, built-in optimization algorithms in either CAM systems or CNC machine tool control software operate as "local" optimization routines; although they optimize given functions, they fail to offer optimized solutions for the machining process, which is left to the CNC machine tool operators or the G-code programmers.

In most cases, CAM system users must develop their own software in order to fully exploit the system's capabilities and to tailor it according to their production needs. In this sense, the present authors developed an optimization environment that interacts with a CAM software, feeding it with proper input, calculating functions and extracting information regarding the machining statistics, e.g. machining time, number of tool lifts, toolpath length, etc, and the resulting product geometry. The outputs were utilized to evaluate machining parameters efficiency in obtaining the machining goal that was decided beforehand. In the present study, the goal was twofold; to minimize (a) the machining time and (b) the remaining material volume after machining simulation.

3 OPTIMIZATION METHODS

In multi-objective problems, the most popular optimization principle is Pareto optimality. There are several variations and methods [16], [14], which are based on this principle. In the same sense, there are previously proposed methods which achieve a reduction of the computational cost, notably artificial neural networks (ANNs) [4]. In the present work the multi-objective optimization was based on conventional EAs, EAs supported by ANNs and the, still EA-based, "n-player game" (n – number of objectives) - introduced by John Nash [11]. The principle of Nash theory is a non-cooperative game in which n different players strive to optimize n different objectives. The same optimization problems were addressed through the use of two methods which produce a Pareto front (EAs, EAs assisted by IPE) and a method which produces a single point (EAs – Nash, equilibrium point).

The three optimization approaches have been implemented using the EA optimization software EASY. The optimization software was used in combination with the evaluation software of geometry nature (Fig. 2). The evaluation process incorporates machining simulation software which comes up with the machining time and a model in the form of a surface triangular mesh representing the rough machined part. The remaining material volume is calculated by subtracting the volume of the final sculptured part from the rough machined model's volume.

The optimization software stands for a generalization of Genetic Algorithms and Evolution Strategies with several add-on features and, for this reason, it will be referred to as an Evolutionary Algorithm. The EA is characterized by 3 numbers (μ, κ, λ) which denote the evolution from the parent population of μ individuals to the offspring population of λ individuals, while allowing maximal life span of individuals equal to κ generations. A set of ε elite or archival individuals is maintained during the evolution.

A comparison is made between conventional EAs, EAs assisted with computational cost reduction techniques [5] and an EA based on Nash's theory for a two-player non-cooperative game. The formation of the two-player game required additional software for handling the communication between players, the assignment of variables and objectives. The design variables and objectives are divided between the two players and each player performs its own search based on its own objective function with the values of the other player's variables kept fixed. Upon completion of the so-called cycle, during which there is no communication between players, these exchange the best so-far computed values, Fig. 3. Thus, the optimizer is invoked twice during each cycle, once for each player. The next cycle continues after replacing the fixed values of the previous cycle with the updated ones provided by 'the other' player. The optimization process ends up with convergence to the Nash equilibrium point. Although the concept of merging EAs and Nash theory (NT) has also been used in the past

[12], it is useful to make a comparison with other more tested techniques.

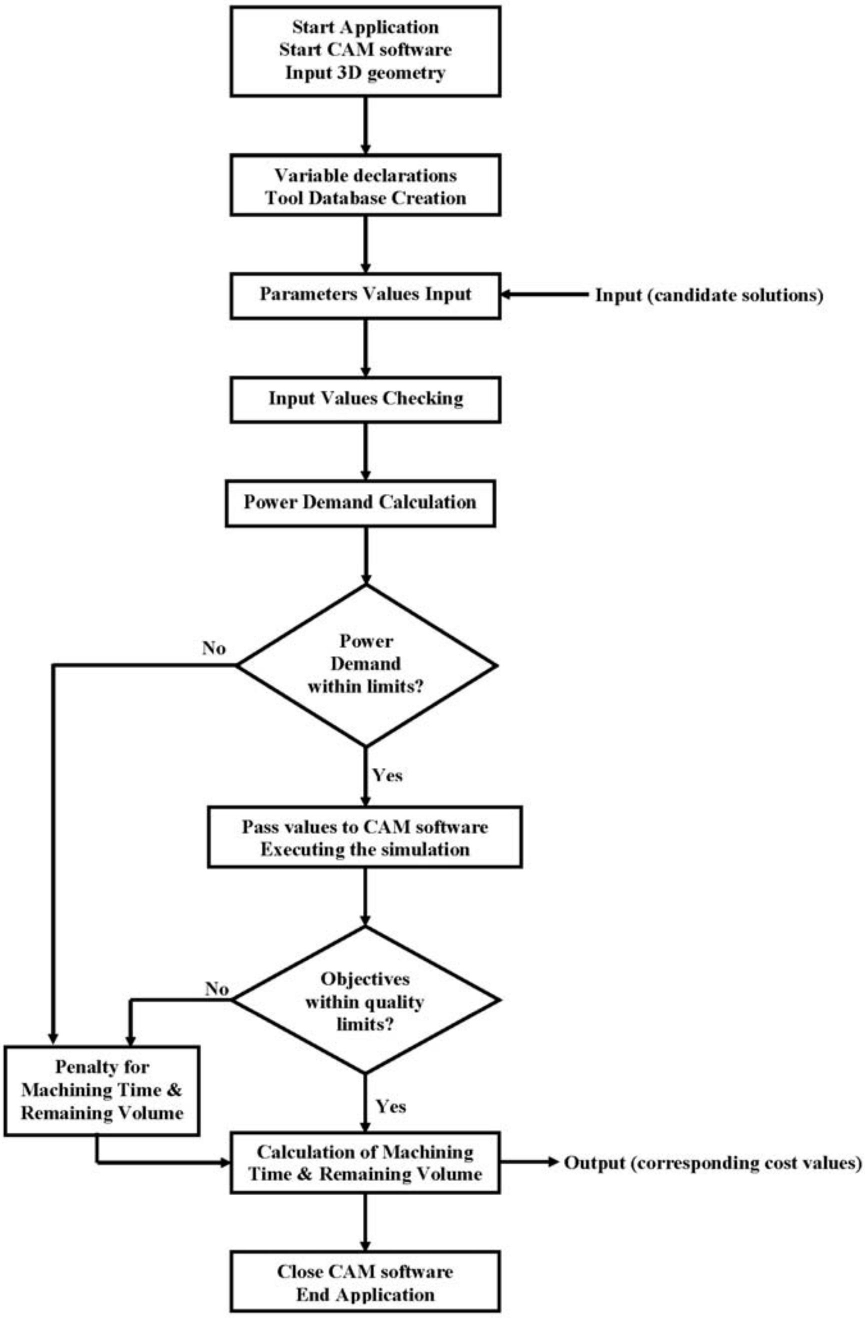

Figure 2: Flowchart of the evaluation steps that need to be carried out for each and every candidate solution, regardless of the optimization method used.

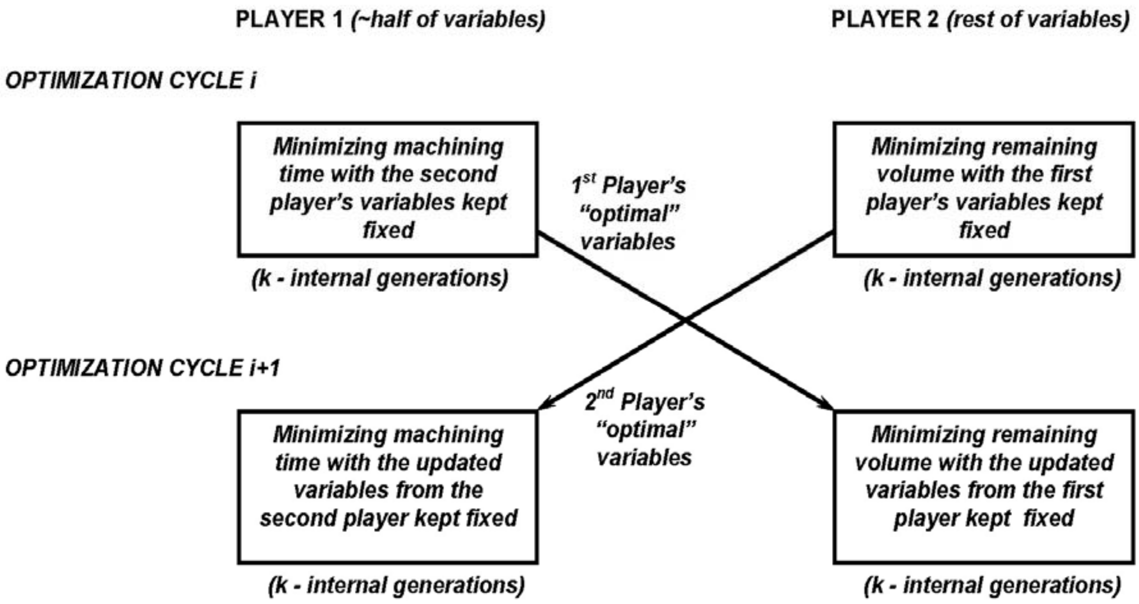

Figure 3: Two-player Nash game: exchange of information after search tasks held in isolation.

4 PRACTICALITIES

All variables handled by the EA were binary encoded. Taking into account the 10 free variables of the problem at hand, a suitable population size was selected, with $\mu = 10$ parents, $\lambda = 50$ offspring and $\varepsilon = 20$ elites. The number of parents, offspring and elites was the same in all test problems.

Concerning the utilization of the *Inexact Pre-Evaluation* (IPE) phase, local Radial Basis Function (RBF) [6] networks were used as surrogate evaluation tools with the number of exact evaluations per generation varying between 10 and 25. The IPE technique was activated after having evaluated 200 individuals, generated during the first EA generations.

During the Nash game optimization each variable was assigned to a single player. The variable assignment and their lower and upper bounds are depicted in Table 1. Variable IDs correspond to the assignment of Section 2. Among the 10 design variables there are those which take on integer values, encoded using 1-3 bits each (Vars. 1, 5, 8) and those which take on real values encoded with 10 bits each (Vars. 2, 3, 4, 6, 7, 9 and 10). The variable bounds and their assignment to players was the same in all test problems.

During the two-player game, a choice was made about the number of internal single objective generations; in the present work, this is equal to five. Any number larger than that increased significantly the computational cost and any number less than that did not achieve a significant proximity to the Pareto front.

Variable ID	Lower Bound	Upper Bound	Player 1	Player 2
1	1	8	x	
2	0.1	99.9		x
3	0.5	3.0		x
4	0.1	6.0	x	
5	1	4	x	
6	0.1	90.0	x	
7	0.1	2.7		x
8	0	1		x
9	0.0	1.0		x
10	10.0	6000.0	x	

Table 1 : Variables' value fields and player assignment.

During the machining process, each turbomachinery blade was treated as made of two sculptured parts, one forming the pressure side and the other the suction side. Thus, all in all, five (5) optimization problems had to be solved.

The chosen machining strategy was raster machining, as this is the most common strategy in 3-axis operations (see Fig. 4). Note that the excessive material at the two sides of the sculptured part represents part clamping on the machine tool bed. Software developed by the authors controls the CAM system while it passes arguments and drives the software to execute machining functions. Moreover, it performs checks for all the arguments ensuring consistency of the code and realistic results. The application inputs ten values that define a certain machining strategy and outputs two values, as mentioned before (machining time and remaining volume).

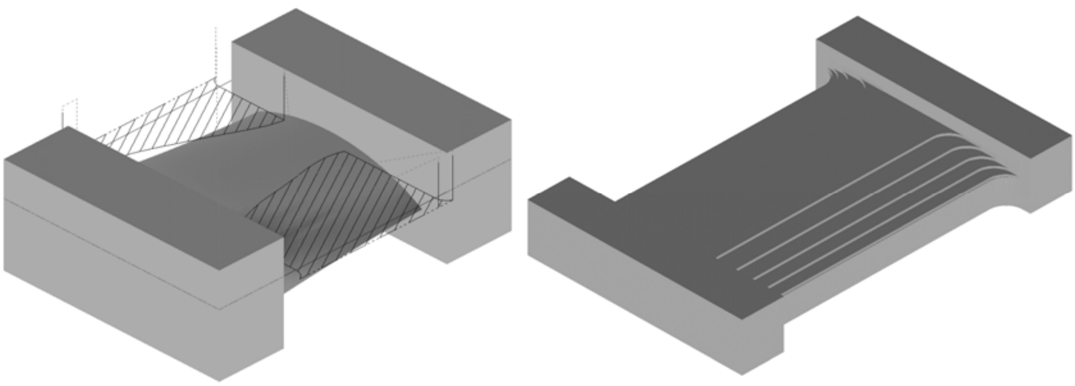

Figure 4: (a) Raster toolpath strategy at a given z-height for the suction side of the turbomachinery blade. (b) A typical rough machining result.

5 METHOD APPLICATION – RESULTS AND DISCUSSION

The optimization process, when based on standard EAs, is quite costly as it takes up to one full day run on a Pentium IV computer. Aiming at lowering the overall optimization time, the process was streamlined through the *Inexact Pre-Evaluation* (IPE) phase in order to avoid a great amount of unnecessary computations. Through the use of IPE, the number of exact evaluations is significantly reduced.

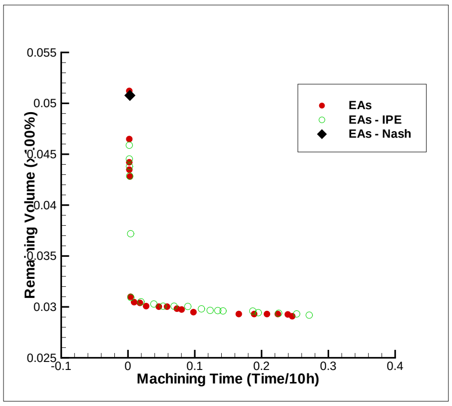

Figure 5: Hip prostheses testcase: Nash equilibrium along with Pareto fronts (EAs -3018 evaluations, EAs – IPE 2220 evaluations, EAs – Nash 1550 evaluations)

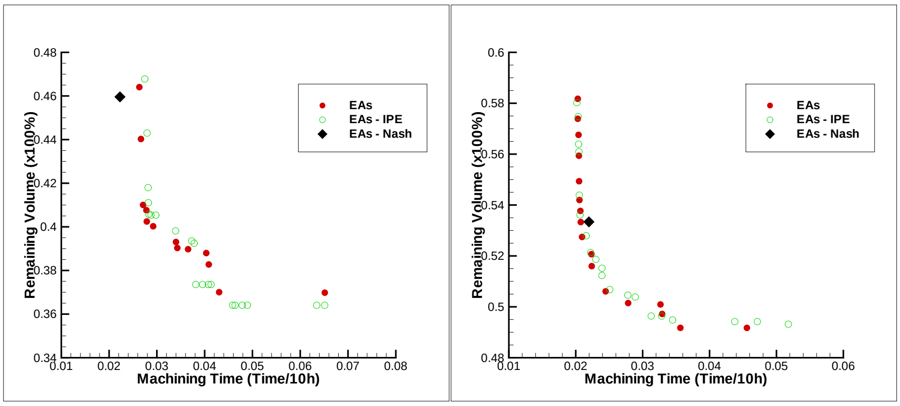

Figure 6: Industrial compressor - Nash equilibrium and Pareto fronts– (a) Pressure side(EAs -2418 evaluations, EAs – IPE 1185 evaluations, eAs – Nash 1050 evaluations), (b) Suction side(EAs -3000 evaluations, EAs – IPE 1929 evaluations, EAs – Nash 1050 evaluations).

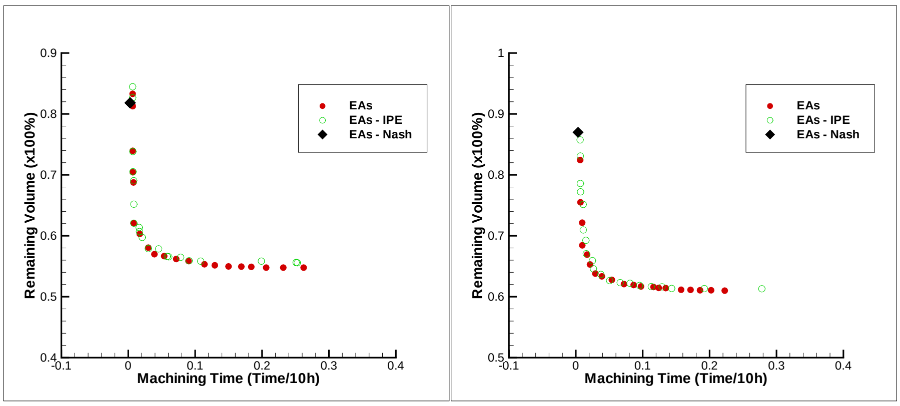

Figure 7: Extruded 2D turbomachinery blade - Nash equilibrium and Pareto fronts– (a) Pressure side(EAs -2447 evaluations, EAs – IPE 1282 evaluations, EAs – Nash 1050 evaluations), (b) Suction side(EAs -2439 evaluations, EAs – IPE 1165 evaluations, EAs – Nash 1050 evaluations).

Through the use of the optimization tool for the design of the machining process, a Pareto front was computed for each of the five test cases. Results are depicted in Fig. 5-7 and compared with the performance of EAs assisted by IPE. The gain by the IPE assisted code is obvious as the Pareto was captured after a fraction of the computational time that is required for the evaluation of each individual of the population. The number of evaluations required through the use of IPE ranges from 40 up to 65 per cent of the total evaluations that would be required without the use of IPE.

The results of the two-player game implementation can be evaluated by comparing the Nash equilibrium point with the corresponding Pareto fronts. As shown in Figs. 5-7, the two-player game has an advantage over typical EAs with respect to computing cost. The convergence to the Nash equilibrium point is very fast resulting in a solution compared to the Pareto fronts which were computed after twice as many evaluations. When the two-player game is compared with the Pareto fronts computed through EAs assisted by IPE, the computational costs for both are of the same order.

In order to compare Pareto fronts computed through the use of EAs with and without IPE a “front of fronts” was formed by merging the two fronts and eliminating dominated solutions. In Figs 8-9, a magnified view of the “front of fronts” for the hip prostheses testcase can be seen along with the fronts for the extruded airfoil testcase.

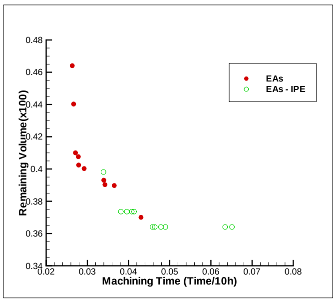

Figure 8: Hip prosthesis part testcase - Computed “front of fronts” (EAs + EAs-IPE)

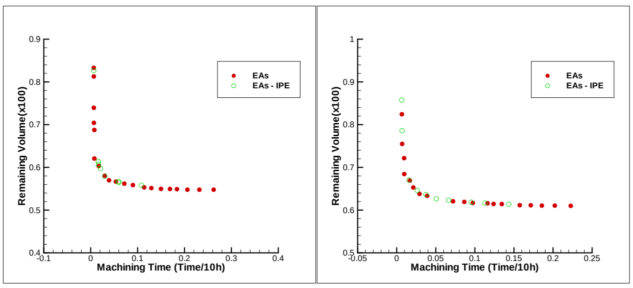

Figure 9: Extruded 2D turbomachinery blade testcase – Computed “front of fronts” – (a) Pres. side, (b) Suc. side.

The optimization methods used result to a set of optimal machining solutions for each of the studied cases. In each of the above diagrams, there are three discrete areas of solutions as far as rough machining is concerned; the first is the series of solutions appearing parallel to the “volume” axis, the second parallel to the “time” axis, and the third near the origin. When these optimal solutions are translated back in terms of machining process parameters, these three areas clearly share common characteristics. The first area refers to machining with the goal of minimal machining time, the second with the goal of minimal remaining volume. In the third area, arbitrary solutions dwell, which simultaneously minimize machining time and remaining volume. The first two areas appear to be connected to a different cutting tool varying from case to case, while the third area favors greater variety of tools. Moreover, some variables seem to reach a specific value and vary very little around it, a typical example being spindle speed, with encountered values in the range 5700 to 5950 rpm. Other variables vary across their whole range in all three areas.

As can be seen in Figs 5-7, the Nash equilibrium point seems to lie in the first area, where machining time is minimal. The criterion for distributing the input variables to the two players was near-equality of the sum of the 'statistical influences' of each player’s variables, as this was calculated in [9]. It is likely that, for a different distribution of variables, if one player takes up all the influential and the other all the non-influential variables, the Nash equilibrium point might have lied in other areas of the Pareto fronts. These three areas allow the machinist to choose a (group of) solution(s), in order to achieve his goal. Should this goal be more oriented towards one of the two directions – reflected by the two objectives, then, it would be wise to properly weigh the objectives. E.g. if machining time minimization is the critical and remaining volume is a secondary target, (this might be the case if a semi-finishing machining operation follows) a candidate weighing could be 2/3 and 1/3 for the two objectives, respectively. The choice of proper weights is dependent on the machinist’s experience and his suggestions on the process planning.

6 CONCLUSIONS

The use of optimization methods for rough machining parameter selection enables realistic and practical application of the latter. The use of EAs offers the ability to choose from a variety of non-dominated solutions. Concerning computational cost, the use of IPE improves performance a lot. Nash game is as fast as EAs with IPE, but results in a single equilibrium point. In the particular applications studied, EAs with IPE proved to be a practical tool offering a variety of possibilities linked to the user's personal preference towards less remaining material volume or less machining time. This methodology can be extended to finish machining of sculptured parts using an appropriate process model, redefined objectives and new quality characteristics.

ACKNOWLEDGEMENTS

This work was funded by the PENED01 program (Measure 8.3 of the Operational Program Competitiveness, of which 75% is European Commission and 25% national funding) under project number 01ED131.

REFERENCES

[1] Amiolemhen, P.E., Ibhadode, A.O.A.: Application of genetic algorithms—determination of the optimal machining parameters in the conversion of a cylindrical

bar stock into a continuous finished profile, International Journal of Machine Tools & Manufacture, Vol. 44, pp. 1403–1412, 2004.
[2] Cus, F., Balic, J.: Optimization of cutting process by GA approach, Robotics and Computer Integrated Manufacturing, Vol. 19, pp. 113–121, 2003.
[3] Davim, J.P., Antonio, C.A.C.: Optimisation of cutting conditions in machining of aluminum matrix composites using a numerical and experimental model, Journal of Materials Processing Technology, Vol. 112, pp. 78-82, 2001.
[4] Giannakoglou, K.C.: Design of Optimal Aerodynamic Shapes using Stochastic Optimization Methods and Computational Intelligence, Progress in Aerospace Sciences, Vol. 38, pp. 43-76, 2002.
[5] Giannakoglou, K.C., Giotis, A.P., Karakasis, M.K.: Low-Cost Genetic Optimization Based on Inexact Pre-Evaluations and the Sensitivity Analysis of Design Parameters, Inverse Problems in Engineering, Vol. 9, pp. 389-412, 2001.
[6] Haykin, S.,Neural Networks: A Comprehensive Foundation, 2/E, Prentice Hall, 1999.
[7] Khan, Z., Prasad, B., Singh, T.: Machining condition optimization by genetic algorithms and simulated annealing, Computers Ops Res., Vol. 24, No. 7, pp. 647-657, 1997.
[8] Kovacic, M., Brezocnik, M., Pahole, I., Balic, J., Kecelj, B.: Evolutionary programming of CNC machines, Journal of Materials Processing Technology, Vol. 164–165, pp. 1379–1387, 2005.
[9] Krimpenis, A., Fousekis, A., Vosniakos, G.: Assessment of sculptured surface milling strategies using Design of Experiments, International Journal of Advanced Manufacturing Technology, published on-line, Springer Verlag, 2004.
[10] Krimpenis, A., Vosniakos, G.C.: Optimisation of roughing strategy for sculptured surface machining using genetic algorithms and neural networks, 8th International Conference on Production Engineering, Design and Control 2004 (PEDAC2004), Alexandria, Egypt, 2004.
[11] Nash J.F., Equilibrium Points in n-person games, Annals of Mathematics, 54:289, 1951.
[12] Sefrioui, M., Periaux, J.: Nash genetic algorithms: Examples and applications, *IEEE International Conference on Evolutionary Computation*, Vol. 1, pp. 509-516, 2000.
[13] Shunmugam, M.S., Bhaskara Reddy, S.V., Narendran, T.T.: Selection of optimal conditions in multi-pass face-milling using a genetic algorithm, International Journal of Machine Tools & Manufacture, Vol. 40, pp. 401–414, 2000.
[14] Srinivas, N., Kalyanmoy, D.: Multiobjective optimization using nondominated sorting in genetic algorithms. Technical report, Department of Mechanical Engineering, Indian Institute od Technology, Kanpur, India, 1993.
[15] www.secotools.com
[16] Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization, in K. Giannakoglou, D. Tsahalis, J. Periaux, P. Papailou and T. Fogarty (eds.) EUROGEN 2001, Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems, pp. 95-100, Athens, Greece, 2002.