Proceedings of the 2015 Winter Simulation Conference
L. Yilmaz, W. K. V. Chan, I. Moon, T. M. K. Roeder, C. Macal, and M. D. Rossetti, eds.

A TECHNICAL APPROACH OF A SIMULATION-BASED OPTIMIZATION PLATFORM FOR SETUP-PREPARATION VIA VIRTUAL TOOLING BY TESTING THE OPTIMIZATION OF ZERO POINT POSITIONS IN CNC-APPLICATIONS

Jens Weber

Business Computing, esp. CIM
Simulation & Manufacturing
Heinz Nixdorf Institute
University of Paderborn
Fuerstenallee 11
D-33102, Paderborn, GERMANY

ABSTRACT

The automatic setup process using simulation-based optimization to provide production parameter sets such as workpiece position, zero point position, and tool range requires high evaluation effort including multiple virtual tooling machine simulation runs to find possible setup solutions. In this contribution, an approach is demonstrated for a general zero point optimization in combination with a NC-parser application for the fitness portion of the optimization system embedded as a preprocessing unit. This preprocessing step leads to near-optimal zero point position coordinates without the huge need for resources or time caused by a high number of iteration runs, which includes complete simulation runs of the virtual tooling machine. This preprocessing optimization unit is implemented using an extended population-based optimization technique to support work preparation, planning departments and industrial project managements.

1 INTRODUCTION

Today, the area of business computing offers a variety of methods to solve a wide field of complex problems for operation research, logistics management, optimization, and scheduling. Typical problems are layout planning, production networks, lot sizing problems, the Traveling Salesmen Problem, decision support, and so on. Most of these named problems are projected onto mathematical models which lead to highly complex solutions and computing time. Heuristics, metaheuristics or simulations are used to find a near-optimal solution including adequate computing time within a given set of restrictions. Another strategy to find an adequate solution is the combination of the optimization with simulation techniques which use the simulation as an evaluation step for the optimization run results (see Laroque, Urban, and Eberling 2010; Suhl, and Mellouli 2006).

The simulation-based optimization method has been used in several areas of research and industry as an useful and simple optimization tool (e.g. material flow systems). From these implementations the current research idea arose to use this approach in the field of tooling machine setup preparation for production systems based on a virtual tooling machine without the need for complicated mathematical optimizations or statistics-based design of experiments (see Kleijnen 2008).

The virtual tooling machine is a 1:1 simulation model of a real machine, offering many features (i.e. verification of setup parameters and NC-programs, collision detection between workpiece and machine as well as material-removal simulation). The optimization portion contains population-based optimization techniques (metaheuristics). The current research project focuses on automatic setup configuration using machine simulations. The usage of a simulation-based optimization loop offers manual machine parameter, production configuration and NC-program verification, which are then examined by the virtual machine simulation. Machine parameters and parts of the production configuration could be the zero point position for the NC-program coordinates, workpiece position in the workspace, tool arrangements or spindle speed etc.. The goal is to provide useful configuration and parameter sets for the machine setup which are made automatically available for following production processes, without a high effort of re-generating the configuration and setup information.

1.1 Problem Description

The problem with using simulation-based optimization in this described context is that it leads to a high level of effort due to the required high number of iterations to check the availability of automatically generated parameter sets. It is fact that each test iteration (via SBO-combination) requires a single simulation run using the virtual tooling machine which would lead to a high duration time or computational time. For real-world use cases, which must handle complex workpiece geometries and NC-programs, this approach seems unable to achieve a solution within an adequate amount of time. This contribution shows an approach including use case within the current research project to solve the described problem.

Due to these described circumstances, the simulation-based optimization loop in combination with the 1:1 simulation model was further developed by using an NC-parser as a so-called “preprocessing” agent. The NC-parser interpolates the tool paths and is able to estimate the runtime, which supports the parser implementation as a fitness function. This submission shows the implemented system and the first use case results for optimizing the zero point position of a digital workpiece. The change of the parameter of the zero point by optimization affect equally the parameter change of the workpiece position in the work space of the tooling machine.

Section 2 of this contribution presents the related work of this research area and introduces the reader to the research project and scenarios. Section 3 deals with the current research level, the simulation-based optimization loop and facts concerning the setup optimization system for the zero point position as well as information about the fitness landscape used. Section 4 presents the use case results and section 5 offers conclusions and an outlook for future research.

2 RELATED WORK AND RESEARCH PROJECT

2.1 Related Work of Simulation-Based Optimization via Metaheuristics for Different Domains

Simulation-based optimization is successfully used in the domain of logistic and production planning, esp. material flow planning (see März et al. 2001). For the implementation of simulation-based optimization, there are no standardized procedures and methods although the used algorithms and methods are familiar. Nevertheless, the development of the simulation model as well as the parametrizing of the optimization part requires often individual configurations (see März et al. 2001; Wenzel et al. (2007); Rabe, Spieckermann, and Wenzel 2008).

The contribution of Laroque, and Pater (2012) consists of a material flow simulation using simulation-based optimization containing a population based optimization techniques (Particle Swarm Optimization, PSO) with the research goal to reach a rapid convergence of the used metaheuristic and offer a useful configuration. The contribution allows stochastic handling which impacts the complexity of the process and the shown approach represent a practical use for industrial problems. A related

contribution is presented by Laroque, Urban, and Eberling (2010) which contains the feasibility study about parameter optimization via simulation-based optimization. The approach presents the use of particle swarm optimization-algorithm to attain a rapid and usable model configuration of a material flow model, which is interesting when conventional mathematical modelling requires high computational resources for the optimization (see Laroque, Urban, and Eberling 2010).

In the contributions of Laroque, and Pater (2012) and Laroque, Urban, and Eberling (2010), it was concretely presented that the material flow models are able to parameterized with support of the PSO-algorithm as optimization component within a restricted range of prototypical applications. This prototypical approach was compared with the further developed 3-phases-PSO (see Angeline 1998) and the result was the detection of an improved runtime.

Furthermore, Laroque et al. (2012) show by using PSO-algorithm, genetic algorithm and the treated "island-concept" a fast converging procedure. The implementation allows automated distributed simulation runs. For this contribution, a material flow simulation model act as reference model and the results demonstrate the advantages of parallel and combinatorial processing. The goal is to set up a production system with capacity restrictions and minimized costs. The production system originates with machine selection from several machines which differ in total costs, machine hour rate as well as cycle time and administered penalty cost if a restriction has been overstepped. This approach leads to optimal parameter values for the simulation model and rapid convergence (see Laroque et al. 2012).

To ensure the usage of metaheuristics for the simulation-based optimization approach to improve setup of virtual tooling, the contribution of Weber, Boxnick, and Dangelmaier (2014) presents evaluations of certain metaheuristics, also with regard to the convergence behavior, the runtime comparison and the number of the dimensions. For this contribution, the metaheuristics are embedded into an experimental design and evaluation was carried out by using a benchmark function. The results were collected for a further use. The comparison of several metaheuristics pointed out how certain metaheuristics change their behavior when faced with changed increasing complexity.

Finding the best parameter for the simulation-based optimization approach in combination with real-life simulation of virtual tooling, a long computation time is required, which is impractical for industrial application and leads to research efforts towards achieving higher speed and improved usage of resources. For this reason, the idea of distributed and parallel simulations arose. One approach for a faster process is presented in Reisch, Laroque, and Schröder (2014) which contains a successful dimension reduction strategy for using particle swarm optimization (PSO). The idea was to shrink the search space size during the search procedure. The approach was performed by using Monte Carlo simulation of magnetic systems. The contribution of Reisch et al. (2015) contains a further method to accelerate the optimizations system of the simulation-based optimization approach via metaheuristics as well as handle stochastic node failures and distributed and limited resources. The approach shows a method to further develop and configure the PSO-algorithm as the optimization part of the simulation-based optimization loop. The result was a successful implementation of a simulation-based optimization system based on an asynchronous PSO-algorithm and a partially synchronous PSO-algorithm (see Reisch et al. 2015). The evaluation takes into account via benchmark function such as shown in the contribution of Weber, Boxnick and, Dangelmaier (2014).

As the related work presents, there are many research activities in the area of optimization using simulation-based optimization. In the further subsection, the used particle Swarm optimization-algorithm is shortly presented.

2.1.1 Particle Swarm Optimization (PSO)

The particle swarm optimization-algorithm is a population-based and evolutionary computation technique developed by Eberhart, and Kennedy (see Kennedy, and Eberhart 1995). This algorithm follows the general steps described below:

- PSO initialized with a population of random solutions called particles

- Search for an optimum by updated generations
- The population evolves based on the immediately preceding generation
- The particles "fly" through the solution space by following the current optimal particle

For the standard PSO which is used in this contribution, the updating step of the new position and velocity of the particles in the solution spaces is performed by using the formulas below (see Shi and Eberhart 1998; Reisch et al. 2015):

$$v_i = \omega_{inertia} * v_i + \beta_1 * \omega_{cognitive} * (x_{globalbest} - x_i) + \beta_2 * \omega_{social} * (x_{localbest} - x_i) \quad (1)$$

$$xi = vi + xi \quad (2)$$

The velocity vector of each particle i is denoted by v_i (1) and x_i describes the corresponding position of particle i within the solution space (2).

Today, related research works and projects show that there are many variants of the PSO-algorithm, for example the further development from Angeline (see Angeline 1998; Shi, and Eberhart 1998), so it is clear that the idea of using a modified PSO approach is not new, as shown by examples such as contributions from Hsieh, and Horng (2012) as well as from Koh et al. (2006). The last two contributions deal with asynchronous, parallel and discrete modified PSO-algorithms used to enhance computational efficiency.

As we can see, there are many approaches using simulation-based optimization, especially using population-based metaheuristics as an optimization component in the area of material flow management, virtual tooling, and related sciences which require a fast system and high computer performance.

2.2 Research Project and Integration of This Contribution into the System

The following section is an overview to better understand the context in which the used approach was researched and developed.

The research project is part of the so-called "Spitzencluster it's OWL", which is separated into several subprojects. The subproject that served as a basis for this contribution contains the development of a work-preparation support platform based on production verification via virtual tooling machines to improve job planning and project management. An expansion upon this idea is to provide this platform as cloud-computing approach (see it's OWL 2015).

To achieve the original research goal of the project, the idea arose that the work preparation platform could provide a cloud-based service platform for industrial companies. With support from this platform, the user can control the work preparation and feed a database via web interface with production dates, machine data, tool information, and jobs. Figure 1 presents an overview of several important elements of the platform.

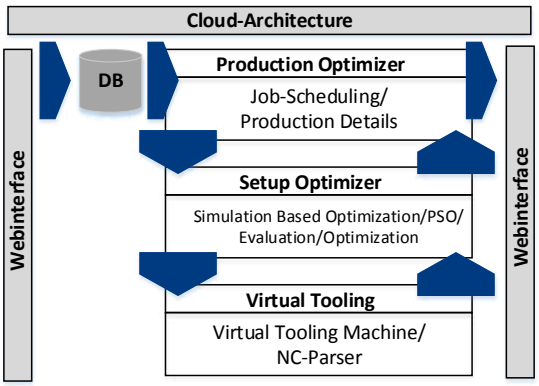

Figure 1: Overview about several system elements of the work preparation platform.

The Production Optimizer step schedules and optimizes several production jobs as well as giving necessary production details for the "micro-optimization" of the production line containing (virtual) tooling machines. A mathematical lot sizing model sits behind the Production Optimizer, which is solved by mathematical methods. Further details about this step were not considered relevant to this contribution.

The Setup optimizer and the virtual tooling element use the described simulation-based optimization method treated in this contribution. The virtual tooling element of the system consists of an interface to the virtual tooling machine simulation as well as an interface to the NC-parser. The combination of Setup Optimizer including simulation-based optimization with the NC-parser represents the tools for the zero point position optimization as an use case. The evaluation of the optimization results leads to a high number of simulation runs via virtual tooling machine for each optimization result. For every evaluation, the machine must normally also be prepared, but with support from the NC-Parser, the simulation runs are currently more practical, because the high number of simulation runs via virtual tooling machine can be avoided and the particle evaluations require no necessary labor-intensive configuring. For the system tests the zero point of the workpiece depending on the constantly changing position of the tool for the represented machine is a promising avenue and use case, due to the disappearance of the need to optimize the zero point as required for a full machine simulation (zero point position implies equally workpiece position).

After the optimization runs end, the information and coordinates are given back to the user via user interface and can be saved back into the database for further production processes. For a project manager, a production manager as well as the planning department, organizations earn an easier way to perform setup preparation without wasting production time, resources and materials.

3 CONCEPT FOR USING SIMULATION-BASED OPTIMIZATION (SBO) VIA PREPROCESSING TO OPTIMIZE THE ZERO POINT OF THE WORKPIECE

3.1 System Architecture of the Simulation-Based Optimization Loop

For a successful implementation of the simulation-based optimization, especially in handling the zero point optimization via PSO-algorithm, a clear overview of the system architecture is necessary. A given constantly changing position of the machine-tool and a zero point start position for the workpiece is loaded for the optimization either as direct-input to the database or via a web interface.

Each new coordinate for the zero point position is regarded as a particle in the solution space for the PSO-algorithm which varies based on the workspace of the tooling machine. Each particle consists of several dimensions. For a three-axis tooling machine, the zero points have three coordinates, each for every linear axis which leads at least to $n = 3$ dimensions. A five-axis tooling machine consists of three linear axis and two turning axis, so that there are 5 coordinates which means the maximum dimension size is $n = 5$.

The zero point of the workpiece is defined as point of origin for the workpiece and it is independent of the workpiece position within the workspace of the used machine. But if the position of the zero point changes, the position of the workpiece will change too. Normally, the tool has to be returned at this point after each production cycle (see Wellers, Kerp, and Lieberwirth 1983). New machines and systems avoid the return to the point of origin after each cycle.

If the work preparation is a manual operation, the zero point will be placed as simply as possible for the machine operator. The individual position of the zero point with regards to the automatic optimization of the production time could lead to circuitous routes for the tool during and after the production cycle which will be minimized with the support of the shown system (see Figure 2) and the use case (see Section 4).

Schematic of the system: An initial parameter set for a zero point coordinate will be loaded into the optimization system which contains the PSO-algorithm. In the presented use case, the PSO is a fully synchronous PSO-algorithm. The asynchronous PSO-algorithm, such as the one treated in Reisch et al.

(2015), is not necessary because there is no risk of stochastic node failures or any cases of limited resources.

Depending on the dimension and particle size, which have significant influence on the computational time, a near-optimal particle is calculated and the coordinates of the particle will be evaluated by using the NC-Parser. The NC-Parser will interpolate the driven tool path during production depending on the used NC-program. The NC-parser calculation is based on LaGrange's equation of motion. An example of a constant standard NC-program was used for the conducted experiments. The program runtime is measured by adding the sum of the path difference from the evaluating zero point position coordinates to the constant defined tool change point position coordinates from the virtual tooling machine (NC-parser). The optimal coordinate for the zero point leads to the minimized runtime of the program which is equivalent to the near-optimal tool path, which is also given by the NC-Parser.

After the particle evaluation, a new parameter set will be initialized until the stopping criteria is reached. The last and near-optimal zero point coordinate can be saved to the database for subsequent production processes without performing a parameter optimization again. Figure 2 presents the simulation-based optimization loop. The elements of the Setup Optimizer and virtual tooling are labeled (see. Section 2.2, Figure 1).

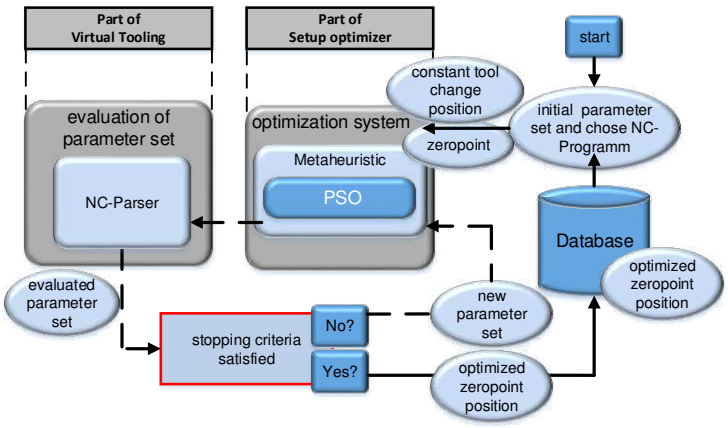

Figure 2: Schematic simulation-based optimization loop of optimizing zero point via NC-Parser interface.

3.2 Fitness Landscape

As a fitness element and synchronous part of the virtual tooling element (see section 2.2), the NC-Parser serves as an evaluation component for the optimized particles from the PSO-algorithm. The basic idea is to minimize the program runtime which goes hand-in-hand with the optimization of the zero point position depending on the constant tool change point position coordinates. The full fitness landscape is given in order to better understand the results from the zero point optimization as an use case. This landscape will represent the solution space if the NC-Parser calculates every possible coordinate within the solution space.

The ranges for the linear axis x and y were chosen to be the following: x ϵ [3000; -3000] and y ϵ [3000;-3000]. The fitness landscape for the x and y-axis is equivalent to the third linear axis (z-axis) in the case of 3-axis (3-axis machine), as well as two turning axis in the case of a 5-axis tooling machine and it is for sufficient illustrative purposes. The vertical axis of the fitness landscape plot represents the given program runtime as a fitness value. The units of the x and y-axis (also z-axis) are millimeters [mm], and the turning axis must be calculated in degrees or radians. The fitness value is given in milliseconds [ms].

The constant tool change point position coordinates is given as x = -1000, y = -500, z = -500. The value of the z-axis (z = -500) is not illustrated in the fitness landscape (see Figure 3), but it is taken into the account during the optimization process.

Because of the minimized runtime (fitness), the optimal solution will be close to the area of the tool change point position coordinates, which is recognizable in Figure 3. The funnel-shaped sloping towards the near-minimum area of the fitness landscape seems to be close to the defined point. To prove this theory, if the developed system works, the convergence behavior of the PSO-algorithm and the identified zero point optimum coordinates should converge in the direction of the lowest point in the funnel-shaped area (see use case details in section 4). If this scenario occurs, the PSO will be practically configured and the NC-parser will offer an adequate substitution for the virtual tooling machine for the optimization of the zero point, which then supports the preprocessing idea (see section 1). For that, it is very important to understand that the collision between machine parts and workpiece do not taken into account during the preprocessing. Figure 3 shows the fitness landscape with the funnel-shaped minimization area using the tool change position ranges with two machine axis (x and y-linear axis, see the previous paragraph).

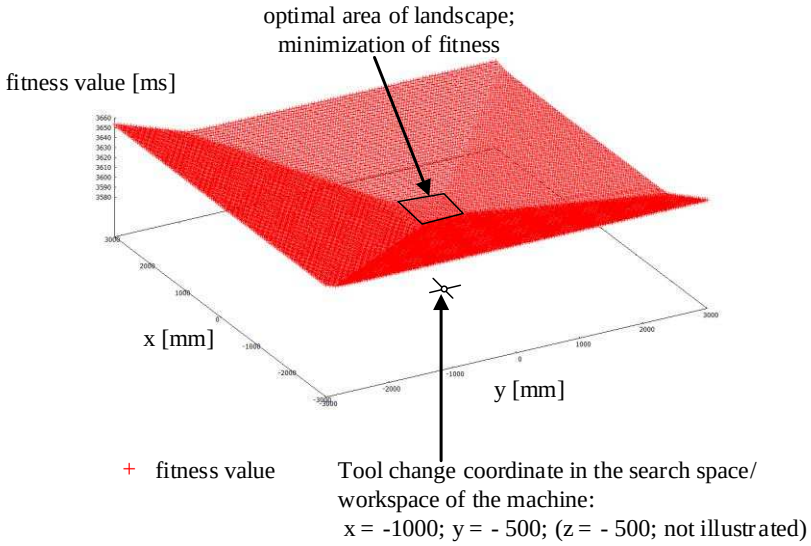

Figure 3: Exemplary overview of the fitness landscape for two linear machine axis.

4 RESULTS FROM THE USE CASE EXPERIMENTS

To check the concept of optimizing the zero point via simulation-based optimization, certain experiments were executed in order to determine the convergence behavior and to compare several configurations of the optimization. The chosen stop criteria for the simulation-based optimization loop was set to 70 generations each particle size. The population size was alternated between 10, 30 and 50 particles and the dimensions were set to 2, 3 and 5 dimensions. The goal was to demonstrate the functional capability of the simulation-based optimization system by minimizing the fitness values (runtime) and synchronously generating coordinates for the zero point, including a determined rapidly converging behavior for the optimization.

4.1 Convergence Behavior

Figure 4 illustrates one case of the convergence behavior of the zero point optimization. The dimension size is 2 and the number of particles is 30. After 14 generations, the fitness value reached the best fitness result which lead to a very rapid convergence behavior (see blue line in Figure 4). The average fitness (see red line in Figure 4) as well as the best fitness within particles in certain generations (see green line in Figure 4) show clearly that the PSO-algorithm finds a local optimum which is likely close to the funnel-shape area shown in Figure 3. The particles try then to find a global optimum in other areas. Because of the funnel-shaped properties of the fitness landscape, the particles converge near the (local)

optimum. The area near the lowest point of the funnel-shape in the fitness landscape allows a higher range of possible coordinates, which leads to the same fitness value.

A similar convergence behavior is shown by the experiments using dimension sizes 3 and 5 (see Appendix A, Figures 6 to 13). As a result, it is notable that the experiments which referred in Figure 4 and Figures 6 through 13 show an useful and rapid convergence behavior for 2 dimensions as well as the other remaining dimension sizes. Only the computational runtime increases with increasing dimension size.

Table 1 gives an overview of the convergence behavior of the remaining dimensions and number of particles, especially when significant convergence behavior is determined for the different dimensions and particle sizes. The given numbers fall within a range of plus to minus 10 % of the recorded generation number, as the convergence is occasionally reached in earlier or later generations, but there are never significant differences.

Table 1: Comparison of the convergence behavior and generation numbers between the experiments and their dimensions size and numbers of particles.

Dimension size	number of particles	Generations when convergence is notable	Generations when average fitness successfully converge
2	10	32	40
	30	14	41
	50	33	41
3	10	45	51
	30	46	48
	50	42	44
5	10	48	46
	30	45	49
	50	48	52

The variance in number of generations required to reach convergence between experiments with different dimension sizes and number of particles are not particularly high, but in general it is recognizable that the higher the number of particles, the longer the runtime required to reach convergence and the higher the required number of generations (see Table 1 and Figures 6 to 13 in Appendix A). The simulation-based optimization loop achieved the desired convergence result which is an useful verification of the PSO-algorithm configuration.

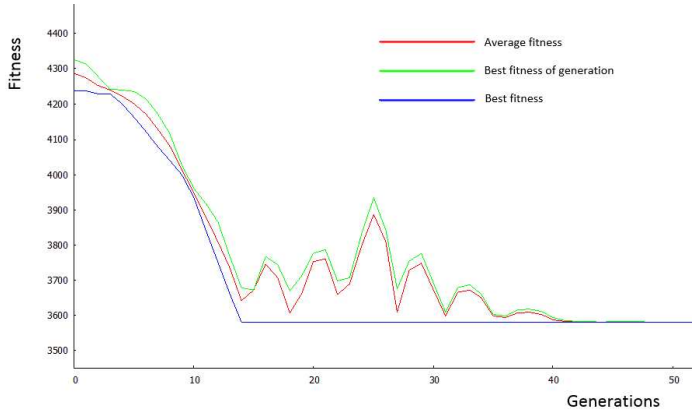

Figure 4: Exemplary convergent behavior for the PSO-algorithm using the NC-Parser fitness component, dimension size: 2, number of particles: 30.

4.2 Optimization Results

Figure 5 show an overview of certain best fitness results in relation to the number of particles. The experiments with a dimension size of 2 show no significant differences between differing numbers of

particles. With increasing complexity (higher dimension size) it is shown that the higher the number of particles the more often a better optimum for the zero point position is determined. One exception appeared during the determined experiments using a dimension size of 5 and 30 particles, which showed the best fitness value in comparison to the experiments using 10 or 50 particles.

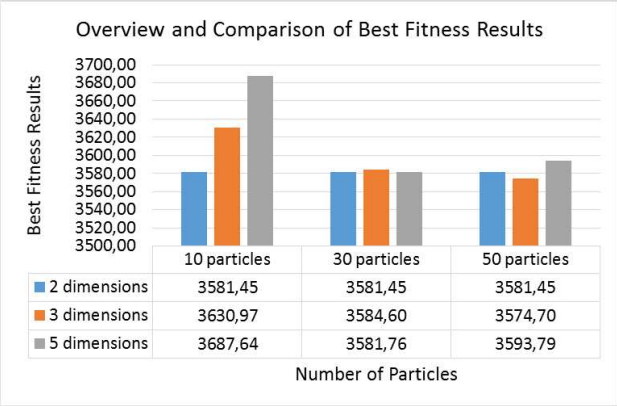

Figure 5: Overview of the comparison of best fitness results with different dimensions and particle size.

Table 2 provides the reached fitness values and the related coordinates for the near-optimal zero point depending on the tool paths given by the standard NC-program. With higher dimension size, the optimization process requires a higher number of generations and beyond 70 generations the coordinates try to jump out of the boundaries of the fitness landscape (see Figure 3). Especially dimension size 5 contains high coordinate values. The coordinates of the rotating axis have to be calculated separately in degrees or radians. In this case an improved configuration for the simulation-based optimization is necessary. The positions of the zero point for 2 and 3 dimensions generally lay near the marked fitness area (see Figure 3).

Table 2: Fitness results including the current position of the zero point after 70 generations.

Dimension size	Number of particles	Average fitness [ms]	Best fitness [ms]	Position [mm for x, y, z] x	y	z
2	10	3581,4467032921	3581,4467032921	836,40	445,83	
	30	3581,4467032921	3581,4467032921	914,47	226,23	
	50	3581,4467032921	3581,4467032921	840,77	261,39	
3	10	3630,9684072900	3630,9684045572	-1727,60	3126,22	2494,11
	30	3584,5954331295	3584,5954317254	1135,15	1000,20	-552,37
	50	3574,7043393985	3574,7043343083	631,98	470,00	-100,00
5	10	3687,6361338880	3687,6361332622	5668,32	-4006,00	3862,48
				A	B	
				155,46	-200,093	
	30	3581,7633272152	3581,7633224462	1044,30	862,038	-92,51
				A	B	
				133,77	-34,31	
	50	3593,7896998604	3593,7896230830	930,06	-464,34	798,06
				A	B	
				-283,57	22,77	

5 CONCLUSIONS AND OUTLOOK

The presented contribution shows the combination of simulation-based optimization using an NC-parser instead of a virtual tooling machine and offers useful results, especially for lower numbers of dimensions

(i.e. a 3-axis machine). The program runtime and the SBO-loop require very short computational time and demonstrate useful convergence behavior without high differences between optimization problems with lower and higher levels of complexity. Further research work concerning the useful implementation of a 5-axis optimization system and the SBO-loop requires a configuration which will take into account collision between tool paths and obstacles within the work space (search space). The position change of the workpiece by SBO which is still represented by the zero point position is also a further developing system element. In general, the NC-parser interface to the simulation-based optimization leads to an efficient preprocessing tool without complicated configurations for optimizing production parameters such as the zero point (see use case in Section 4).

ACKNOWLEDGMENTS

This work is supported by the German Federal Ministry of Education and Research as part of the "*Spitzencluster it's OWL*". This contribution was created by the research group Business Computing, especially Computer Integrated Manufacturing (CIM) under Prof. Dr.-Ing. habil. Dangelmaier of the Heinz Nixdorf Institute in Paderborn, Germany. We thank André Müß for implementing the framework needed for the necessary experiments. The used NC-parser was provided by Dr. rer. nat. Benjamin Jurke (PhD) from DMG Mori Seiki AG, project leader of the current research project. Many thanks to all of the project participants.

A APPENDIX

Figure 6 to 13 demonstrate the convergence behavior for different particle size (abbreviation as part.) and dimension number (abbreviation as dim.).

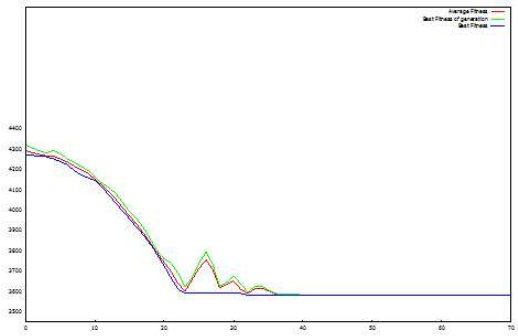

Figure 6: Convergence**:** 2 dim., 10 part.

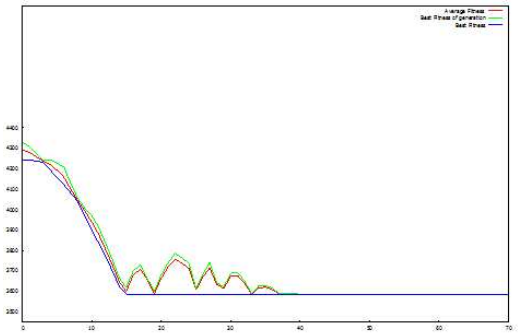

Figure 7: Convergence**:** 2 dim., 50 part.

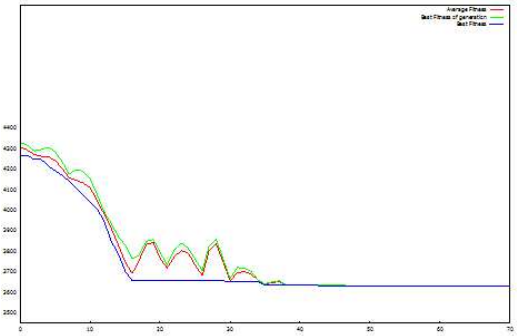

Figure 8: Convergence: 3 dim., 10 part.

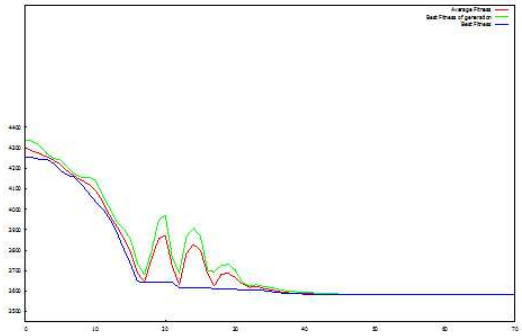

Figure 9: Convergence: 3 dim., 30 part.

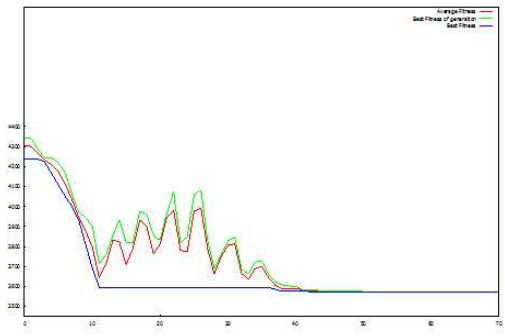

Figure 10: Convergence**:** 3 dim., 50 part.

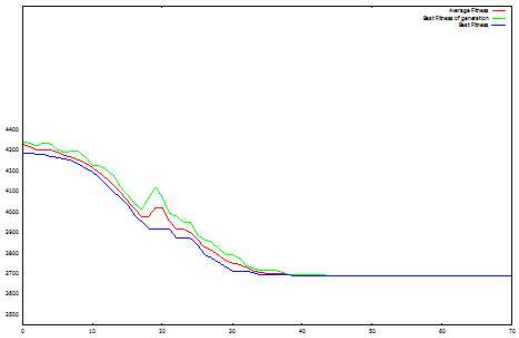

Figure 11: Convergence**:** 5 dim., 10 part.

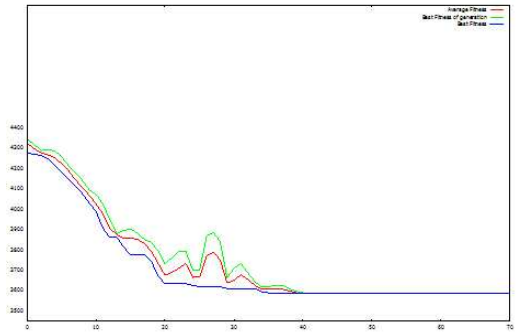

Figure 12: Convergence**:** 5 dim., 30 part.

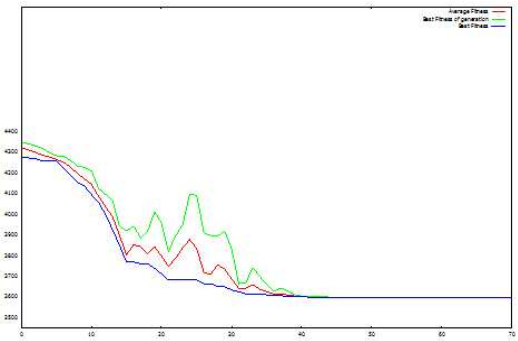

Figure 13: Convergence: 3 dim., 50 part.

REFERENCES

Angeline, P. J. 1998. "Using Selection to Improve Particle Swarm Optimization" In Proceedings of the Evolutionary Computation Proceedings of the IEEE World Congress on Computational Intelligence, edited by N.N. 84-89. Anchrorage, Alaska: Curran Associates, Inc.

Hsieh, W.-T., S.-J. Horng. 2012. "Feature Selection Based on Asynchronous Discrete Particle Swarm Optimal Search Algorithm." In Proceedings of 5th International Symposium on Parallel Architecutres, Algorithms and Programming, edited by Horng et al. 262-268. Taipei, Taiwan: Institute of Electrical and Electronics Engineers; Inc, Curran Associates, Inc.

It's OWL Clustermanagement GmbH. 2015. "Intelligent Technical Systems OstWestfalenLippe". 2015. Accessed April 23, 2015. http://www.its-owl.com/projects/innovation-projects/details/virtual-machine-tools-for-production-planning/

Kennedy, J., and R. Eberhart. 1995. "Particle Swarm Optimization." In Proceedings of the 4th International Conference on Neural Networks. Vol. 4, edited by N.N. 1942-1948. Perth, Western Australia: Institute of Electrical and Electronics Engineers, Inc.

Kleijnen, J. P. C. 2008. "Design of Experiments: Overview." In Proceedings of the 2008 Winter Simulation Conference, edited by S. J. Mason, R. R. Hill, L. Mönch, O. Rose, T. Jefferson, and J. W. Fowler. 479-488. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Koh, B.-I., A.-D. George, R. Haftka, and B. Fregly. 2006. "Parallel Asynchronous Particle Swarm Optimization." In International Journal for numerical methods in engineering. Vol. 67, edited by R. de Borst, C. Farhat et al. 578-595. John Wiley & Sons, Ltd.

Laroque, C., A. Klaas, A. Fischer, and M. Kuntze. 2012. "Fast Converging, Automated Experiment Runs for Material Flow Simulations Using Distributed Computing and Combined Metaheuristics." In Proceedings of the 2012 Winter Simulation Conference, edited by C. Laroque, J. Himmelspach, R.

Pasupathy, O. Rose and A. M. Uhrmacher. 2887-2898. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Laroque, C., B. Urban, and M. Eberling. 2010. "Parameteroptimierung von Materialflusssimulationen durch Partikelschwarmalgorithmen." In Proceedings of the Multikonferenz Wirtschaftsinformatik 2010, edited by M. Schumann, L. M. Kolbe, M. H. Breitner and A. Frerichs. 2265-2275. Göttingen, Germany: Universitätsverlag Göttingen.

Laroque, C., and J.-P. Pater. 2012. "An Automatic Approach for Parameter Optimization of Material Flow Simulation Models based on Particle Swarm Optimization." In Proceedings of the 4th International Conference on Advances in System Simulation, edited by P. Dini and P. Lorenz. 50-57. Lisbon, Portugal: IARA XPS Press.

März, L., W. Krug, O. Rose, and G. Weigert. 2011. Simulation und Optimierung in Produktion und Logistik. 1st ed. Heiderberg, Dordrecht, London, New York: Springer-Verlag.

Rabe, M., S. Spieckermann, and S. Wenzel. 2008. Verifikation und Validation für die Simulation in Produktion und Logistik. Vorgehensmodelle und Techniken. Berlin: Springer-Verlag.

Reisch, R.-E., C. Laroque, and C. Schröder. 2014. "Accelerating the Convergence of Parameter Search Techniques During Simulation-Based Optimization by Dimension Reduction Methods." In Proceedings of the 28th European Simulation and Modelling Conference, edited by A. Carvalho Brito, J. Manuel, R. S. Tavares, and C. Braganca de Oliveira. 13-20. Porto, Portugal.

Reisch, R.-E., J. Weber, C. Laroque, and C. Schröder. 2015. "Asynchronous Optimization Techniques For Distributed Computing Applications." In Proceedings of the 2015 Spring Simulation Multi Conference, 48th Annual Simulation Symposium. Vol. 47. No. 2, edited by A. Tolk, J. J. Padilla and S. Jafar. 49-57. Alexandria, Virginia: Institute of Electrical and Electronics Engineers, Inc.

Shi, Y. and R. Eberhart. 1998. "A Modified Particle Swarm Optimizer." In Proceedings of the Evolutionary Computation Proceedings of the World Congress on Computational Intelligence, edited by N.N. 69-73. Anchrorage, Alaska: Institute of Electrical and Electronics Engineers, Inc.

Suhl, L., and T. Mellouli. 2006. Optimierungssysteme: Modelle, Verfahren, Software, Anwendungen. 1st ed. Berlin, Heidelberg, New York: Springer-Verlag.

Weber, J. S. Boxnick, and W. Dangelmaier. 2014. "Experiments Using Meta-heuristics to Shape Experimental Design for a Simulation-based Optimization System." In Proceedings of the Asia-Pacific World Congress on Computer Science and Engineering, edited N.N. 317-324. Nadi: Curran Associates, Inc.

Wellers, H., N. Kerp, and F. Lieberwirth. 1983. Einführung in die Programmierung von CNC-Werkzeugmaschinen: ein Lehr- und Arbeitsbuch. 1st ed. Essen: Gierardet Buchverlag GmbH.

Wenzel, S., M. Weiß, S. Collisi-Böhmer, H. Pitsch, and O. Rose. 2007. Qualitätskriterien für die Simulation in Produktion und Logistik. Berlin: Springer-Verlag.

AUTHOR BIOGRAPHIES

Jens Weber is a research associate and PhD student at the department of Business Computing, especially CIM at the Heinz Nixdorf Institute, University of Paderborn. He studied industrial engineering with focus on mechanical engineering. His email address is Jens.Weber@hni.uni-paderborn.de.