Evolutionary Optimization of Mold Temperature Control Strategies

Encoding and Solving the Multi-Objective Problem with Standard ES and KEA

J. Mehnen*, Th. Michelitsch*, Th. Beielstein**, K. Schmitt**

*Department of Machining Technology, University of Dortmund, Germany
**Chair of Systems Analysis, University of Dortmund, Germany
*{mehnen, michelitsch}@isf.mb.uni-dortmund.de
**{beielstein, schmitt}@ls11.cs.uni-dortmund.de

Abstract: Modern injection molds are high technology tools that are used for mass production. Typically, the products have to meet high quality demands, because they are often directly visible to the user. In order to manufacture high quality parts, the temperature of the mold has to be kept at an appropriate level and the temperature has to be equally distributed in the form. A net of interconnected deep hole bores leads a water-oil-emulsion into the critical areas to control the temperatures in the tool. Today, the layout of the bores is often still constructed by heuristic means. In this article an objective mathematical measure for the multi-objective quality of a mold temperature control systems is introduced. Two different encoding approaches of cooling systems are given. An approach to solve the complex layout problem using an evolutionary algorithm is described. In oder to show the complexity of this multi-objective problem and to interpret and compare the quality of the existing algorithms, a new software environment called KEA (**K**it for **E**volutionary **A**lgorithms) is used.

Keywords:
Mold temperature control, multi-objective evolutionary algorithms, encoding, fitness function design.

1 Introduction

In modern injection molding high technological tools are used to manufacture mass production parts. The high effort of constructing and manufacturing such molds can increase the costs to half a million Euro per tool. Therefore, the demand for a maximum utilization of the tools is high.

The quality of the workpieces is crucially determined by the heat balance in the tool. Depending on the workpiece material, a specific mold temperature has to be kept. E.g., for *Durethan AKV* plastics, the mold temperature lies between $80°C$ and $120°C$, the melt temperature is $280°C$ to $300°C$ and the demolding temperature is $140°C$. A too low mold temperature may yield an incomplete filling of the tool or results in insufficient surface properties of the workpiece. A mold that is too hot also has disadvantageous effects on the workpiece surface quality and increases the cycle times of the machine [3].

The molds are penetrated with a net of deep-hole bores to balance the tool temperature. These bores are interconnected to circuits which leads fluids (typically a mixture of water and oil) to the critical parts of the form. The circuits realize a well directed heat exchange to reduce or to increase the mold temperature locally and globally.

Today, mold temperature control strategies are mostly designed by experienced toolmakers. The results of the manual layout of the bores are tested by experiments and sometimes also with Finite Elements Method (FEM) analyzes. A well defined objective function to characterize the quality of a mold temperature control strategy does not exist.

Practical problems often show multi-objective characteristics. Here, an evaluation function for several practical demands and restrictions is introduced using physical as well as heuristic knowledge. Experience has shown that – depending on the surface to be cooled and the number of bores used – the fitness function becomes multi modal and the search space is high dimensional. Currently meta-heuristics like evolutionary algorithms are the most auspicious approaches to solve such complex optimization tasks. Here, a new optimization environment called KEA (Kit for Evolutionary Algorithms) as well as a classical Evolution Strategy (ES) are used to develop automatically mold temperature control strategies.

This article begins with a short introduction to the problem. As the encoding of the problem is not unique two different approaches will be discussed in short. The multi-objective fitness functions will be described in the following section. Subsequently, a description of the evolutionary approaches and KEA is given. A discussion of the first results follows. The summary reflects the essence of the research work.

2 Technological Background

Molding tools are basically composed of two parts: an upper and a lower chuck which contain the molds with the workpiece geometry. Both parts contain complementary mold surfaces that define the workpiece structure. Due to the fact that the fluid workpiece material is injected with high pressure, the tool halves are closed with several tons closing force. In injection casting a narrow gap of several hundredth millimeter is left to avoid deformations of the mold inserts. In this case, the pressure is absorbed by small inserts.

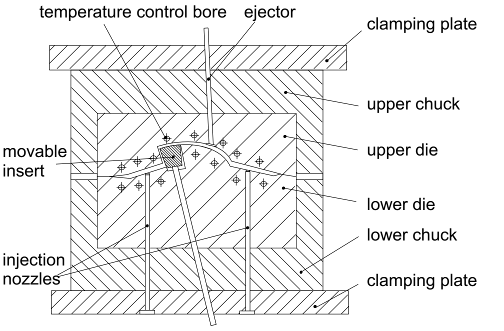

Fig. 1: Sketch of an injection casting tool.

Usually the workpiece has a complex surface structure. If the workpiece shows an undercut, additional movable inserts are necessary. The molds and inserts are typically made of hot working steel (e.g. X 38 CrMoV 5 1) or mold steel (e.g. 40 CrMnNiMo 8.6.4). A completely manufactured moulding tool can have a dimension of several cubic meters (e.g. tools for car bumpers), can weight many tons and can cost up to more than half a million Euro.

Mold temperature control allows to adjust the distribution and progression of the temperature in the tool according to a desired temperature profile. 50 to 70 percent of the cycle time is needed by cooling. The rate and duration of temperature exchange depends on the thickness and consistency of the material between the bore and the mold surface and the throughput of the cooling liquid. A too intense or too local cooling has a disadvantageous influence on the workpiece quality. One possible damage is the so called 'record effect', where the workpiece surface shows concentric grooves. Too slow cooling may yield dull surfaces and increases the cycle times unnecessarily. Especially in die casting, the hot temperatures induces stress and hence increases wear and the possibility of more severe tool damages.

Additionally to the problems which are due to the temperature distribution in the tool also technical restrictions have to be taken into account during the layout of optimal temperature control systems. Basically, the number of bores should be kept as low as possible because every hole implies additional manufacturing costs. The deep drilling process becomes easier the shorter the bores are. Usually, a temperature control system is a combination of many bores which are interconnected with each other to form one large *temperature control circuit*. Plugs are used to reuse already manufactured bores. These plugs introduce a unique direction of the cooling liquid flow through the tool. Additional movable inserts increase, bores for ejectors and injection nozzles increase the complexity of the tool. Neither the bores for the ejectors nor the space used by the inserts must be penetrated by the mold temperature control bores because any leakage must be avoided.

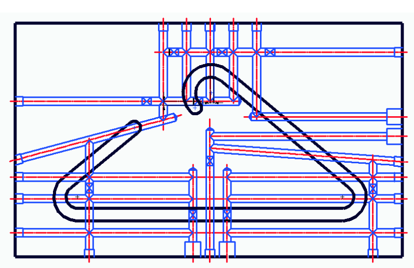

Fig. 2: Realistic cooling circuit for a coat-hanger.

The complex interaction of the circuits, the difficult estimation of the optimal temperature distribution in the workpiece and the problem to design a temperature control network for complex 3D mold geometries necessitate the application of computer aided optimization tools. Fast, objective and comprehensive measures of the mold temperature control design are as necessary as the application of powerful optimization strategies.

3 Two Encoding Approaches

Solving real-world problems automatically implies the formulation of a mathematical description of a problem (the encoding) and its evaluation (via fitness functions). The encoding and fitness functions depend on each other and should be defined as flexible and efficient as possible. Since this is not an easy task, two different encoding approaches for mold temperature control strategies will be compared exemplarily.

Given a numeric optimization problem, the search space is often characterized by quantitative parameters. These can be a set of real or Boolean values. A transfer function may convert the compact raw encoding into a new encoding that is evaluated by the final quality function. In biology as well as in evolutionary algorithms the raw encoding is called the *genotype* (e.g. stored in the DNA in the cell of an individual) and the transformed description is called the *phenotype* (e.g. the corresponding individual). In biology this mapping is typically not unique, what may be advantageous in real-world problems.

The jackstraws approach: The first encoding method for the mold temperature control problem is called the *jackstraws* or *Mikado* approach [7]. This encoding is motivated by the fact that an evolutionary algorithm prefers infinite search spaces without restrictions. The jackstraws encoding describes an individual as a set of several bores that are initially randomly arranged in the volume of the tool. Only bores that do not penetrate a forbidden region – such as the mold surface – are used. It was taken into account that the deep hole drilling machine was only able to adjust bores in integer degrees to the tool surface. Each bore is described as an infinite cylinder that penetrates the mold volume. The random starting points of the bores lie on these parts of the surface that do not belong to the mold surface. Two inclination angles of the bore in 3D-space, the position of the starting point and the diameter of the cylinder define each bore uniquely. The set of all n bores is characterized by a vector of $5n$ real values (the diameter is fix for all bores and is not used for optimization). The vector describes the *genotype* of an individual. In figure 3 an individual for $n = 4000$ bores is shown. A spherical cavity in the top of the tool replaces the mold surface.

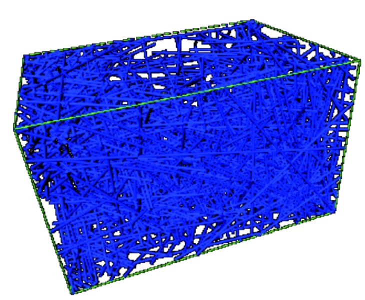

Fig. 3: Jackstraws approach ($n = 4000$ bores).

The phenotype of an individual consists, of course, of *one* mold temperature control circuit. Therefore, the set of bores is scanned recursively for all legal circuits. The probability that two cylinders intersect increases with the number of bores. Here, a limit of 65 percent overlap is used to characterize an intersection of two cylinders. Actually, depending on the number n of cylinders, the number of intersections increases rapidly (maybe exponentially). For very large n the complete block of the tool is filled with bores and therefore the possibility to find a good circuit is high. In this case an infinite number of circuits is possible. For low n and relatively large tool volumes the possibility that cylinders intersect is small. Depending on the number of intersections the recursive scan yields a set of legal circuits. The best cycle found in this set represents the phenotype of the individual. In order to find the best circuit, a multi-objective fitness function is used (definition see below).

The polyline approach: Due to the recursive search, for large values of n the jackstraws approach may need quite long run times. From a practical point of view, the fitness function evaluation of an individual should be as fast as possible. The polyline approach [5, 4] is very intuitive and easy to verify. In this method one circuit or a set of several circuits is represented by a series of interconnected cylinders. These polylines can directly be interpreted as a mold temperature control strategy, i.e. the *phenotype* of an individual. The genotype is a vector of 3D values that define the start point and end point as well as the vertices of the polyline. Each circuit starts and ends at points at the surface of the mold. The parts of the bores that are not used by the circuit are closed by plugs. An additional binary parameter defines the drilling direction of the bore because all holes are implemented as blind holes. Therefore, the genotype of one circuit consisting of n bores can be described by $3(n+1)$ real values plus n Boolean values.

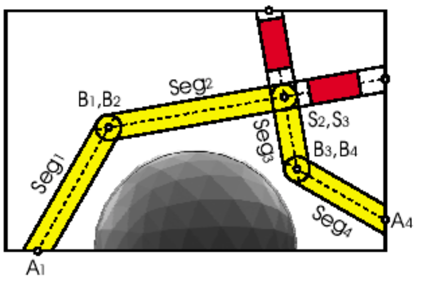

Fig. 4: Polyline approach.

4 Fitness Function Definition

Fast algorithms are crucial for the acceptance in practical applications. Therefore, the fitness calculation has to be as efficient as possible. Especially for complex problems, often large populations are used. This implies a huge number of fitness evaluations and thus increases the run times of the algorithm. On the other hand the fitness function should also reflect the quality of a solution as realistic and comprehensive as possible. In practice, a solution has to fulfill many different and sometimes contradictory demands. Additional restrictions increase the complexity of a problem. Some restrictions are optional ('soft'), i.e. leaving the space of legal solutions is possible but not desirable. Other restrictions are strict ('hard'), i.e. a solution is forbidden, absurd or cannot be evaluated. Soft and hard restrictions can be modeled e.g. by penalty functions. The penalty value depends on the strength of the violation. For soft penalties, a smooth but strongly increasing function should be used. Hard penalties imply individuals that are very bad and should not survive the selection process. A desirable – but perhaps not always possible – alternative to penalty functions is a design of a search space that contains only legal solutions. This implies an adequate encoding or an (efficient) transformation of the search space to an unrestricted space, if possible.

Evolutionary algorithms are created to handle very difficult multi modal optimization problems. Often only the fitness functions are given, their structure is unknown and must not be changed. If an engineer has to design a new quality function, a little bit of sure instinct is needed to avoid complex fitness functions with many extremal points, discontinuities, or narrow and meandering 'valleys'. The fitness function should be able to describe the desirable solutions and should contain a coherent way towards the solutions. Nor the path neither the position of the best solution must be unique. Large plateau areas usually hamper the progress of the search because the direction toward the optimum is not clear. Isolated optima, so called 'needle in the haystack'-solutions, should also be avoided. Of course, neither the encoding nor the fitness functions should be defined to fit a specific search algorithm. Anyway, it is recommended to follow the general remarks for fitness functions given above, to increase the search speed and the possibility to find a reasonable solution.

In the case of the design of mold temperature control circuits, the following general technical demands have to be fulfilled:

- the temperature at each point of the mold surface has to be controlled effectively;
- each temperature control circuit must be separate from each other;
- the bores of one circuit must not intersect other bores of the same circuit to prevent the creation of subcycles;
- bores must not interfere with the tool surface or specific parts of the tool (e.g. nozzles, ejectors, etc.);
- shorter control circuits should be preferred to longer circuits;
- each control circuit should be manufactured with the minimum number of bores;
- the total number of temperature control circuits should be small;
- the angle between intersecting bores should be as large as possible (best values are between 30° to 90°);
- bores should lie in one common plane to reduce machining costs;
- several machine specific deep drilling restrictions and demands should be taken into account (e.g. to avoid machine collisions, drilling angles at the molds outer sides should be near 90°).

In order to calculate the surface temperature of the mold or to evaluate the forbidden intersections of the bores and tool parts, the geometry of the complete tool has to be known. In the approaches described here, each complete half of the mold has been represented by a triangulation of the mold surface. Triangulations have the advantage that the distances between the bores and the object surface can be calculate efficiently. The tool is modeled as a solid cuboid.

Depending on the model, two different fitness functions have been defined. The fitness function for the polyline approach is an enhancement of the simpler approach used for the jackstraws model.

Fitness function for the jackstraws approach: In order to simplify the theoretically very complex calculation of the cooling effect of a complete temperature control strategy, a simple distance criterion has been used in the jackstraws approach. Better approximations of the bores to the mold surface result in reduced distances and lead to an improved cooling effect. The distance is calculated using the n minimum distances of the center points of each triangle of the mold surface to the next neighboring bores. The sum is a linear combination of the distances with a minimum (best value) of zero. In the area beyond the allowed tolerance, a strong penalty function is added to the partial distance values to reduce the chance to get too near or too far away from the mold surface. This sort of heuristic is motivated from the procedure of human engineers. In the jackstraws method, a multi-objective fitness function was implemented as a weighted sum (aggregation approach) of the following criteria: The approximation quality, the length of a circuit and the angles between the bores. All other costs have been estimated to be fix. For demonstration purposes, a mold temperature control strategy consisted here only of one circuit.

Fitness function for the polyline approach: The intention of the enhanced fitness function design was to find a more realistic and fast method to describe the heat exchange effect. In the jackstraw approach all bores laid automatically *within* the tool. In the polyline approach this is not always the case. Therefore, a special penalty function was introduced that increases linearly with the distance of the illegal vertex position from the tool border. This definition induces a search direction toward legal vertex positions that are within the tool. Due to the fact that the algorithm should be able to handle nearly arbitrary tool surfaces, a criterion was established that cares for the safety distance. Therefore, each triangle of the surface is covered by a sphere with radius δ_{secure}. The distance calculation sphere-to-line can be implemented effectively. If a bore touches one of the spheres, a collision is detected. A high resolution of the triangles is advantageous for the collision detection but takes more time. Furthermore, this approach demands for small and isosceles triangles.

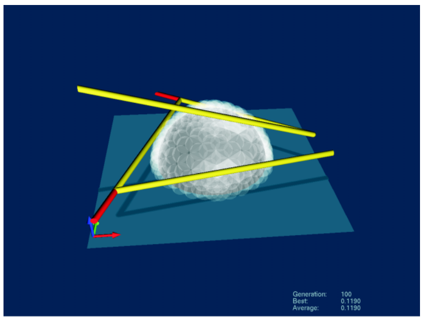

Fig. 5: Safety distance test.

If a bore penetrates the mold surface, a method must be provided that gives a search direction towards better solutions. Therefore, a binary penalty is not helpful. Here, depending on the depth of the penetration of the surface, a continuous penalty function is defined that increases linearly with the orthogonal distance to the surface. If the mold surface is penetrated deeply, some bores may lie completely within the forbidden region, resulting in local minima. Due to the fact that the mold surface of the workpiece is always embedded in the tool, a penalty function is used that repels the bore in z-direction away from the center of gravity. Both case depending penalties are added together to form one global penalty function to model mold surface-to-cooling circuit interaction. In a similar way the interaction between the bores are handled by distance depending linear penalties. Other quality measures like the angle between the bores, the angle between the tool border and the bore (tapping angle), and drilling machine specific properties are modeled with (optional) penalty functions to be able to compare typical real-world designs with automatically calculated constructions.

The definition of the quality function of the cooling effect is directly motivated by the physical process of heat conduction and heat radiation. The heat flux $\dot{q}$, as formulated by Fourier (1822), is proportional to the temperature difference $\nabla\vartheta$, i.e. $\dot{q} = -\lambda \cdot \nabla\vartheta$. The heat radiation follows the same rules as light radiation. Therefore, efficient radiosity algorithms from computer graphics can be adopted. The radiation approach has the advantage that – in contrast to a simple distance criterion – also the length and position of a bore has an influence on the cooling effect. Modeling a bore with start point P_i and end point P_j as a light emitting cylinder (e.g. a neon tube), the effect on a facet of the triangulation of size A and central point M can be estimated by the formula:

$$\tau_{i,j} = A\,|P_i - P_{i+1}| \int_0^1 \frac{1}{(P_i + t\cdot(P_{i+1} - P_i) - M)^2}\,dt$$

The sum of the effects of all bores denotes the cooling effect. The integral can be solved explicitly.

In order to model different heat distributions in the workpiece (e.g. due to different thickness ratios), each triangle has an additional weight that characterizes the local demand for cooling.

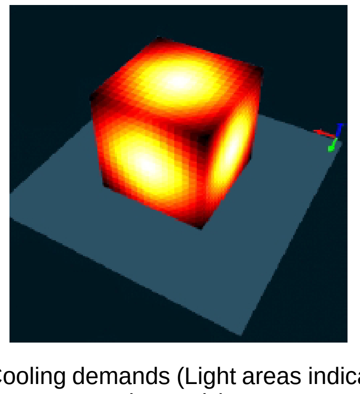

Fig. 6: Cooling demands (Light areas indicate high demands).

Additionally to the local cooling effect modeled by the light emitting method, a homogeneous heat distribution is important. This additional objective is modeled by the statistical standard distribution of the illuminated triangles. A high distribution indicates better mold temperature control strategies. In the polyline approach the multi-objective fitness functions has been modeled via a special multiplicative aggregation method. A separate Pareto approach just contains the single objectives. The fitness function is a weighted product of the standardized restrictions, the manufacturing costs (e.g. length of the bores), and temperature effect, respectively.

Finite Element Method (FEM): Comparing the radiation method used here to model the temperature exchange effect with the typical FEM approach shows quite good similarities. This result is shown in figure 7. The brightness of the colors are proportional to the intensity of the heat exchange.

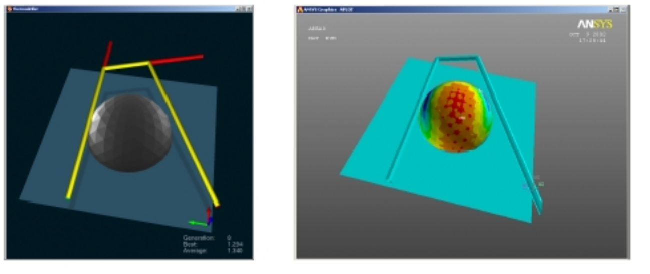

Fig. 7: The light emitting approach (left) versus the FEM model (right).

The advantage of the simplified radiation model against the FEM approach lies in its low evaluation times. This is very important for population based optimization strategies with high numbers of evaluations per run. The time consumed by the FEM for one analysis of the geometry shown in figure 7 is 16 minutes while the radiation method took only $0.786ms$. The polyline algorithm has an interface for integrating ANSYS (tm) FEM analyzes into the optimization cycle. The long evaluation times and the missing stability of the FEM system prevented its continuous application.

5 Multi-Objective Evolutionary Optimization

The automatic design of temperature control systems can be transformed into a multi-objective optimization problem. The design parameters of the system layout (e.g. the number, length, and position of the bores, etc.) have to meet several practical criteria in parallel. At the same time, all mandatory restrictions (e.g. cycle-free temperature control circuits) and optional restrictions (e.g. intersection angles of two bores should not be too steep or too obtuse) have to be fulfilled. Therefore, the basic structure of automatic temperature control system design is multi-objective, strongly restricted, and high dimensional. The solution are generally very difficult to estimate. A general 'solution from the handbook', which can be used to encode a direct optimization algorithm is not available today.

Deterministic optimization and probabilistic strategies can be distinguished. Deterministic algorithms typically try to minimize or maximize a given function by following a fixed scheme (e.g. they follow the gradient direction). They can be very efficient, if the basic characteristics of a problem is known and thus the algorithmic scheme can be adopted to the problem. Evolutionary algorithms belong to the class of probabilistic algorithms. These algorithms have often shown that they are able to find the desired solutions or good approximations of the solution without much a priori knowledge even for very complex problems.

The first experiments using the jackstraws approach and the polyline approach have been executed using a standard evolution strategy. In order to use a well tested ES algorithm for the jackstraws method, the software tool-box *Ease*[6], which provides a standard ES framework for m-to-1-criterion optimization (i.e., $m \geq 1$ criteria mapped to one quality value, e.g. by aggregation), was used. The basic *Ease* framework is written in Tcl/Tk while the operators recombination, mutation, selection and the fitness functions are included as precompiled C/C++ functions.

The basic polyline approach was implemented as a monolithic system using a multi-objective m-to-1-criterion ES with slight changes in the variation operators. The variation of the internal ES for the polyline approach is characterized by a special mutation operator. A fixed binary variation probability was used to vary the direction of the bores. The positions of the vertices of the circuits are adapted with a standard ES step size adaptation mechanism with optional correlation. An additional dynamic lower bound $\sigma_{min}(generation) = \sigma_{min}(0) \cdot 2^{-(generation/\nu)}$ reduced the adaptation speed of the step sizes. A standard ES recombination operator was used in the polyline approach. $0 < \nu < \infty$ is a variable parameter.

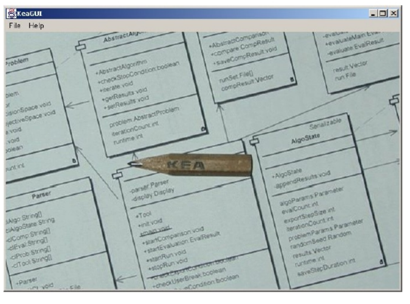

Fig. 8: The Graphical User Interface of KEA.

In order to present a common programming environment for multi-objective optimizations, a new tool called *KEA* (Kit for Evolutionary Algorithms) was also used for the polyline approach. *KEA* provides an interface to evaluate multi-objective fitness functions written in Java or C/C++ using a variety of multi-objective evolutionary algorithms (MOEA), e.g. *NSGA-II* [2] and *SPEA2* [1]. Furthermore, *KEA* provides several state-of-the-art comparison methods for the performance of algorithms (e.g. R-metrics, attainment surfaces and a modified hyper-volume metric) and an interface to a simple visualization tool. In order to use different multi-objective EA for comparison purposes, the monolithic structure of the polyline optimization algorithm was opened and the input and output of the fitness functions were connected through *KEA*. A TCP/IP interface was used to exchange the data between the polyline fitness evaluation and KEA.

In the following, the results using the jackstraws approach were optimized using *Ease*. The polyline approach was tested with a optionally adapted versions of an *ES* and with *KEA*.

6 Experimental Setup

In all ES variants a (μ,λ)-truncating selection was used. Typically, the best ratio of λ/μ (selection pressure) lies between 5 and 7. Here, $\mu = 10$ and $\lambda \in \{10, 30, 100, 300, 1000\}$ was tested for the polyline approach. Intermediate and discrete recombination was tested for the objective and strategy parameters of the ES. The experiments were performed with either 4 and 10 bores per circuit.

In the jackstraws approach, the (μ,λ)-ES applied bisexual intermediate recombination to the objective parameters and discrete recombination to the strategy parameters of an individual.

A cuboid tool with a spherical mold surface was used as a test object. No additional geometric restrictions were tested because the basic structure of the best approximation of the spherical cap was needed. The restrictions bore/tool surface, bore/tool border, and circuit/circuit were applied. The criteria were used as described above. $1,000$ generations were chosen as a general reference limit.

The jackstraws approach runs under LINUX and was programmed in C++. The polyline approach was developed in C++ under Windows2000. KEA is written in Java and is available for Windows2000 and LINUX. A 512MB PC (1.46 MHz) is sufficient for all experiments.

7 Results

The jackstraws approach: An exemplary solution of the jackstraws approach is shown in figure 9. The temperature control circuit coils nearly regularly around the sphere in cyclic steps. 8 to 15 segments are used to form the spiral. The angles between each bore is nearly 90^0 and no bore touches the tool surface. Although the jackstraws approach has the tendency to need a lot of bores, only 100 bores were sufficient to find the solution shown in figure 9.

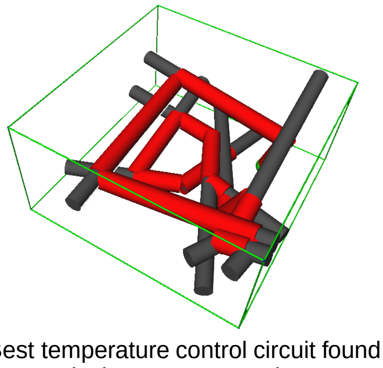

Fig. 9: Best temperature control circuit found by the jackstraws approach.

The algorithm has found this structure during the first few steps. Because small changes in the parameters of an individual lead to very different circuits, here the evolutionary optimization process has a very difficult task to adapt the step sizes properly. A stronger relation between the offspring and the parents in the encoding would be more desirable. From a practical point of view, the solution found is a first and still quite coarse hint for the designer. A special problem of the jackstraws approach are the long evaluation times of the individuals. The solution displayed in figure 9 was found after evaluating 30 individuals using a Pentium PC (450 MHz). The evaluation time took about 3 hours. Furthermore, the restrictions to integer degree values of the adjustment of the bores is unnecessary difficult. The disadvantages of the jackstraw approach were taken into account and have been avoided by the polyline approach.

The polyline approach (aggregation): The complexity of the problem to find an optimal mold temperature control strategy – even for simple mold geometries such as the spherical cap – can only be seen intuitively, if fast evaluation function are available. In this case, the best parameter settings of the ES can be estimated experimentally. Even for only four bores and optimal settings, the evolutionary algorithm yielded various similarly good local optima with very different circuit structures. Best results were found with $\mu = 10$, $\lambda = 100$, a discrete recombination of the objective parameters and an intermediate recombination of the strategy parameters. The experiments revealed that a limitation of the mutation operator using a dynamic limitation with $\sigma_{min}(0) = 0.015$ and $\nu = 400$ was the best way to avoid to get stuck in suboptima. The application of the correlated mutation operator performed slightly better then the uncorrelated version. The best solutions for 4 and 10 bores can be seen in figure 10.

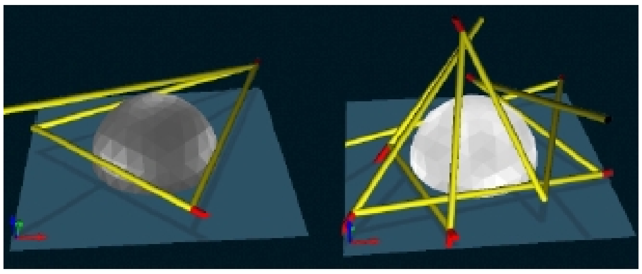

Fig. 10: Best temperature control circuit found by the polyline approach for 4 (left) and 10 (right) bores.

The polyline approach (Pareto): Using a weighted sum or product to evaluate a multi-objective problem has the advantage that well tested uni-criterion optimizers like an ES can be used. Due to several disadvantages of the aggregation approaches such as the necessity of defining the weights and the problem that convex Pareto fronts cannot be approximated correctly, the more general multi-objective evolutionary optimization algorithms NSGA-II – provided by KEA – was used. NSGA-II scans a population for non-dominated individuals and builds iteratively hierachized Pareto fronts of non-dominated individuals. NSGA-II spreads the Pareto-fronts by application of a crowding-distance. In order to use two most dominant criteria for mold temperature control designs, the cooling quality and sum of the lengths of bores were chosen. The variation of the bore directions was set constant in order to make use of the standard mutation operators of the NSGA-II.

Under the condition that also bad cooling effects are allowed, very small bores appeared in the first approach. Although the solutions have a good crowding-distance, these solutions are not practical. Therefore, in a second approach the function of the bore lengths was extended by a penalty. Figure 11 shows the front generated by 20 NSGA-II runs (each run with $1,000$ generations).

After this modification, the multi-objective solutions became practically highly relevant and very interesting for the engineers. Actually, in the discussions with designers the fact become apparent that the solutions of different designers often differ for the same object according to their specific experience and 'personal weighting' of the relevance of each criterion.

The Pareto-front gives an intuitive idea of this phenomenon and can also help to test the designers solution or to motivate alternative and equally good constructions.

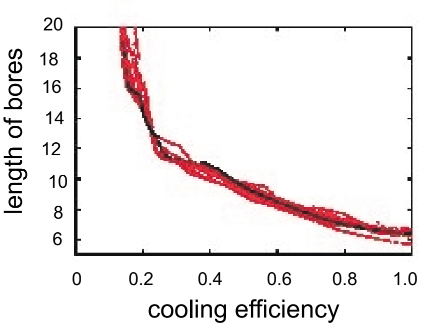

Fig. 11: Pareto front for two criteria.

For population sizes of 100, the NSGA-II was about 20% slower than the ES with optimal parameter settings. The NSGA-II needed nearly no specific parameter adjustment to yield similarly good results. For population size $1,000$ the speed of the NSGA-II was about five times slower than the monolithic ES. This may have its cause in the costly computation of the crowding distance which increases with the square of the number of individuals.

8 Summary and Conclusion

In practice, mold temperature control strategies were solved manually. Here, several realistic quality criteria are introduced that can be used for optimization purposes or to test existing designs. The multi-objective problem is modeled using two different encoding schemes. The high efficiency of the polyline approach prevailed against the jackstraws approach. For both schemes the complex optimization problem is processed by evolutionary algorithms. Both schemes are optimized using an aggregation approach. Additionally, a Pareto solution is calculated for the polyline method using the NSGA-II from the flexible multi-objective programming environment *KEA*.

The experiments show that the design of very fast and realistic fitness functions is essential for the use in evolutionary algorithms. Additionally, an adequate mathematical encoding of the problem itself is important to find fast good solutions. Evolutionary methods have shown to be very good means to solve the complex real-world problem to find also practical convincingly good mold temperature control strategy designs.

Acknowledgments

We would like to thank the DFG (German Research Foundation), grant SFB 531, project C12, for financial support. The optimization environment KEA was developed by a project group of 12 students (PG 419, Multikriterielle Optimierung mittels evolutionärer Algorithmen). Within a years work, the high motivation and engagement of the students did not decrease so now a professional tool for multi-objective optimization is available.

References

[1] Bleuler, S., Brack, M., Thiele, L., and Zitzler, E.: Multi-objective Genetic Programming: Reducing Bloat Using SPEA2. In: Proceedings of the Congress on Evolutionary Computation 2001 (CEC'2001), volume 1, p. 536–543, Piscataway, NJ, 2001. IEEE Service Center.

[2] Deb, K., Agarwal, S., Pratab, A., and Meyarivan, T.: A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II. KanGAL report 200001, Indian Institute of Technology, Kanpur, India, 2000.

[3] Gastrow, H.: Der Spritzgießwerkzeugbau in 130 Beispielen. Hanser Verlag, München, 5 edition, 1998.

[4] Mehnen, J., Michelitsch, T., and Weinert, K.: Evolutionäres Design der Temperierbohrungen für Spritzguss- und Druckgusswerkzeuge. wt Werkstattstechnik online, Springer-VDI-Verlag, 92 (2002) 11/12, p. 588–590. http://www.werkstattstechnik.de/.

[5] Mehnen, J., Michelitsch, T., and Weinert, K.: Evolutionary Optimized Mold Temperature Control Strategies using a Multi-Polyline Approach. In: EuroGP03, p. 374–383, Colchester, UK, 2003. Springer.

[6] Sprave, J.: Ease - Evolutionary Algorithms Scripting Environment. Technical Report CI 55/99, University of Dortmund, SFB 531, 1999.

[7] Weinert, K., Mehnen, J., and Meyer, H.-W.: Evolutionary Optimization of Deep Drilling Strategies for Mold Temperature Control. In: 3rd CIRP International Seminar on Intelligent Computation in Manufacturing Engineering, ICME 2002, p. 291–296, ISBN 88-87030-44-8, Ischia, 2002. Teti, R. (ed.).