INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED'07
28 - 31 AUGUST 2007, CITÉ DES SCIENCES ET DE L'INDUSTRIE, PARIS, FRANCE

IDEAS-GENERATOR FOR CONCEPTUAL DESIGN USING EVOLUTION STRATEGIES

Hehenberger Peter[1], Schauer Stefan[1] and Zeman Klaus[1]
[1]Johannes Kepler University of Linz, Austria

ABSTRACT

Conceptual Design is one of the most important phases during the product development process, as the main characteristics, properties and costs of the solution and consequently also the main portions of success of the new product are pre-determined there. During this design phase the assistance of a computer aided software tool to the design engineer is desirable for the task of finding principal solutions.

In our contribution the application of evolution strategies to typical conceptual design problems is examined using standard algorithms and tools. As a result of this approach, two important advantages arise. First of all, alternative effect chains may be established and secondly, it is possible to evaluate the concepts found, and to select the best of them with respect to a suitable choice of weighting factors for the relevant criteria. Only those solutions are generated and selected, which correspond best to the specified criteria of the multi-requirements problem. The presented approach was realized in a first software prototype using Matlab. It is shown by means of some significant examples, that evolutionary algorithms are useful for the generation and pre-evaluation of principal solutions during the concept phase of product development processes. The possibilities of generating different design concepts via evolution strategies will be demonstrated by a case study.

Keywords: Conceptual Design, Evolutionary Algorithms, Function Structures, Ideas-Generator

1 INTRODUCTION

Today time-to-market and rapid product development are very important aspects of innovation processes. The evolution of the corresponding market requirements and technical possibilities during the last years have deeply modified the designer's way of thinking and operating for all product development stages. In fact, besides the ever important demands on highest quality levels and minimum costs, high speed product development, minimum environmental impact and the accomplishment of highest safety standards have become additional essential requirements to the whole life cycle of the product. In order to push the performance of new products, the beneficial interaction between different fields of mechatronics is stressed more and more, which results in increased complexity of the products themselves as well as of the corresponding development processes. In the conceptual design stage, the specification of requirements, good ideas, significant concept descriptions and proper decisions are of utmost importance for the success of the product.

Using evolution strategies in engineering design is not completely new. It is reported e.g. in [7] and [11], that such strategies were already applied to design optimization of behavior and shape of different products. It is promising to apply the strategies also to conceptual design problems in order to support creativity on the functional level. This paper represents an attempt to apply evolution strategies for finding appropriate function structures with matching effect chains for the realization of the required overall function. As the properties and behavior of a product resulting from a specific solution concept cannot be evaluated finally at the conceptual design stage, the fitness function of the evolutionary algorithm has to be based on the information content available already at this early design phase. Hence, this fitness function uses criteria that are intended to evaluate the potentials of a concept in terms of general information such as structural properties and expert knowledge regarding the list of effects taken into account. These ideas were realized in a software tool under Matlab, some simple examples demonstrate the potential of this approach for generating ideas for new solution concepts.

The paper is organized as follows: Section 2 presents an overview about conceptual design. Section 3 introduces a set of possible function structures and grammar rules for functional decomposition. Section 4 describes the evolutionary algorithm including the fitness function. Finally, a power transmission system is investigated in section 5 in order to demonstrate the potentials of the chosen approach. Conclusions are drawn in section 6.

2 PRODUCT DEVELOPMENT PROCESS

2.1 Overall Design Process

Especially in today's economic environment with its keen, global competition and high dynamics, a superior design concept for a product is crucial, as it pre-determines the main part of success of the product. For any new product, the question is less, how to realize it (realizing a selected concept to a specific product), than to find a superior product concept. In the traditional linear model of design, the process flows from synthesis over analysis to evaluation and decision. Design methodology at the conceptual level includes as a mission the creation of innovative concepts, comprising a description in low detail but with sufficient relevance for evaluation of the potentials of a concept in comparison to others with respect to the requirements. An effective computer-supported method should assist the design team in the performance of these tasks of the conceptual design process.
In order to carry out the concept development as effectively as possible, it is eminent to divide the design task for the overall function into sub-tasks for sub-functions and to create several solutions afterwards to address each function. Once these sub-solutions are conceived, different overall solutions may be synthesized by selecting synergistical combinations of compatible individual solutions for each function. In Figure 1 the integration of conceptual design in the product development process is illustrated (according to [1] and [2]).

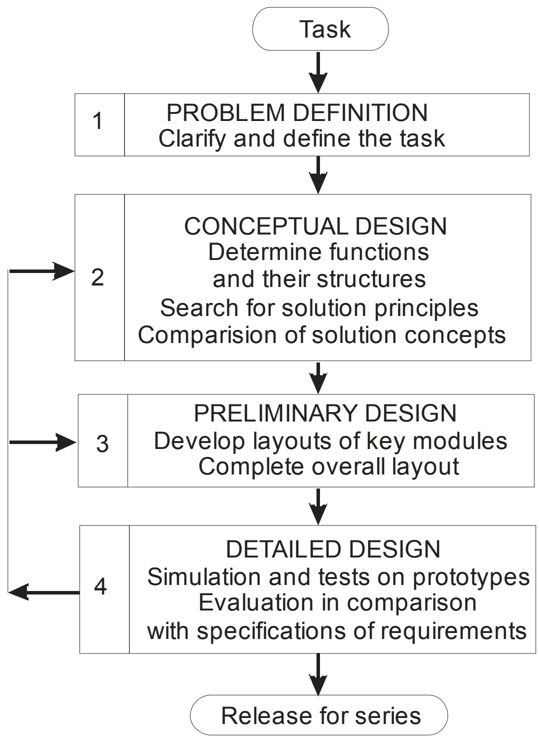

Figure 1. Overall design process according to [1] and [2]

Many definitions for product models in design processes are known from different authors. For example Weber [13] distinguishes between characteristics and properties. The characteristics describe the structure, shape and material consistency of a product. They are determined directly by the designer. The properties describe the product's behavior (e.g. weight, safety, manufacturability, eigenfrequencies, transient response). They cannot be influenced directly by the designer.
Another classification is pointed out by Ehrlenspiel [14]. The characteristics of a product will be divided into three categories:

- Design characteristics ("Beschaffenheitsmerkmale") are characteristics defined directly on the products.
- Function characteristics ("Funktionsmerkmale") describe the purpose of the product.
- Relation characteristics ("Beziehungsmerkmale") describe the interrelationship to other systems or the user.

According to [3], in our paper the term "effect" describes the transformation of physical quantities. An elementary function represents a function, which may be regarded as indivisible (within the set of functions) and uses primarily one distinguished effect.
Amongst the variety of information available to the designer, standards, directives, guidelines and suppliers' data generally provide characteristics, functions, properties and specific aspects of products and the requirements they are to fulfill. On the other hand, the engineer has to follow several design rules (e.g. Design for X) representing general information. Both aspects require a more detailed knowledge of the structure of the product to be developed, its functions and its production, in order to make reliable predictions regarding the properties, behavior and costs of the product. One crucial step in the design process is synthesizing design concepts according to the specified requirements. In the following analysis step, potential properties and the performance of a specific design concept representing the class of resulting detailed solutions are predicted. As a general rule, although the characteristics and properties of a product can be influenced to the greatest extent during its design, the information for design is in general still mainly derived from experience that can only be gained from the phases of the life cycle following the design of the product.

2.2 Conceptual Design

Especially in the early phases of the design process (conceptual design phase, which is shown in more detail in Figure 2) a computer-aided systematic assistance is desirable for finding different solution principles, because in this phase the largest part of the later resulting product costs (60% - 80%, see e.g. [1]) is pre-determined or even fixed. As a consequence, the scope for design is limited to merely small changes in the subsequent design phases.

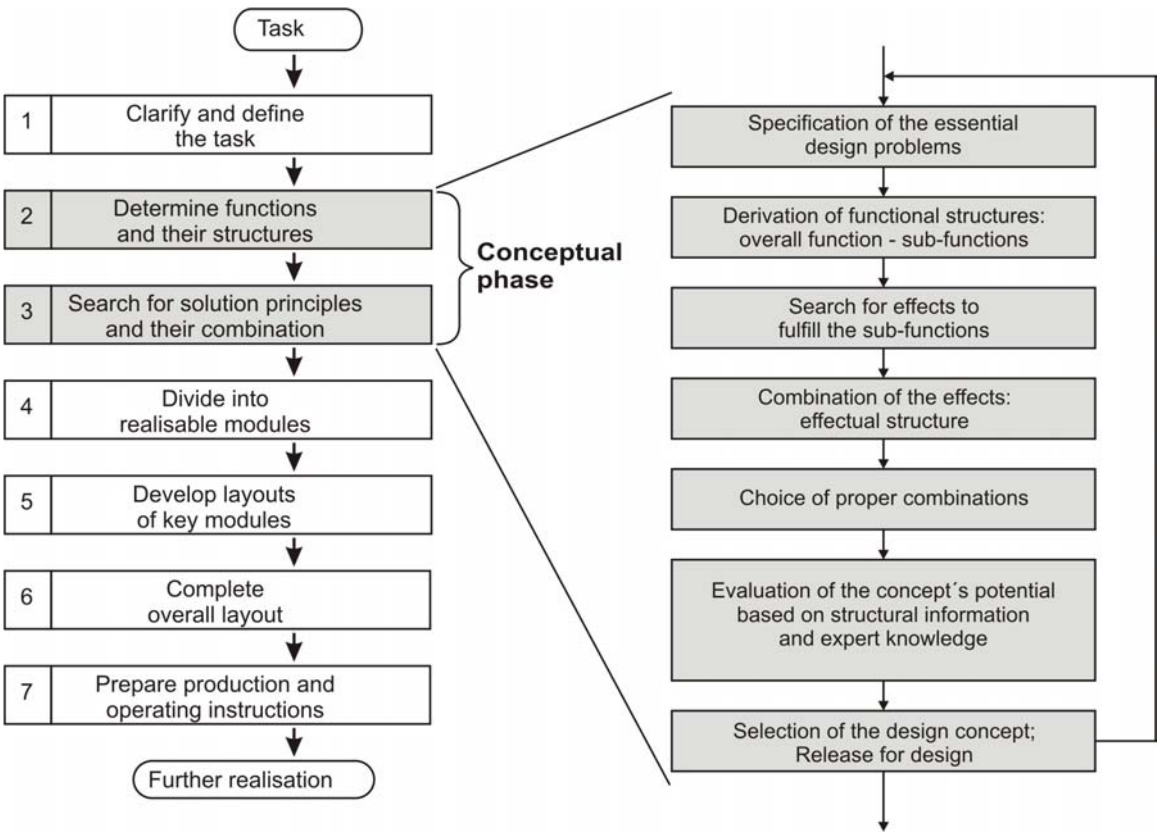

Figure 2. Sequence of the conceptual design phase using evolution strategies

The conceptual phase contains of two main tasks [1]:

- Function synthesis
- Effect synthesis

In the majority of cases, the design process is an iterative process. According to Koller [3], an optimum solution S_{opt} has to fulfill on the one hand the desired purpose P, and, on the other hand, the several criteria (C_i), which may be weighted by different factors (g_i).

$$S_{opt} = f(P, g_i \cdot C_i, \ldots, g_n \cdot C_n) \quad (1)$$

Iterative sequences, as well as the optimization of an objective function, are characteristics typical for evolutionary algorithms. This gives the motivation for their application to conceptual design tasks, the already mentioned phases of which, namely the function and the effect syntheses, are treated in the following. Figure 2 shows a detailed sequence of the conceptual design stage (see [1]).

3 SYNTHESIS AND ANALYSIS OF FUNCTION STRUCTURES

3.1 Function Synthesis

Within this step, a function structure of the product under consideration is created. It has to serve the desired purpose, in other words, it has to fulfill the specified requirements, hence, the function structure has to be derived therefrom. If we assume, that complex technical systems comprise a certain number of elementary functions, we can try to understand this function structure as a cooperation between these elementary functions. A single elementary function is characterized in that it uses primarily one clearly defined (e.g. physical, chemical or biological) effect, hence, it represents a function, which may be regarded as indivisible within the set of functions (see [3]). Figure 3 shows a simple representation of the decomposition of a system into a function structure.

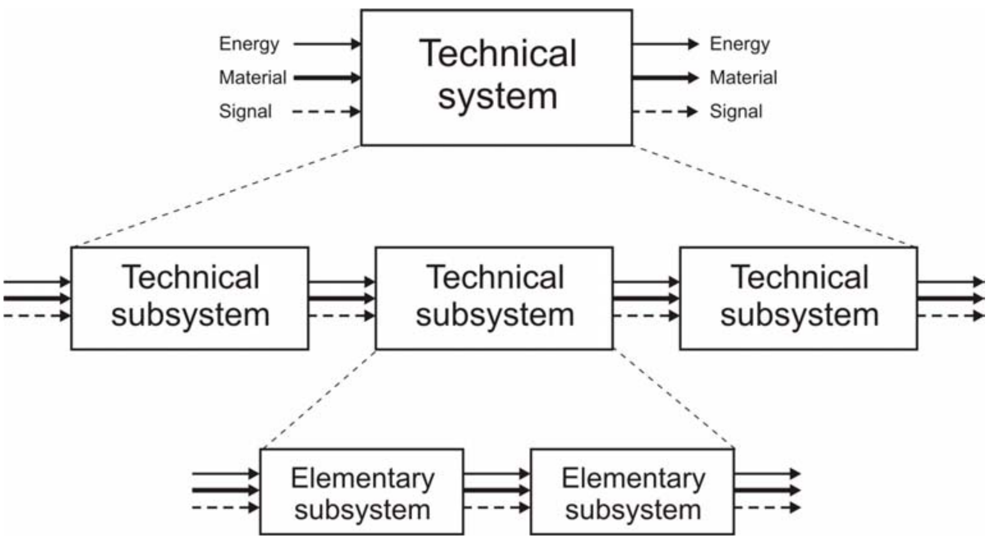

Figure 3. Development of a function structure by decomposition of an overall function into sub-functions according to [1]

Koller [3] defines the following basic operations:

- Convert
- Increase / decrease
- Change of direction
- Conduct / insolate
- Accumulate / divide
- Mix / separate

These operations can be applied to the physical values “energy”, “material” and “signal”, which are also significant for a mechatronic system (see [4] and [5]).

3.2 Effect Synthesis

The effect synthesis combines the necessary effects to realize the specified functions. If different suitable effects are available to realize a certain function, the "most favorable" effect for the concerning task should be chosen. Here the weighting factors for the criteria derived from the list of requirements have to be taken into consideration. On the other hand, it is also possible that no suitable effect is known which would fulfill a specified function directly.

An additional example is given by Koller in [3]: The exemplary task to transform rotational speed into position, cannot be solved directly by the usage of one single effect, because up to now no useful effect with a suitable cause-effect-connection (input value: rotational speed, output value: position) is known. Nevertheless, this task can be solved indirectly by converting rotational speed into centrifugal acceleration and acceleration into force and force into position. This may be realized, for example, by means of the physical effect between angular speed and centrifugal acceleration, by means of Newton's Law into force and finally, by means of Hooke's Law into the deflection of a spring.

3.3 Application of evolutionary algorithms

Many design catalogues are known making principal solutions available in more or less detailed form. For our investigations, the collection of Koller, contained in [3], is used. Another function basis is shown by Stone in [6].

Establishing proper function structures, as well as finding (additional) effects suitable to realize a desired function of the function structure, represent a creative and combinatorial task. The function structures and the suitable effect structures found as solutions of this task have to be evaluated in order to select the most promising solutions. Evaluation, selection and effect-combination to better solutions can be formulated as an optimization problem. As already mentioned in the introduction, the evaluation has to be based on general information, such as structural properties and expert knowledge regarding the list of effects taken into account. This information must be made accessible to the algorithm.

The criteria should be held as general as possible in order not to limit the solution variety unnecessarily. As such evaluations are never strictly objective, they should be accomplished by a sufficiently large number of experienced engineers, and the results should be generated and expressed by statistical methods and significant parameters such as mean value and variance, to objectify the preserved data as much as possible. Another way can be the variation of weighting factors for the evaluation criteria in order to look, which solution is the best for different profiles of criteria representing different customer demands. This approach is proposed in [15] and called "Inverse value benefit analysis" there.

As evaluation criteria for the function structure we chose:

- Number of blocks (should be as low as possible)
- Repetitions of the same block or of block patterns (should be as low as possible)

A simple evaluation of the effect structure addresses the following properties:

- Feasibility:
  Can this effect be realized by standard solutions? Are any complications expected?
- Power supply:
  Are additional power supplies necessary?
- Linearity:
  Is the input-output behavior linear?
- Accuracy:
  Are there accruing very wide tolerances or inaccuracies because of huge dimensions?
- Signal range of values:
  Which range of the physical quantities involved is covered by the effect?

With "Signal range of values" we mean e.g., that the range of the physical quantities (electrical voltage, angular velocity, etc.) between two serially connected effects should fit. If an effect needs as an input, for example, a large position, a piezoelectric crystal would be poorly suited as a preceding effect, because it cannot afford large actuations as output.

Of course numerous other criteria for the evaluation of the principal solutions (e.g. geometrical dimensions, overall efficiency,…) could be mentioned. Some criteria can be evaluated only after an elaboration of the principal solution in more detail, as the data required for evaluation are not yet

available during the conceptual phase. Nevertheless, for demonstration purposes the number of criteria in our example was limited to the abovementioned ones, because they represent at least a reasonable subset of important criteria in the course of the considered conceptual design phase. Afterwards the user must estimate how important these criteria (linearity etc.) are, accordingly he has to define the weighting factors for the fitness function in order to compare the different principal solutions.
Now those principal solutions that attain highest scores of the fitness function should be found by the algorithm. The optimization cycles should yield not only the very best, but also several other very good solutions, because the designer wants to be supplied with several promising concept alternatives.

4 IDEAS-GENERATOR

Concepts and methods for supporting the conceptual design phase have been increasingly reported in proceedings, journals, books during the last years. Many researchers have carefully compiled some of the considerations from the software point of view ([8], [9], [10] and [11]). The authors of this paper will focus especially on mechatronic design which can be defined as an integrated, multidisciplinary design approach using solution principles from different domains, such as mechanical engineering, electrical engineering, electronics and information technologies. Typical mechatronical products are e.g. video-cameras, tooling machines, ABS-systems for cars.

4.1 Preparation of Expert-Knowledge

First of all, some important effects were selected from the abovementioned design catalogue which realize the functions “reduce/increase” and “convert”. Table 1 shows, how many effects for these selected functions are available. In the first version of the software tool, the number of the used physical quantities was restricted to 4 (position s, angular velocity ω, force F, electrical voltage V).

Table 1. Number of available effects

		to			
		1,s	2,ω	3,F	**4,V**
from	**1,s**	2	1	2	**2**
	2,ω	2	1	2	2
	3,F	2	2	3	2
	4,V	2	1	2	1

Table 1 shows e.g. that for the conversion of position s into electrical voltage V, two different effects are available that fulfill this task directly (see also [12]). In this example at least one (direct) effect is available for any conversion, respectively for any dimensional change from one physical dimension into any other one. Unfortunately, in general this is not the case. Then this effect must be substituted by forming an alternative (indirect) effect chain. To be able to simulate this situation with our software tool, the possibility to disable an effect was implemented. In this case the algorithm must form alternative (indirect) effect chains. The effects shown in Table 1 were evaluated according to the following criteria:

Table 2. Evaluation criteria for single effects

Evaluation criterion	Degree of effect-performance
Signal range of physical input value	1 (low) to 3 (high)
Signal range of physical output value	1 (low) to 3 (high)

Power supply	0 (no) or 1 (yes)
Feasibility	1 (very low) to 10 (very high)
Linearity	0 (no) to 1 (yes)
Accuracy	1 (very low) to 10 (very high)

The degrees of effect-performance represent consciously only a rough classification in order to limit the solution variety not too much and to facilitate the evaluation of the single effects. For the input and output ranges, multiple entries are allowed for the case, that an effect covers a very large range of a physical quantity. After assigning the values to the degrees of performance for each effect, they were stored in order to make this knowledge accessible to the algorithm.
For our first tests we reduced the number of the used physical values to four, merely in order to demonstrate the feasibility of the idea. This limitation led to very strong restrictions for the description of technical systems, so in the first instance we had to restrict ourselves to systems with simple structures and a low number of hierarchical levels. Nevertheless, the optimization job of the algorithm can be demonstrated also using this strongly simplified structure.

4.2 Operation of the evolutionary algorithm

In this chapter the different operations of the algorithm are discussed.

4.2.1 Coding of the individuals

Each single individual is fully characterized and coded by two vectors, shown in figure 4. The first vector contains the information about the function structure; the second vector contains the chosen effects.

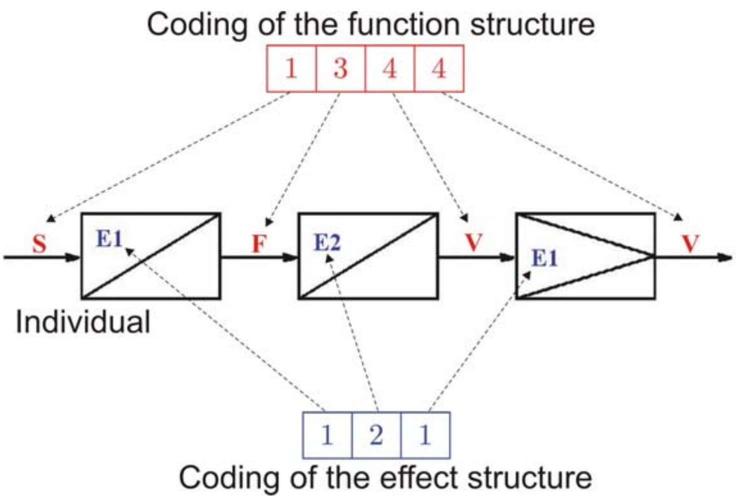

Figure 4. Coding of individuals

The entries in the vector of the function structure represent the physical input and output quantities of the elementary functions involved the function structure. The entries in the effect vector indicate the identification of the effects selected from the available effects (see Table 1) used for the respective elementary function. In this case, for example, the first elementary function shows the conversion of position into force (position s 1, force F 3 in Table 1). The selected effect is identified by the number 1 as first entry in the effect vector, which signifies that the first stored effect of this elementary function is used. This effect describes the use of a lever with adjustable support. If on both sides of the lever static forces are applied maintaining equilibrium of the lever, a change of the position of the support (position s) leads to a change of the relation of the two forces.

4.2.2 Genetic operators

In our algorithm four different genetic operations are implemented. The frequency of the use of the single operations can be adjusted in the user interface. Each of these functions is formed in a manner that after the execution only valid individuals may originate. This is the reason, why e.g. with a function mutation also the effect structure may change.

These operators are:

- Function mutation:
  This genetic operation chooses randomly an entry in the vector of the function structure. Valid choices are all entries, except the first and the last one, because these show the input and output dimensions of the whole individual (function to be realized). Then the chosen entry is mutated, while by means of probability with uniform distribution one of the other three physical dimensions is chosen. To hold the mutated individual valid, the effects which lead from and to this physical dimension are corrected afterwards.
- Function structure length change:
  First it is chosen arbitrarily whether the function structure should be extended or contracted. In the case of an extension, the number of an again randomly chosen certain physical dimension is inserted at an arbitrary place (but not at the beginning nor at the end) of the vector. In the case of a contraction, an entry in the vector is simply deleted. For both variations, the effect structure has to be repaired again to obtain only valid individuals with compatible entries. The number of the elementary functions is changed by 1.
- Effect mutation:
  With the effect mutation, the chosen effect is substituted by a randomly chosen other effect with identical physical input and output, provided such an effect exists. A correction of the existing function structure is not necessary. The function structure does not change by this operation, only a different feasible effect for the fulfillment of an elementary function is selected.
- Recombination:
  For this genetic operation two individuals (in total four vectors) are used. First a physical quantity is chosen in the function structure of the first individual. Afterwards this quantity is searched in the function structure of the second individual. If it is discovered, both individuals become recombined. Then the new individual consists of the first individual up to the chosen physical quantity, and of the second individual from this physical quantity. If the physical quantity chosen from the first individual is not found in the second one, no recombination takes place. The new individual represents a different solution way. It is a combination of both original individuals.

4.3 Evaluation criteria and fitness function

The fitness function contains a sum of different weighted evaluation criteria. The different criteria for the evaluation of a specific individual are shown in the following:

- Block number:
  The number of the used blocks, i.e. the number of the entries in the vector of the effect structure. This quantity denotes the number of different effects that are used to realize the complete function represented by the specific individual.
- Block repetitions:
  Here it is checked whether a physical quantity or a certain block pattern appears more than once in the function structure. The number of the repetitions is saved. A high number of repetitions is undesirable.
- Identical Individuals:
  A high number of identical individuals in a population is in contradiction to a great variety of different solutions. Hence, it is evaluated, how often a specific individual is already included in the population. If for example, an individual is found three times, then the number 0 is assigned to the first appearance, the number 1 to the second and the number 2 to the third appearance, leading to a lower fitness value with increasing number of appearance of identical individuals.
- Signal range:
  First it is checked whether an individual fulfils the demanded input and output signal ranges. Further it is checked whether the output range of every effect also matches the input range of the next effect. Then the assessment is standardized to 10. If the number 10 is assigned to an

individual, all input and output signal ranges agree. If they do not, the individual is worse valued according to the number of violations.

- Feasibility:
  Here the worst feasibility out of all effects involved in the individual is stored. Hence, this value can lie between 0 (difficult to realize) and 10 (easy to realize).
- Demand for external power supply:
  The number of involved effects which need external energy supply is determined.
- Linearity:
  Only if all used effects are linear, this value is set to 1. If one or several effects are non-linear, the value is 0.
- Accuracy:
  Here the worst accuracy of all used effects of the individual is stored. This value can lie between 0 (very inaccurate) and 10 (very accurate).

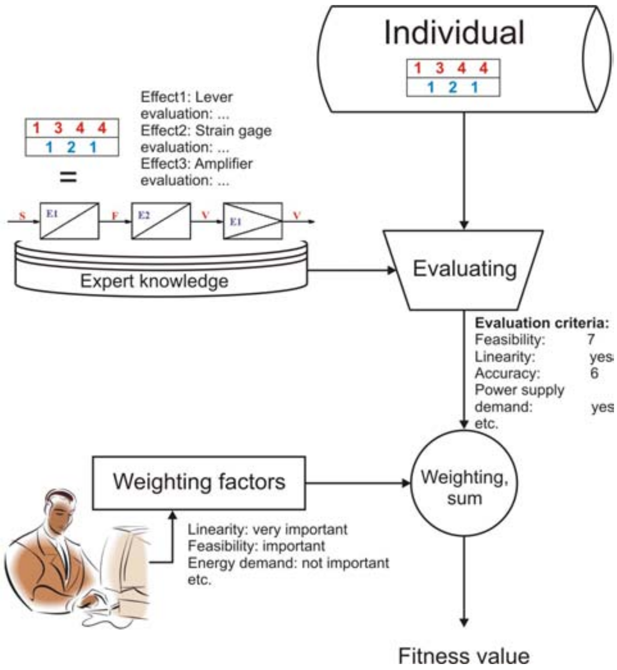

Figure 5. Evaluation process for one individual (solution)

It is to be noted that the evaluation criteria are calculated up to now only from the structure of the current individual (or the current population) and the 'expert's knowledge' stored in Table 1. Now the single criteria become weighted by different, mostly user adjustable factors. Afterwards all weighted criteria are summed up, in order to evaluate the fitness value of the individual under consideration. This process is shown in figure 5.

The fitness function is very important for the algorithm, because it controls the progress of the optimization. If the fitness function is defined wrong, or some factors dominate excessively in proportion to others, the algorithm may force the solutions into a wrong direction. Now the user's task is to find the weighting factors in such a way that the different criteria are well balanced with respect to the fitness function. Another difficulty arises from the fact that the different criteria, in general, are not evaluated in the same physical domains and intervals. Some show a number (e.g.: number of block repetitions, identical individuals ...), others are evaluations from the effect table (e.g.: feasibility, linearity ...), which are assigned a specific value out of an interval ([0,1], [0,1,..10],…).

In the following some aspects of the development of the fitness function will be discussed. The first investigation suggests the formulation of a simple sum over the weighted criteria with the right sign and weighting factors according to the preferences of the user. However, some passes of the algorithm showed, that this was not the optimum configuration. Some criteria have too low impact on the selection of the solutions. Obviously, they were weighted too low. The increase of a single weight (e.g. accuracy to the value 5 = 'extremely important') hardly influenced the selected solutions. To take

remedial action for that, some weighting factors were squared. The finally used fitness function comprises a sum of weighted single criteria with linear and quadratic weighting factors. The adaptation of weighting factors was carried out manually.
The definition of the fitness function could also be attained by giving the user the possibility to adjust not only the weights in a pre-determined function, but also the function itself via the user interface. A further approach would be the possibility to let the fitness function be optimized automatically on the basis of feedback from the user to the program.

5 IMPLEMENTATION AND APPLICATIONS

5.1 Software implementation

According to the definitions in the above chapter, a first software prototype was developed under Matlab. To make the programming effort feasible for the purpose of first demonstrations and tests, some restrictions were made. In the first version of the software tool, the number of the used physical quantities was reduced to 4 (position s, angular velocity ω, force F, electrical voltage V). Furthermore, this program is only able to generate directly lined up (serial) function structures without any branching. The user interface of this first example is shown in figure 6. In addition, two other windows appear. The first one shows the graphical representation of the individual. The second window contains a textual description of every single elementary function including an exemplary application of the chosen effect.

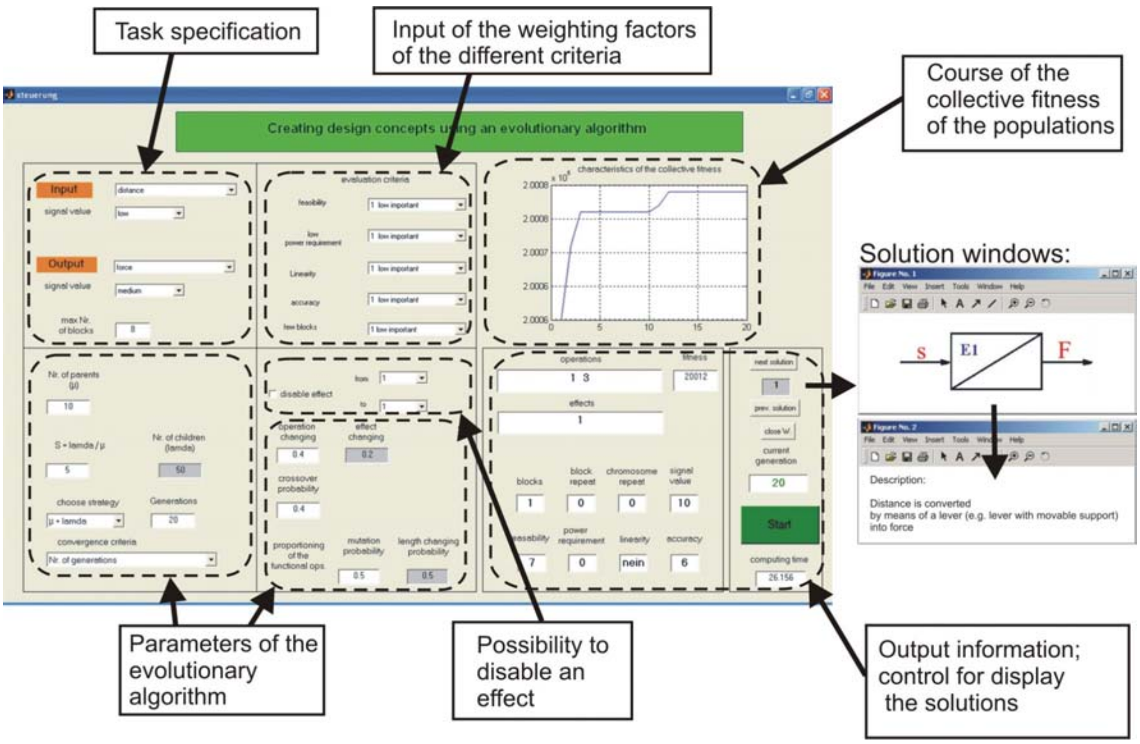

Figure 6. Graphical User Interface

5.2 Case studies

The following examples show results from the algorithm. For every example we distinguish between solutions realizing the demanded function directly and additional alternative solutions developed by the algorithm. In order to show the potential of the algorithm in principle, a simple example was chosen, which will be discussed in the following. The task is to convert a low angular velocity to a higher value. This function is used in all technical systems which need a transformation between two angular velocities. In the table of effects a very common solution is found for this task:

- Solution 1: "Geometrical physical effect"
  With this effect some standard solutions like gears, belt drives, chain drives etc. are indicated.

Additionally to solution 1, the algorithm yields further solutions:

- Solution 2: Generator + amplifier + electrical motor
  This so-called ‘electrical gear’ works in such a way that the input angular velocity is attached to the rotor of a generator, in which the angular velocity is transformed into a voltage. Now this voltage is amplified at first and afterwards supplied to an electrical motor, which further generates an angular velocity. The advantage of this solution lies in the fact that no geometrical constraints such as coaxiality, low distance, fixed gear ratio etc. are involved in the transformation between both angular velocities.

- Solution 3: Centrifugal acceleration + strain gage + amplifier + electrical motor
  The angular velocity is converted into a force by means of centrifugal acceleration. The force is transformed into a voltage by means of Hooke's Law in combination with a strain gage. An amplifier and an electrical motor change the voltage into the desired angular velocity.

- Solution 4: Centrifugal governor + potentiometer + amplifier + electrical motor
  A centrifugal governor is a device with a shaft and two rotating articulated bars with masses at their ends. If the centrifugal governor is rotating at a specific angular velocity, the articulated bars are moved in radial direction due to the centrifugal forces, resulting in a position change of the movable hub (see Fig. 7). This position change can be measured by a potentiometer and be converted into a voltage, from which an amplifier and an electrical motor may generate the desired angular velocity.

The solution principles shown here are only a reasonable choice out of all solutions found by the algorithm. It should be noted that the sketches of the different solutions shown in Fig. 7 represent already a restriction of the solutions delivered by the algorithm, because for some effects other realizations than sketched in Fig. 7 are possible. The "geometrical physical effect" used in solution 1 shows such an example. The sketch may signify a gear box, belt drive, chain drive or something equivalent.

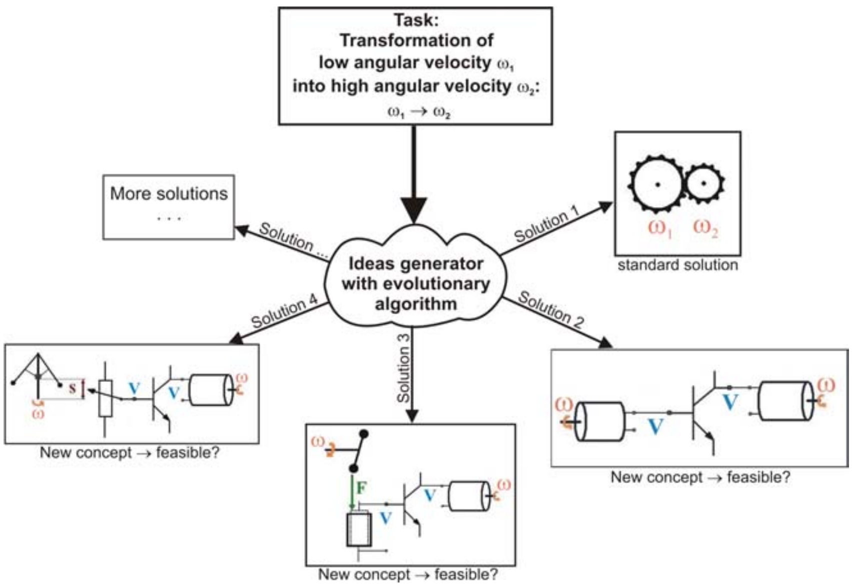

Figure 7. Different solution concepts generated by the genetic algorithm

The chosen approach of finding principal solutions by means of evolutionary algorithms [7], of course, does not guarantee to find the optimal solution. Undoubtedly, it will be possible to (skilled and creative) engineers to find several more and even better solutions. It is not the intention, to substitute the (systematic or intuitive) mental process of the development engineer, but to support him as much as possible. This can be attained, for example, if the algorithm suggests function structures or physical effects, the design engineer did not consider yet. Furthermore, this approach allows to include an increased number of weighted criteria.

6 CONCLUSION AND REMARKS

In this contribution the application of evolution strategies within the range of conceptual design is examined. As a result of this approach, two important advantages arise. First of all, alternative effect chains may be established and secondly, it is possible to evaluate and select the solutions found with respect to a suitable choice of weights for the relevant criteria. Only such solutions are generated and depicted, which correspond best to the desired requirement parameters. The presented concept was implemented in a first software prototype under Matlab. It is shown by means of some significant examples, that evolutionary algorithms are useful for the generation and identification of principal solutions during the concept phase of product development processes. The possibilities of generating different design concepts via evolution strategies were demonstrated by a simple case study.

REFERENCES

[1] Pahl G. and Beitz W. *Engineering Design – A Systematic Approach*, 1999 (Springer Publishing Group, UK).

[2] VDI 2221 Design Handbook 2221, Systematic Approach to the Development and Design of technical Systems and Products, 1993 (VDI Publishing Group, Düsseldorf).

[3] Koller R. and Kastrup N. *Prinziplösungen zur Konstruktion technischer Produkte*, 1994 (Springer Publishing Group, Berlin).

[4] Isermann R. Mechatronic Systems, Fundamentals, 2002 (Springer Publishing Group, Berlin).

[5] VDI 2206 Design Handbook 2206, Design Methodology for Mechatronic systems, 2003 (VDI Publishing Group, Düsseldorf).

[6] Stone R. and Wood K., Development of a Functional Basis for Design. *Journal of Mechanical Design*, 2000, 122(4), 359-370.

[7] Gerdes I., Klawonn F. and Krause R. *Evolutionäre Algorithmen*, 2004 (Vieweg Publishing Group).

[8] Qui S.L., Fok S.C., Chen C.H. and Xu S. Conceptual Design Using Evolution Strategy. *The International Journal of Advanced Manufacturing Technology*, 2002, pp. 683-691.

[9] Yu T.L., Yassine A.A. and Goldberg D.E. *A genetic Algorithm for Developing Modular Product Architectures*. In Proceedings of the 2003 ASME Design Engineering Technical Conferences, Chicago, Illinois, September 2003, DETC2003/DTM-48657.

[10] Güroglu S. and Erden A. *Development of an evaluation measure for Genetic approach to functional level conceptual design*. In Proceedings of the 9th Mechatronics Forum International Conference, MECHATRONICS2004, Ankara, Turkey, September 2004.

[11] Goldberg D.E. *Genetic Algorithms as a Computational Theory of Conceptual Design*. Proceedings of Applications of Artificial Intelligence in Engineering 6, pp. 3-16, 1991.

[12] Hirtz J.M., Stone R.B., McAdams D.A., Szykman S. and Wood K.L. *Evolving A Functional Basis for Engineering Design.* In Proceedings of 2001 ASME Design Engineering Technical Conference, Pittsburgh, PA, September 2001, DETC01/DTM-21688.

[13] Weber C. *CPM/PDD – An extended theoretical Approach to modeling Products and Product Development Processes.* In Proceedings of the 2nd German-Israeli Symposium on Advances in Methods and Systems for Development of Products and Processes, Stuttgart, Germany, 2005.

[14] Ehrlenspiel K. *Integrierte Produktentwicklung*, (2007, Hanser Publishing Group, München).

[15] Naderer R. *Konzeptbewertungmethoden und Weiterentwicklungsprozesse für innovative mechatronische Systeme.* PhD-thesis, Johannes Kepler University of Linz, 2006.

Contact: P. Hehenberger
Johannes Kepler University of Linz
Institute for Computer Aided Methods in Mechanical Engineering
Altenbergerstr. 69
4040, Linz
Austria
Phone: ++43/7322468-9703
Fax: ++43/7322468-9700
e-mail: peter.hehenberger@jku.at