In Need of Methods to Solve Imprecisely Posed Problems

Andreas Geiger, Andreas Kroll and Hanns J. Sommer
Faculty of Mechanical Engineering, University of Kassel, Mönchebergstrasse 7, 34121 Kassel, Germany

Keywords: Artificial General Intelligence, Computational Intelligence, Imprecisely Posed Problems.

Abstract: Problem solving and problem understanding are interwoven. Often at the beginning of a project, engineers' knowledge is insufficient with regard to the posed problems. The correct specification of a task may depend on its solution, which is why the conventional sequence of specification and solution cannot be maintained. Methods are needed to deal with 'imprecisely posed problems', by which the solution process will be adapted to the essential problem structures only. We present procedures to detect these essential structures and to acquire problem solving skills from the solution process itself.

1 EXPOSITION OF THE THESIS

Albert Einstein stated in his Herbert Spencer lecture of 1933 (Kragh, 2013) that "we can discover by means of pure mathematical considerations the concepts and the laws ..., which furnish the key to the understanding of natural phenomena".

Following Einstein's dictum, many scientists believe that also engineering is restricted to the solutions of mathematically precisely posed problems. But is this attitude really the most fruitful? Einstein declared in subsequent lectures: "experience alone can decide on truth", and our experience from successful engineering projects tells us that a thinking that restricts himself exclusively to well posed problems will never be open for the needs of the problems emerging from practice. The openness for new ideas is a capacity which is indispensable for every engineer involved in practical work. Tools are needed to support this capacity:

Thesis 1. *Instead of solving engineering problems by first transforming them into presisely posed problems, it is more effective to solve them with methods suitable for solving imprecisely posed problems.*

To bring this thesis into a precise form, we suggest the following definitions:

Definition 1. *By an imprecisely posed problem we understand a problem whose parameters, boundary conditions or other descriptions are not completely fixed.*

Definition 2. *A method to solve imprecisely posed problems is a solution method which accepts detailed definitions and specifications with the benefit of hindsight.*

The requested methods should not only be able to transform imprecisely known data into a mathematically well-defined fuzzy-knowledge (Köpcke, 2004) but also to prepare and to elaborate solution methods whilst essential requirements for a specification of the solution are missing. Soft constraints are in terms of Definition 1 mathematically precisely specified. It is our intention, to select from an imprecisely specified problem the essential structures by which the general features of solution method can be determined.

To justify Thesis 1, we demonstrate first that important objectives of engineering design strategies are in contradiction with an early specification of the final solution. In the third section, three methods are presented that correspond to the different types of inaccuracies. Section 3.1 discusses a concept-oriented approach, by which the essential structure of a problem will be detected and used for the selection of an adequate solution method. The approach presented in section 3.2 is suitable when no structures are cognisable and, as every determinate approach implies some bias, in section 3.3 a method is discussed to acquire problem solving skills from the solution process itself. Finally we summarise by emphasising the importance and universality of the demand for methods to solve imprecisely posed problems. Our consideration shows that Artificial General Intelligence (AGI) provides a theoretical foundation for the design of strategies to solve imprecisely posed problems.

Geiger A., Kroll A. and J. Sommer H..
In Need of Methods to Solve Imprecisely Posed Problems.
DOI: 10.5220/0004592702940299
In *Proceedings of the 10th International Conference on Informatics in Control, Automation and Robotics* (ICINCO-2013), pages 294-299
ISBN: 978-989-8565-70-9

2 ENGINEERS NEED FLEXIBLE DESIGN METHODS

The following examples demonstrate that engineers in design, development and construction should and very often must avoid an early determination of the final outcome of their task. Some principal reasons for this demand are presented.

2.1 Plant Engineering

In this subsection we present experiences of five years of collaboration in the construction of broadcasting plants. Plants have a quite long lifetime (30 years in the case of broadcasting plants). This means that when new plants are designed and constructed, the experiences of the operators of the old plants as well as new technological advances should be considered (Aztiria et al., 2010). The operators of the older plants are usually not familiar with new technologies and can appreciate them only after some testing time whilst they gain real experience. The design of a plant must be adapted to the operator's feedback, but this feedback will only be disposable after the main components of the plant and its control system have been realised.

2.2 Evolvability

Other industries, like automobile industry, face the problem of plant modernisation every two or three years. The knowledge obtained in the production process must be transferred from the actual plant to the future designs. The concept of evolvability plays an important role in all engineering tasks (van Beek and Tomiyama, 2012). For factories, it is not only important to have good products but just as much necessary that these products and including their production processes offer the possibility to be continuously adapted to new, today unknown requests of future clients. Examples from the history of industry demonstrate that is is often better to have no solution than to have the second best solution. Already more than hundred years ago, Thomas Edington made this sad experience. He lost a lot of money because his solution for an electrification of the United States (using DC technology) was only the second best compared with Westinghouse's solution using AC technology.

2.3 Chemical Engineering

Process design in chemical engineering (Gross, 1999) confronts the problem that exact models are not available. Exact knowledge has always to be complemented with heuristic rules whose adequacy cannot be guarantied or tested in advance. As model evaluation is impossible without fixed parameters and parameter estimation is only possible for well defined model classes, in the early stages, plant design can never be transposed into a mathematically well defined optimisation problem. Flexibility is therefore one of the main exigencies a design process of chemical plants should satisfy. Even in its use, the task to control the unknown dynamics of many components corresponds to the class of imprecisely posed problems. Many chemical processes had to be optimised during their operation time.

3 METHODS TO SOLVE IMPRECISELY POSED PROBLEMS

In this section we present and discuss strategies which we consider suitable for solving imprecisely posed problems. Each of these strategies corresponds to a specific problem of common designs.

3.1 Concept - Orientation

The worst case for system engineers occurs, when complete components of a project must be replaced. For example in software technology and system engineering optimised algorithms are often strongly adapted to a special problem and not flexible enough to accept small changes without significant redesign. Algorithms are highly specialised entities, which are very costly and it is a priori never an obvious task and often even impossible to adapt them. To prove the conformity of an algorithm with a problem, the algorithm must be completely accomplished. But does an alternative exist to a mathematical theory of algorithms?

Computational Intelligence (CI) offers (compared to the huge and intractable set of algorithms) a better manageable set of concepts which have the property of being adaptable by a suitable realisation to special problems. Whereas algorithms yield most efficiently the solutions to well-defined problems, CI-procedures are universally applicable (Kroll, 2013).

Concepts can be characterised by principles describing their main idea. In Table 1 some CI methods with their underlying operational principles are listed.

A further option, to obtain concepts for problem solving consists in generalising and representing the methods of mathematical procedures in a general language. This method has been used by Kroll and Som-

Table 1: CI-methods and their corresponding principles.

Neural Networks	Let many groups of agents try to solve a problem and select the most effective one.
Evolutionary Computing (cross over-strategy)	Combine good parts of several entities to obtain a better new entity.
Computed Swarm	Learn from the best.
Tabu Search	Realise a greedy search but avoid to test twice the same element of the search space.
Diversity Search	Distribute the search over all parts of the search space.
Fuzzy Logic	Information granulation.

mer (Kroll and Sommer, 2013) to deduce principles for solving coordination problems from a list of methods for constraint satisfaction problems that was given by a theorem of Schaefer.

The concept based approach presented in this subsection has its root in the spirit of Computational Intelligence. To solve a problem using a concept based approach, the following steps must be realised (compare Figure 1):

Steps of a Concept-oriented approach:

(I) Choose a list of CI-methods with it's corresponding concepts.

(II) Select from the list those methods whose principles match the characteristics of the problem.
For an imprecisely posed problem, those methods are selected whose specification parameters enable a later fixation of a priori unknown problem specifications.

(III) The experiences obtained from tests with realisations of the selected methods will be used to fix the unknown problem specifications.

(IV) Finally the more precisely stated problem will be solved with the algorithm that has been obtained from the selected methods.

The concept oriented approach has been successfully applied to solve a

Coordination Problem for Plant Control:

Find control sequences for the agents controlling a plant such that the mutual influences will be minimised which the actions of one agent produce in the parts which are in the responsibility of others (Dürrbaum et al., 2012).

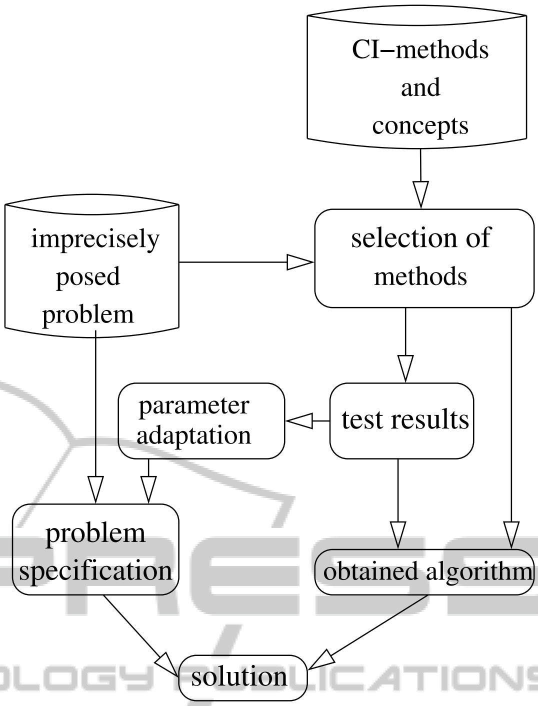

Figure 1: A Concept-oriented Approach.

For this problem it is a priori unknown:

- Which rules must be applied by the agents to guarantee absence of mutual interference?
- Are these rules acceptable for the plant operators?
- Do additional constraints exist that may be satisfied? (E.g. smoothness of actions, time restrictions, etc.)

The selection of problem adapted methods had provided for Tabu Search and Diversity Search. A flexible procedure was realised with these methods which made it possible to find the unknown data with the benefit of hindsight after a period of extensive tests with the system.

3.2 Belief-based Modeling

In the case when no physical model exists, the most often applied approach to obtain a mathematical frame is system identification (Ljung, 1987). But even this approach may be impossible in cases where no model class can be selected or where no error model is available. An example of such a problematic task was presented by the requirement to develop an information system to forecast drill fracture in chipping processes (Dürrbaum et al., 2008).

Due to the variety in design and material, it was absolutely unrealistic to develop a physical model for the chipping processes. Even more, the temporal distribution of these events was not available, as it was impossible to precisely specify a definition of fracture events. To solve the problem only the following data could be utilised:

- During the chipping process measurements of axial force $F(t)$ and moment of torque $M(t)$ were available.
- The result of each particular chipping process was categorised respective to two classes: 'good' or 'bad'.

As no other information was available, the fracture forecasting system had to be based on this data exclusively.

3.2.1 Idea of the Solution Approach and its Mathematical Background

Each observation made during a test beginning at time t_{start} and ending at time t_{end} can be described by a fuzzy-sentence of the form:

Example 1. *In the first time interval $I_{begin} \subset [t_{start}, t_{end}]$, $F(t)$ is strongly increasing, in the middle time interval $I_{mid} \subset [t_{start}, t_{end}]$, $F(t)$ and $M(t)$ are oscillating, in the final time interval I_{fin}, $F(t)$ and $M(t)$ are zero and the result is bad.*

Let the letter S refers to a particular sentence describing measurements $m = \{(F(t), M(t), result) \mid t \in [t_{start}, t_{end}]\}$ obtained from one chipping process. A formal representation of the propositions into a sentence S (e.g. 'In the first time interval $I_{begin} \subset [t_{start}, t_{end}]$ $F(t)$ is strongly increasing.') using fuzzy-membership-functions, transforms this sentence itself into a fuzzy-set with a membership-functions $\mu_S(m)$ for measurements m.

Each measurement $m_i = \{(F_i(t), M_i(t), result_i) \mid t \in [t_{start}, t_{end}]\}$ for $i = 1, \ldots, N$,that had been obtained during a test phase, provides a confirmation $\mu_S(m_i)$ for a sentence S. The belief-degree $\mathrm{bel}(S)$ of a sentence S is given by the aggregation of all these confirmations $\mu_S(m_1), \mu_S(m_2), \ldots, \mu_S(m_N)$ with an aggregation operator Agg (Jager, 1994). The following Lemma 1 demonstrates that the conditions claimed for an aggregation determine the operator Agg up to a rescaling.

Lemma 1. *(Benferhat, 2010), (Sommer and Schreiber, 2012)*
Let Agg *be an operator that assigns to values $d_1, d_2, \ldots, d_N \in [0,1]$ an aggregated value $\mathrm{Agg}(d_1, \ldots, d_N) \in [0,1]$ such that the following conditions hold for all values $d_i \in [0,1]$:*

$$\mathrm{Agg}(d_1, \ldots, d_i, d_j, \ldots, d_N) = \mathrm{Agg}(d_1, \ldots, d_j, d_i, \ldots, d_N),$$

$$\mathrm{Agg}(d_1, \ldots, d_i, \ldots, d_N) \leq \mathrm{Agg}(d_1, \ldots, \bar{d}_i, \ldots, d_N), \quad \textit{for } d_i \leq \bar{d}_i,$$

$$\mathrm{Agg}(d_1, \ldots, d_k, \mathrm{Agg}(d_{k+1}, \ldots, d_N)) = \mathrm{Agg}(d_1, \ldots, d_N), \tag{1}$$

There exists a value $e \in (0,1)$ such that:

$$\begin{aligned} &\mathrm{Agg}(d_1, \ldots, d_N) = \mathrm{Agg}(d_1, \ldots, d_N, e), \\ &\mathrm{Agg}(d_1, \ldots, d_N) < \mathrm{Agg}(d_1, \ldots, d_N, d_{\max}) \\ &\qquad \textit{for } d_{\max} = \max\{d_1, \ldots, d_N\} < e \end{aligned} \tag{2}$$

The operator Agg *can be represented with the fuzzy-operators $\wedge$ and $\vee$,*
(with $a \wedge b = a \cdot b$; $a \vee b = 1 - (1-a)(1-b)$),
if the values d_i have been re-scaled by functions $\Phi_\wedge : [0,e] \to [0,1]$ and $\Phi_\wedge : [e,1] \to [0,1]$
with the equations:

$$\begin{aligned} \mathrm{Agg}(d_1, d_2) &= \Phi_\wedge^{-1}(\Phi_\wedge(d_1) \wedge \Phi_\wedge(d_2)) \\ &\quad \textit{for } d_1, d_2 \in [0,e] \\ \mathrm{Agg}(d_1, d_2) &= \Phi_\vee^{-1}(\Phi_\vee(d_1) \vee \Phi_\vee(d_2)) \\ &\quad \textit{for } d_1, d_2 \in [e,1] \end{aligned} \tag{3}$$

For $d_1 \in [0,e]$ and $d_2 \in [e,1]$ various possibilities exist:

- *Remove any value $d_2 \in [e,1]$ from the aggregation,*
- *Remove any value $d_1 \in [0,e]$ from the aggregation,*
- *Use the following equations to calculate the aggregation:*

$$\begin{aligned} \mathrm{Agg}(d_1, d_2) &= \Phi_\vee^{-1}\left(1 - \frac{1 - \Phi_\vee(d_1)}{\Phi_\wedge(d_2)}\right) \\ &\quad \textit{for } (1 - \Phi_\vee(d_1)) < \Phi_\wedge(d_2) \\ \mathrm{Agg}(d_1, d_2) &= \Phi_\wedge^{-1}\left(1 - \frac{1 - \Phi_\wedge(d_2)}{\Phi_\vee(d_1)}\right) \\ &\quad \textit{for } (1 - \Phi_\vee(d_1)) \geq \Phi_\wedge(d_2) \end{aligned} \tag{4}$$

For a balanced decision, the last of the possibilities to aggregate values from different zones $d_1 \in [0,e]$ and $d_2 \in [e,1]$ will be used.

Lemma 1 has been presented in different formulations by various authors. A prove can be found in (Sommer, 1995).

3.2.2 Realisation of the Solution Approach

The method presented in section 3.2.1 had been used to construct a **fracture-forecast system** by the following steps (Dürrbaum et al., 2008):

- Construct randomly fuzzy-sentences S describing the measurements that had been obtained during a chipping process.
- Compare the constructed sentences S with real measurements $\{m_1,\dots,m_N\}$, using their fuzzy-membershipfunction μ_S to calculate the values $\mu_S(m_i)$.
- Highly believable sentences, (satisfying bel$(S) \approx 1$) which assert bad results are selected. (Nearness between $bel(S)$ and 1 is the selection criterion. Lemma 1 shows that a different choice of the aggregation operator can be compensated by a modification of this criterion.)
- The selected sentences are included into an alarm system, forecasting breaks in the work process.

During the use of the fracture-forecast system in the work process, the following rule will be used:

- If a measurement taken from an individual work process has a high membership degree into some of the selected sentences, an alarm will be given and the drill must be changed.

The method used in this project may be transferred to many other problems. Belief-Based Modeling is a search technique for a mathematical description which can be applied for problems whose mathematical precision is questionable.

3.3 Co-evolution-Method

As mentioned in section 2, over and above the problem description, solving imprecisely posed problems calls for information obtained in the whole solution process. This implies a different view. It is now necessary to consider the whole context of a problem. The theory which tries to extend Artificial Intelligence and to understand this view is called Artificial General Intelligence (AGI) (Goertzel and Pennachin, 2007),(Sun, 2009). AGI offers a theoretical background for Computational Intelligence by examining the relations between phenomena that had been detected in cognition and their reasons in knowledge processing. To stress this relation, we use the definition:

Definition 3. *A solution system for an imprecisely posed problem that has been designed with CI-methods will be called an AGI-system.*

The phenomenon of cognitive synergy implies the empirically well confirmed experience that to obtain a powerful AGI-system multiple subsystems focused on learning regarding different sorts of information must interact in such a way as to actively aid each other in overcoming combinatorial explosion. Due to an effect called "trickiness" by Goertzel, Ihle and Wigmore (Goertzel et al., 2012), cognitive synergy is responsible for the difficulties to evaluate and optimise AGI-systems:

> "'Trickiness' means the effect that, in each case of a practical test [to evaluate an AGI-system] it seems likely that there is some way to "game" the test via designing a system specifically oriented towards passing that test, and which doesn't constitute progress towards a general power [of the tested AGI-system]."

Because of cognitive synergy, methods for searching solutions to imprecisely posed problems should be composed of various procedures, but due to the trickiness-effect, the influence of the particular procedures on the overall method is a priori unknown and only detectable from the final solution. As a consequence, the overall method should be selected from a **co-evolutionary-competitive system** in which various optimisation processes influence each other.

Co-evolution-Method:

When solving a problem, at the very beginning we dispose of a huge set of possible search processes (searching for a solution of the problem) which are differently composed of the basic procedures and which may use different assumptions with regard to the missing specifications. Some of these collections are selected into a competitive system and adapted with the following steps:

Evaluation: Evaluate the solutions offered by the individual processes and learn from them their recommendations of a final problem specification.

Information Exchange: The currently best solutions with their corresponding problem specification will be placed at the disposal of the others.

Elimination: Eliminate ineffective processes.

Configuration of new Processes: Insert new processes using the information obtained from the competition.

Co-evolution is an adequate method to solve imprecisely posed problems because it uses besides the problem information the information that will be provided by the search process itself. In this way, information can be exploited which is not available for a conventional search, using an evolutionary algorithm(Geiger et al., 2010).

4 CONCLUSIONS

The motivation of Thesis 1 reveals the relations between Computational Intelligence (CI) and Artificial General Intelligence (AGI). The understanding provided by these scientific disciplines is indispensable for every approach to solve imprecisely posed problems.

In the same way as problem solving is the objective of Artificial Intelligence (AI), imprecisely posed problem solving can be understood as the objective of AGI. Two well known experts of AGI, Richard Loosemore and Ben Goertzel wrote in a recent paper (Loosemore and Goertzel, 2012):

"There are two possible types of intelligence speedup: one due to faster operation of an intelligent system (clock speed increase) and one due to an improvement in the type of mechanismus that implement the thought process ('depth of thought' increase)."

But as speedup may be limited by quantum mechanical limitations for the hardware and by the intractability of many problems, the second type of intelligence amplification seems to be more promising. Really intelligent systems must be "self-understanding". To solve a problem means first, to understand this problem, but secondly to understand also the search process and context for its solution. Many industrial systems like plants are not only technical systems but also social systems, and as such systems they enable emergent events. The complexity residing in those systems requires a parallel design of it's parts. For that reason, plant engineering always confronts imprecisely posed problems. The "depth of thought" in engineering reasoning is therefor strongly related to the ability for solving imprecisely posed problems. The methods of CI and AGI are suitable to create strategies for that task. The presented strategies correspond to the strongest confinements of common designs.

REFERENCES

Aztiria, A., Izaguirre, A., and Augusto, J. C. (2010). Learning patterns in ambient intelligence environments: a survey. *Artificial Intelligence Review*, 34:35–51.

Benferhat, R. (2010). Interventions and belief change in possibilistic graphical models. *Artificial Intelligence*, 174(2):177–189.

Dürrbaum, A., Kroll, A., and Sommer, H. (2012). Ci-methoden zur koordination und entkopplung komplexer systeme. *at-Automatisierungstechnik*, 60(10):600–610.

Dürrbaum, A., Scherm, W., and Sommer, H. (2008). Systemidentifikation mittels glaubenskalkül. *Automatisierungstechnik*, 7:350–356.

Geiger, A., Sommer, H., and Kroll, A. (2010). Koevolutionaerer algorithmus zur analyse struktureller steuerbarkeit - eine methode der 'komplexen netzwerke' zur systemanalyse. In F. Hoffmann, E. H., editor, *20. Workshop Computational Intelligence*, volume 33, pages 235– 249, Karlsruhe. Institut fuer Angewandte Informatik/Automatisierungstechnik, Institut fuer Technology.

Goertzel, B., Ikle, M., and Wigmore, J. (2012). *Foundattions 'AGI' Mind Architecture*. Atlantis Press, Amsterdam.

Goertzel, B. and Pennachin, C. (2007). *Artificial General Intelligence*. Springer, New York.

Gross, B. (1999). Gesamtoptimierung verfahrenstechnischer systeme mit evolutionären algorithmen. In *Fortschritt-Berichte VDI*, volume 3, page 131. VDI-Verlag, Düsseldorf.

Jager, R. R. (1994). Aggregation operators and fuzzy systems modeling. *Fuzzy Sets and Systems*, 67(2):129–145.

Köpcke, H. (2004). Optimierung von aufstellungsentwürfen von chemieanlagen durch constraint programming. In *14 Workshop Fuzzy-Systeme und Computational Intelligence*, page 139, Dortmund.

Kragh, H. (2013). Testability and epistemic shifts in modern cosmologiy. *Studies in History and Philosopy of Modern Physics*. to appear.

Kroll, A. (2013). *Computational Intelligence, Eine Einführung in Probleme, Methoden und technische Anwendungen*. Oldenbourg Verlag, München.

Kroll, A. and Sommer, H. J. (2013). Coordination of distributed control systems. *to appear in International Journal of Computing Anticipatory Systems*.

Ljung, L. (1987). *System Identification, Theory for the User*. PTR Prentice-Hall, Inc., London.

Loosemore, R. and Goertzel, B. (2012). Why an intelligence explosion is probable. http://www.fhi.ox.ac.uk/archived-events/winter-conference.

Sommer, H. (1995). The logic in knowledge processing formalisms. In H.J., Z., editor, *EUFIT'95*, volume 31, pages 274–278, Aachen.

Sommer, H. and Schreiber, L. (2012). Is logic in the mind or in the world? why a philosophical question can affect the understanding of intelligence. *Journal of Artificial General Intelligence*, 3(1):25–47.

Sun, R. (2009). Theoretical status of computational cognitive modeling. *Cognitive Systems research*, 10:124–140.

van Beek, T. J. and Tomiyama, T. (2012). Structured workflow approach to support evolvability. *Advanced Engineering Informatics*, 26:487–501.