Bogataj, L. (ed.): Inventory Modelling - Lecture Notes of the International Postgraduate Summer School (Portoroz, Slovenia, August 1995). ISIR, Budapest, pp. 87-92

A Knowledge-based Approach to the Optimization of Technical Systems and Its Application to Inventory Systems

Sven Hader

Department of Computer Science
Technical University of Chemnitz-Zwickau
09009 Chemnitz, P.O. Box 964
Germany
email: sha@informatik.tu-chemnitz.de

Abstract

This paper describes knowledge-based optimization as an alternative to manual optimization traditionally carried out by engineers in the design of technical systems. By make use of this approach, human experts shall not be superseded but supported and relieved of the monotonous part of their work. The optimization system DIM_EXPERTE is presented that uses a combination of knowledge-based and numerical optimization techniques. The usage of the system is explained using an example from the domain of inventory systems.

1 Introduction

Interest in the analysis and optimization of technical systems (e.g. manufacturing systems, computer systems) has been growing within the industry for years. This is due to the rising complexity and capital investment of systems being used today. Usually, engineers occupied with design and optimization of systems use an iterative ("manual") *redesign* process[1)], that is, they follow a sequence of trial-and-error steps of the form

... design → design analysis → design modification → new design ...

using their domain knowledge, experience, and intuition. Since the analysis codes (first of all simulation programs) take a long time to run, the parameter space is large, and the time period for optimization is limited, only a small portion of the parameter space can be explored. So there is a need for tools that support engineers in optimization by making better use of information provided as well as by automating the monotonous part of the search. Recently, there have been promising attempts to meet these demands using Artificial Intelligence (AI) techniques (cf. the IDESIGN system [Arora 1988] and the EnGENous system [Tong 1989], [Powell et al. 1990], and [Powell et al. 1991]).

In this paper, a *knowledge-based* approach for the automatic optimization of technical systems is presented. Based on a description of the general principles of knowledge-based optimization (Section 2), the optimization system DIM_EXPERTE is introduced (Section 3).

---

1) " *It was a surprise to me that many technology businesses within a company like GE* [General Electric, S.H.] *would rely for the most part on expert designers iterating analysis codes by hand.* " ([Tong 1989], p. 421)

Following it, the practical application of this system is illustrated using an example from the domain of inventory systems (Section 4). The paper concludes by presenting a summary and directions for future work (Section 5).

2 Knowledge-based Optimization

Knowledge-based optimization is concerned with solving optimization problems using explicitly represented domain knowledge[1)] (usually in the form of IF-THEN rules). For this, a so-called *inference engine* tries to reproduce the behaviour of human experts incorporated in optimization by searching a knowledge base for applicable knowledge and carrying out appropriate changes to system designs (scenarios).

In general, engineers use a two-stage iterative approach in optimizing systems. In the first stage they analyse the current system design (e.g. by simulation) and try to discover existing bottlenecks. In the second stage they change values of certain input parameters to get rid of the detected bottlenecks and improve system's performance. After that, they return to the first stage and analyse the new design. If no bottlenecks can be found and the design suffices the predefined objectives then the optimization is finished.

To simulate engineer's behaviour, two types of domain knowledge have to be incorporated in the knowledge base:

(1) **Analysis Knowledge** (current design → bottlenecks/deficiencies)
This type of knowledge is used to examine the current system design (using results obtained from analysis codes) to identify existing bottlenecks and deficiencies.

Example: Considering an (s,Q) inventory system with a lot of stockouts, we may draw the conclusion that the average inventory level is too low.

(2) **Transformation Knowledge** (bottlenecks/deficiencies → new design)
This type of knowledge is used to change the current system design so that the identified bottlenecks and deficiencies will probably disappear. This knowledge specifies the parameters to change and the direction and magnitude of suitable changes.

Example: To increase the average inventory level of the system mentioned above, we may increase the reorder point s or the reorder quantity Q using appropriate step sizes to get a "better" system.

Although the qualitative formulation of the necessary knowledge is rather simple (see above), problems may occur in representing it in the quantitative way needed by the inference engine (what does "a lot of" or "an appropriate step size" mean ?).

---

1) The knowledge used to perform this task, in general, has been extracted from the human experts in the domain and encoded in a formal representation.

3 The Optimization System DIM_EXPERTE

DIM_EXPERTE[1)] (cf. [Hader 1994b] and [Hader 1994c]) is a general-purpose optimization system that uses a combination of knowledge-based and numerical techniques. Optimization problems containing both discrete and continuous parameters and constraints can be solved. Concrete scenarios are evaluated using analytical or simulative methods in the form of external programs[2)] (→ value of the objective function, performance measures). The system provides techniques for handling statistical data like simulation results (e.g. computation of confidence intervals). Figure 1 shows the conceptual layout of the system.

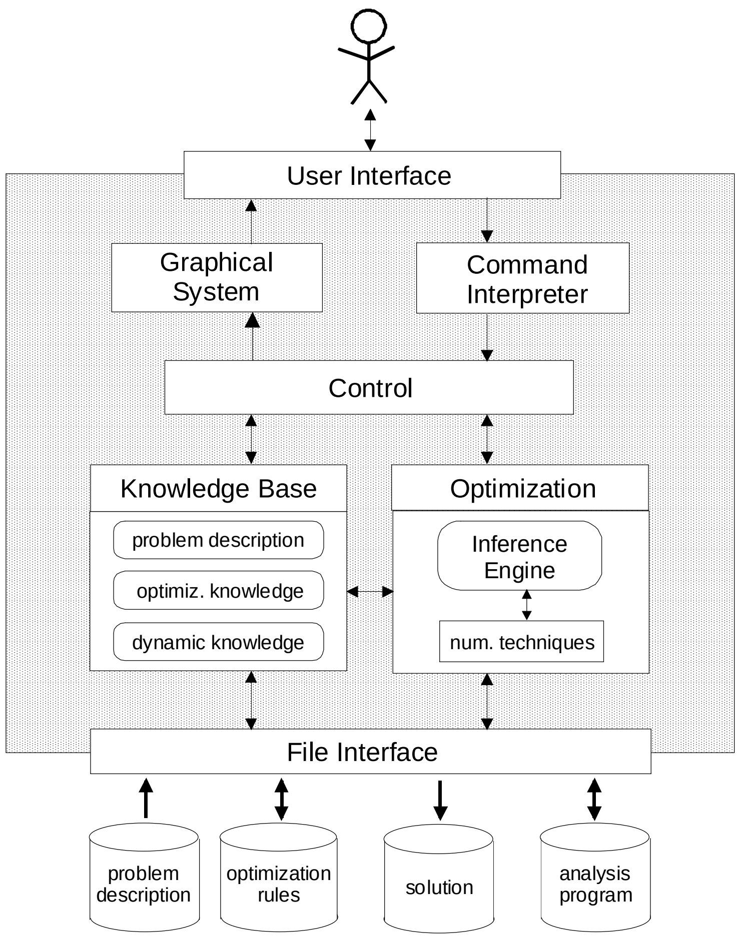

Figure 1. Conceptual layout of DIM_EXPERTE

The most important components of the system are the knowledge base and the optimization module. The knowledge base contains, among others, a problem description, the corresponding domain knowledge (both loaded from text files) and information about current and past system designs. The domain knowledge is represented as *rules* (heuristics) in the usual form

IF <Condition> **THEN** <Conclusion> .

1) DIM_EXPERTE was designed and implemented by the author and is still under development

2) connected via an ASCII file interface

The Condition part of a rule describes the situation in which this "chunk" of knowledge is applicable. The Conclusion part describes changes to knowledge base elements (e.g. system designs, internal states) which have to be carried out if the rule becomes active. To ease the representation of appropriate parameter changes in the transformation knowledge (cf. Section 2), qualitative directions of change (e.g. *increase*, *change*) as well as qualitative magnitudes of change (e.g. *much*, *less*) can be stated. The syntax of the rules contains principles and elements from both the Prolog and the C programming language.

The optimization module contains tools for various optimization techniques like rule-based techniques and techniques for one-dimensional, multidimensional, and random search. These techniques are combined under the control of the inference engine (a forward chaining rule interpreter with various rule-priorization strategies). Numerical techniques are incorporated if a system design to be analysed contains qualitative elements, e.g.

qualitative system design $\mathbf{x}_i$ = (20.5, *increase*(4), -7) .

A *qualitative* system design can be considered as the description of a set of quantitative (concrete) designs. In this case, a numerical search procedure is started to determine the "best" quantitative design (or just a "good" one because of time constraints) out of this set. This combination of rule-based and numerical optimization increases the efficiency of DIM_EXPERTE, in particular for systems with only incomplete respectively uncertain knowledge given. Besides that, the user is relieved of formulating quantitative transformation knowledge[1)].

DIM_EXPERTE runs on personal computers and is written in SWI-Prolog[2)] and C. It consists of approximately 10.000 lines of Prolog code and 11.000 lines of C code at the moment. Analysis codes used can be realized using any programming language on condition that an ASCII file interface is provided (for data exchange between DIM_EXPERTE and the analysis codes).

4 An Example: Inventory Systems

In this section, we shall demonstrate the use of DIM_EXPERTE in optimizing a problem[3)] from the domain of inventory systems. We consider an one-product manufacturing system (MS) with input storage (IS) for raw parts (cf. [Hader et al. 1994a]). The IS is controlled by an (s,Q) policy and can order parts (in batches) from an unlimited central depot. Deliveries arrive after a positive lead time. Finished products leaving the MS are replaced by parts from the IS after a positive insertion time. The demand for the product manufactured is considered unlimited. The problem parameters are the inventory policy (*s*,*Q*) of the

1) Nevertheless, if appropriate quantitative knowledge is known, it should be used to increase the efficiency of the optimization.

2) a Prolog implementation by the University of Amsterdam

3) This example shall serve as a simple test case for DIM_EXPERTE, practical relevance is not considered.

IS and the number of pallets *P* of the MS. The expected net profit of the MS per time unit computes as

$$G(s,Q,P) = g * D(s,Q,P) - h * L(s,Q,P) - K * B(s,Q,P)$$

with	D(.)	-	expected throughput of the MS
	L(.)	-	expected inventory stock of the IS
	B(.)	-	expected number of orders at the central depot
	g	-	reward per part produced in the MS
	h	-	holding cost per part and time unit of the IS
	K	-	ordering cost per order received from the central depot

and serves as objective function to be maximized. The determination of the expected net profit as well as other performance measures for concrete system designs is done by simulation[1)].

The knowledge base contains 9 analysis rules (for deficiencies like *material_shortage* and *pallet_shortage*) and 10 transformation rules. If no deficiencies can be detected then the actual system design is considered to be "optimal" and the optimization process is finished. Alternatively, an additional numerical optimization technique can be carried out to verify and possibly improve the design found.

Optimization was carried out for an example system with $g = 100$, $h = 3$, $K = 2500$ and a constraint on the expected sojourn time of parts in the MS (< 2.0 time units). After the examination of approximately 400 scenarios an "optimal" value of the objective function of 523.15[2)] was found ($s = 46$, $Q = 152$, $P = 18$). It can be established that the "surface" of the objective function around the best scenario found is rather flat[3)] with regard to the inventory policy but rather steep with regard to the number of pallets.

5 Summary and Future Work

In this paper, the usage of the knowledge-based approach for optimization was motivated. The advantage of this approach consists, first of all, in a temporal relief of the expert in the phase of design optimization. It should be noted, however, that because of incomplete knowledge bases[4)] only "good" solutions may be found, so that a verification of the results reached seems necessary. The optimization system DIM_EXPERTE was presented that uses the knowledge-based approach in combination with numerical optimization techniques to optimize systems modelled by appropriate analysis codes (e.g. simulation programs).

---

1) for this task an special simulator called INPROSS is used

2) Because this value was determined through stochastic simulation, it is to be considered as an <u>estimation</u> of the mean value. The 95% confidence interval for this mean value is [521.0 .. 525.3].

3) Therefore, difficulties may occur in comparing values of the objective function especially for "neighbouring" designs.

4) " *Even for a design of minimal complexity, rule completeness is not possible because the engineer does not understand the design space completely.* " ([Powell et al. 1991], p. 314)

As a further development DIM_EXPERTE should be incorporated into a comprehensive package for knowledge-based modelling, simulation, and optimization. This package should support the user in the development of simulation programs as well as in the design of the necessary knowledge bases and the system optimization regarding user-defined objectives. To realize these additional features, especially newer concepts of Artificial Intelligence (e.g. blackboard architecture, deep reasoning) and software engineering (e.g. object-oriented programming) shall be used. For efficiency and availability reasons also a conversion of the whole system into C++ is planned.

References

[Arora 1989] J.S. Arora: *Introduction to Optimum Design*. McGraw-Hill, 1989.

[Hader et al. 1994a] S. Hader, P. Köchel: *A Model for Manufacturing Systems with Input Storage*. Proceedings of the 8th International Working Seminar on Production Economics, Innsbruck, Austria, February 21-25, 1994.

[Hader 1994b] S. Hader: *Ein regelbasiertes System zur automatischen Planung und Durchführung von Simulationsläufen*. 7. Workshop des ASIM-Arbeitskreises "Simulation und Künstliche Intelligenz", Braunschweig, 13.-14. April 1994.

[Hader 1994c] S. Hader: *Regelbasierte Optimierung von Produktionssystemen.* In: G. Kampe, M. Zeitz (eds.): Simulationstechnik - 9. Symposium in Stuttgart, Oktober 1994, Vieweg-Verlag, pp. 603-606.

[Powell et al. 1990] D.J. Powell, M.M. Skolnick, S.S. Tong: *EnGENous: A Unified Approach to Design Optimization*. In: J.S. Gero (ed.): Applications of Artificial Intelligence in Engineering V, Vol. 1: Design. Computational Mechanics Publications, 1990, pp. 137-157.

[Powell et al. 1991] D.J. Powell, M.M. Skolnick, S.S. Tong: *Interdigitation: A Hybrid Technique for Engineering Design Optimization Employing Genetic Algorithms, Expert Systems, and Numerical Optimization.* In: L. Davis (ed.): Handbook of Genetic Algorithms. Van Nostrand Reinhold, 1991, pp. 312-331.

[Tong 1989] S.S. Tong: *Coupling Symbolic Manipulation and Numerical Simulation for Complex Engineering Designs*. Mathematics and Computers in Simulation, 31(1989), pp. 419-430.