Chapter 76
Intelligent Computation for Manufacturing

Ashraf Afify
King Saud University, Saudi Arabia & Zagazig University, Egypt

ABSTRACT

Intelligent computation refers to intelligence artificially realised through computation. This chapter reviews six intelligent computation techniques. They are: knowledge-based systems, fuzzy logic, inductive learning, neural networks, genetic algorithms, and swarm intelligent techniques. All of these tools have found many practical applications. Examples of applications in manufacturing are given in the chapter.

KNOWLEDGE-BASED SYSTEMS

Knowledge-based systems, or expert systems, are computer programs embodying knowledge about a narrow domain for solving problems related to that domain. An expert system usually comprises two main elements, a knowledge base and an inference mechanism. The knowledge base contains domain knowledge which may be expressed as any combination of "IF-THEN" rules, factual statements (or assertions), frames, objects, procedures and cases.

The inference mechanism is that part of an expert system which manipulates the stored knowledge to produce solutions to problems. Knowledge manipulation methods include the use of inheritance and constraints (in a frame-based or object-oriented expert system), the retrieval and adaptation of case examples (in a case-based expert system) and the application of inference rules such as modus ponens (If *A* Then *B*; *A* Therefore *B*) and modus tollens (If *A* Then *B*; NOT *B* Therefore NOT *A*) according to "forward chaining" or "backward chaining" control procedures and "depth-first" or "breadth-first" search strategies (in a rule-based expert system). With forward chaining or data-driven inferencing, the system tries to match available facts with the IF portion of the IF-THEN rules in the knowledge base.

When matching rules are found, one of them is "fired," ie. Its THEN part is made true, generating new facts and data which in turn causes other rules to "fire." Reasoning stops when no more new rules can fire. In backward chaining or goal-driven inferencing, a goal to be proved is specified. If the goal cannot be immediately satisfied by existing facts in the knowledge base, the system will examine the IF-THEN rules for

DOI: 10.4018/978-1-4666-7456-1.ch076

rules with the goal in their THEN portion. Next, the system will determine whether there are facts that can cause any of those rules to fire. If such facts are not available they are set up as subgoals. The process continues recursively until either all the required facts are found and the goal is proved or any one of the subgoals cannot be satisfied, in which case the original goal is disproved. Both control procedures are illustrated in Figure 1. Figure 1(a) shows how, given the assertion that a lathe is a machine tool and a set of rules concerning machine tools, a forward-chaining system will generate additional assertions such as "a lathe is power driven" and "a lathe has a tool holder." Figure 1(b) details the backward-chaining sequence producing the answer to the query "does a lathe require a power source?."

In the forward chaining example of Figure 1(a), both rules *R2* and *R3* simultaneously qualify for firing when inferencing starts as both their IF parts match the presented fact *F1*. Conflict resolution has to be performed by the expert system to decide which rule should fire. The conflict resolution method adopted in this example is "first come, first served": *R2* fires as it is the first qualifying rule encountered. Other conflict resolution methods include "priority." specificity" and "recency."

The search strategies can also be illustrated using the forward chaining example of Figure 1(a). Suppose that, in addition to *F1*, the knowledge base also initially contains the assertion "a CNC turning centre is a machine tool." Depth-first search involves firing rules *R2* and *R3* with *X* instantiated to "lathe" (as shown in Figure 1(a)) before firing them again with *X* instantiated to "CNC turning centre." Breadth-first search will activate rule R2 with X instantiated to "lathe" and again with *X* instantiated to "CNC turning centre," followed by rule *R3* and the same sequence of instantiations. Breadth-first search finds the shortest line of inferencing between a start position and a solution if it exists. When guided by heuristics to select the correct search path, depth-first search might produce a solution more quickly, although the search might not terminate if the search space is infinite (Jackson, 1999).

For more information on the technology of expert systems, see (Giarratano and Riley, 1998; Metaxiotis et al., 2002; Nurminen et al., 2003).

Most expert systems are nowadays developed using programs known as "shells." These are essentially ready-made expert systems complete with inferencing and knowledge storage facilities but without the domain knowledge. Some sophisticated expert systems are constructed with the help of "development environments." The latter are more flexible than shells in that they also provide means for users to implement their own inferencing and knowledge representation methods. More details on expert systems shells and development environments can be found in (Price, 1990).

Among the six tools considered in this chapter, expert systems are probably the most mature, with many commercial shells and development tools available to facilitate their construction. Consequently, once the domain knowledge to be incorporated in an expert system has been extracted, the process of building the system is relatively simple. The ease with which expert systems can be developed has led to a large number of applications of the tool. In manufacturing, applications can be found for a variety of tasks including scheduling, process design, facility layout, production planning and control, material selection, condition monitoring, fault diagnosis, machine and process control, process planning, and quality control. Some recent examples of specific tasks undertaken by expert systems are:

- Designing of products and their assembly processes (Zha et al., 1998),
- Handling the production rescheduling problem (Li et al, 2000),
- Scheduling the production of small and medium sized manufacturing companies (Metaxiotis et al., 2002),

Related Content