



Fuzzy agent-based expert system for steel making process

M.H. Fazel Zarandi *, P. Ahmadpour

Department of Industrial Engineering, Amirkabir University of Technology, P.O. Box 15875-4413, Tehran, Iran

ARTICLE INFO

Keywords:

Steel production
Agent-based
Fuzzy expert systems
Adaptive neuro-fuzzy inference system
Negotiation

ABSTRACT

In this paper, a fuzzy multi-agent system is proposed for electric arc furnace steel making processes. Each process of electric arc furnace steel making is assigned to be an agent, which works independently whilst coordinates and cooperates with other acquaintance agents. Adaptive neuro-fuzzy inference system is used to generate agents' knowledge bases. Contract net protocol is used as negotiation protocol in the proposed multi-agent system.

© 2008 Elsevier Ltd. All rights reserved.

1. Introduction

Iron and steel industry is an important basic industry for any industrial economy providing the primary material for construction, automobile, machinery and other industries. Furthermore, the iron and steel industry is capital as well as energy intensive. The importance of effective process control in this industry on cost and energy reduction and environment protection is by no means less than that in other industries.

Iron and steel industry should mainly rely on the new integrated production processes to improve productivity, reduce energy consumption, and maintain competitiveness in the market. The adoption of new production processes has brought an urgent need to be studied. Process control is the key function in the production management. Without rational process controlling systems, the potential benefits of new production processes can not be fully realized. Furthermore, a high degree of real-time operation and dynamic adjustment capabilities are required. In particular, the coordination of different production stages must be considered so as to achieve global goals of the entire production processes.

In most steel companies, the principal production planning and scheduling techniques have been essentially manual techniques with little computerized decision support beyond that provide by a central mainframe database. These manual techniques are based on the know-how and the professional experience of expert people who have worked in the plant for years. Recently, there have been several attempts to introduce automation and optimization principles (Sven & et al., 2005) to handle the problems within a steel plant because of the complexity of the problems involved in steel making, on one hand, and the great possibility of saving energy

and money, on the other hand (Cowling, Ouelhadj, & Petrovic, 2003; Cowling, Ouelhadj, & Petrovic, 2004; Lahdelma & Rong, 2006; Ouelhadj, Petrovic, Cowling, & Meisels, 2004). Other benefits of computerized decision support may include reduced planner workload, reduced company dependency on key planning staff and more uniform plan quality.

The characteristics of a steel industry are:

- Steel production is a multi-stage process, logically and geographically distributed, involving a variety of production processes (Ouelhadj et al., 2004).
- There are different problems and different problem solving methods for steel making stages.
- These manual techniques used in steel making processes are based on the know-how and the professional experience of expert people who have worked in the plant for years.
- The output of some stages is the input of some other stages, so the integrated process is necessary.
- Making contracts requires negotiation between buyer and supplier.

According to the above characteristics of steel industry, a steel automation system is needed to represent distribution and integration exist in such industry. Using multi-agent systems depicts such capabilities.

Multi-agent systems are intelligent distributed approach suited for applications that are modular, changeable and complex. They have autonomy, integration, reactivity and flexibility capabilities. They can dynamically integrate new agents, remove existing agents, or upgrade agents independently. So, multi-agent systems using fuzzy systems can improve steel production to be less dependent on expert staff.

In this paper a multi-agent architecture is proposed for electric arc furnace steel making processes. Adaptive neuro-fuzzy inference system is used to generate agents' knowledge bases. Contract net

* Corresponding author. Tel.: +98 21 641 3034; fax: +98 21 641 3025.

E-mail addresses: zarandi@aut.ac.ir (M.H. Fazel Zarandi), pegah.ahmadpour.ext@nsn.com (P. Ahmadpour).

protocol is used as negotiation protocol in the proposed multi-agent system.

The rest of this paper is organized as follows. Section 2 illustrates the electric arc furnace steel making processes. Sections 3 and 4 review the multi-agent systems and adaptive neuro-fuzzy inference systems and their characteristics respectively. The proposed fuzzy multi-agent system is presented in Section 5. The proposed negotiation protocol and negotiation algorithm are presented in Sections 6 and 7 respectively. Finally, Section 8 includes conclusion and future researches.

2. Steel production from scrap metals

Steelmaking from scrap metals involves melting scrap metal, removing impurities and casting it into the desired shapes where, electric arc furnaces (EAF) are often used. The EAFs melt scrap metal in the presence of electric energy and oxygen. Production of steel from scrap can be economical on a much smaller scale.

2.1. Electric arc furnace

The electric arc furnace (EAF) is the principle furnace type for the electric production of steel. The primary application of the EAF is for the remelting of steel scrap; however, EAFs can be charged with limited amounts of iron scrap, pig iron, and direct reduced iron.

2.2. Ladle metallurgy

After tapping from an EAF, molten steel for high quality or specialty applications is subjected to further refining in a number of alternative processes collectively known as ladle metallurgy, ladle refining, or secondary steelmaking. The objectives of these processes are:

- *Deoxidization* – Removal of oxygen
- *Degassing* – Removal of hydrogen
- *Desulfurization* – To sulfur concentrations as low as 0.002%
- *Microcleanliness* – Removal of undesirable nonmetallic elements
- *Inclusion morphology* – Changing the composition of remaining impurities to improve the microstructure of the steel
- *Mechanical properties* – Increases toughness, ductility, and transverse properties

2.3. Ingot casting

After it is refined, the molten steel is tapped into refractory lined ladles. The steel is then teemed (poured) into a series of ingot molds. Ingots range in size from several hundred pounds for specialty steels to 300 tons for large forging applications. Slab ingots range in weight from 2 to 50 tons – averaging about 20 tons. Ingot shape and weight are selected to meet the specific requirements of the product to be made. After the ingots solidify, the ingot molds are stripped and the ingots are placed in soaking pits for heating and to equalize the internal and external temperature. Following the soak, the ingots are hot rolled in large primary rolling mills to produce slabs, blooms, or billets depending on the final products desired.

Steel production is a distributed process, involving different problem solving methods, requiring integration between different steps. Multi-agent systems are distributed systems well suited for such systems. The steel processes are naturally uncertain and dependent on human know-how. Fuzzy systems can appropriately cope with this uncertainty. Multi-agent systems and fuzzy systems are addressed in Sections 3 and 4, respectively.

3. Multi-agent systems

Multi-agent systems are an emerging sub-field of artificial intelligence that is concerned with a society of agents interacting in order to solve a common problem. Multi-agent systems are a relatively new field of study. They have only been studied since about 1980, and the field has only gained widespread recognition since about the 1990s. In 1980 a group of researchers held the first Distributed Artificial Intelligence (DAI) workshop at MIT to discuss issues concerning intelligent problem solving with systems consisting of multiple problem solvers, termed multi-agent systems (Oliveira, Fischer, & Stepankova, 1998; Wooldridge, 2002). From these beginnings, the multi-agent paradigm has grown into a major research area. Multi-agent systems have become more and more important in many aspects of computer science by introducing the issues of distributed intelligence and interaction. Multi-agent systems seem to be a natural metaphor for understanding and building a range of what were called artificial social systems. They represent a new way of analyzing, designing, and implementing complex software systems.

According to Ferber and Ghallab (1988), an agent is a physical or abstract entity that can act in its environment. This entity possesses a partial representation of the environment and can communicate with other agents. Its behavior is determined by its knowledge, its perception, and its interaction with other agents. For Maes (1995) of MIT Media Lab, one of the pioneers of agent research, autonomous agents are computational systems that inhabit some complex dynamic environment, sense and act autonomously in this environment, and by doing so realize a set of goals or tasks for which they are designed. Jennings (2000) stated that one of the most widely used working definitions, which an increasing number of researchers find useful, is Wooldridge and Jennings' (1995) definition: an agent is a computer system that is situated in some environment, and that is capable of flexible and autonomous action in this environment in order to meet its design objectives. By flexible we mean that the system must be responsive, proactive, and social.

Various definitions have been proposed for the term multi-agent system (MAS). Durfee, Lesser, and Corkill (1989b) defined a MAS as a loosely coupled network of problem solvers that work together to solve problems that are beyond the individual capabilities or knowledge of each problem solver. More recently, Ferber defined a MAS as a system composed of a population of autonomous agents, which interact with each other to reach common objectives, while simultaneously each agent pursues individual objectives. Oliveira et al. (1998) defined a MAS as a collection of, possibly heterogeneous, computational entities, having their own problem solving capabilities and which are able to interact in order to reach an overall goal.

Several theoretical architectures have been defined in literature to describe the basic structure of cognitive agents. The most common modules of the internal architecture of an agent are: perception, execution, self-knowledge, acquaintance knowledge or social knowledge, domain knowledge, reasoning, learning, cooperation, and communication (Ferber, 1999; Jennings, Sycara, & Wooldridge, 1998; Labidi & Lejouad, 1993; Ouelhadj et al., 2004; Shen, Norrie, & Barthes, 2001; Weiss, 1999; Wooldridge, 2002; Wooldridge & Jennings, 1994).

In order for MAS to solve common problems coherently, the agents must cooperate, coordinate, and communicate amongst themselves. Cooperation, coordination, and communication are central to a MAS. Agents need to interact with other agents to achieve their objectives either because they do not have sufficient capabilities or resources to complete their problem solving alone, or because there are interdependencies between the agents that

follow from being situated in a common environment (Cheng, Henry Chan, & Lin, 2006; Choi, Liu, & Chan, 2001; Dell'Orco & Giordano, 2002; Faratin, Sierra, & Jennings, 1998; Fatima, Wooldridge, & Jennings, 2004; He, Jennings, & Leung, 2003; Huang & Sy-cara, 2002; Lahdelma & Rong, 2006; Lopes, Mamede, Novais, & Coelho, 2001).

Work on cooperative distributed problem solving began with the work of Durfee, Lesser, and Corkill (1989a) and Durfee et al. (1989b) where he defined that cooperative distributed problem solving studies how a loosely coupled network of problem solvers can work together to solve problems that are beyond their individual capabilities. Each problem solving node in the network is capable of sophisticated problem solving and can work independently, but the problems faced by the nodes cannot be completed without cooperation. Cooperation is necessary because no single node has sufficient expertise, resources, and information to solve a problem and different nodes may have expertise for solving different parts of the problem.

A classical technique for distributed task allocation is the contract net protocol. It is the most common and best-studied mechanism for distributed task allocation in agent-based systems (Shen et al., 2001; Wooldridge, 2002). The contract net protocol is a high level protocol for achieving efficient cooperation introduced by Smith based on a market-like protocol (Smith, 1980). The basic metaphor used in the contract net protocol is, as the name of the protocol suggests, contracting. Smith took his inspiration from the way that companies organize the process of putting contracts out to tender in public markets. The contract net protocol received an enthusiastic welcome from the distributed artificial intelligence community. In this approach, a decentralized market structure is assumed and agents can take on two roles: a manager and a contractor. The basic premise of this form of cooperation is that if an agent cannot solve an assigned task using local resources/expertise, it will try to find other willing agents with the necessary resources/expertise to solve the tasks. The tasks are assigned using a contracting mechanism.

Multi-agent systems can work more similar to real world using fuzzy systems. Applying neuro-fuzzy inference system dedicates the learning and adaptation capabilities to agents.

4. Adaptive neuro-fuzzy networks (ANFIS)

Adaptive neuro-fuzzy networks code the fuzzy if-then rules into a neural network like structure and then use a learning algorithm to minimize the output error using training data. There are a number of methods to develop adaptive fuzzy networks (Jang, 1993; Lin & Lee, 1992; wang & Mendel, 1992; Yager, 1994). The decision was made to use the adaptive neuro-fuzzy inference system (ANFIS) structure because of its accuracy (Chiu, 1997; Sadeghian & Lavers, 2000). The ANFIS approach uses Gaussian functions for fuzzy sets, linear functions for the rule outputs, and Sugeno's inference mechanism (Takagi & Sugeno, 1985). The parameters of the network are the mean and standard deviation of the membership functions (antecedent parameters) and the coefficients of the output linear functions (consequent parameters). The ANFIS learning algorithm is then used to obtain these parameters. This learning algorithm is a hybrid algorithm consisting of the gradient descent and the least-squares estimate. Using this hybrid algorithm, the rule parameters are recursively updated until acceptable error is reached. Each iteration includes two sweeps, forward and backward. In the forward pass, the antecedent parameters are fixed and the consequent parameters are obtained using the linear least-squares estimates. In the backward pass, the consequent parameters are fixed and the output error is back-propagated through the network and the antecedent parameters are updated

accordingly using the gradient descent method. The application of these methods to determine the steel making parameters is described in next section.

5. The proposed multi-agent architecture

The proposed multi-agent architecture is organized as a population of cognitive, autonomous, and heterogeneous agents, see Fig. 1: the User agent, Ingot Casting (IC) agent, Vacuum Degassing (VD) agent, Ladle Furnace (LF) agent, Electric Arc Furnace (EAF) agent, and Customer agent. Each agent is responsible for locally executing the tasks of the resource in charge, and has the ability to co-operate and communicate with other agents.

- User agent is responsible for receiving customer orders. It manages and announces the orders to produce, and deals with dynamic changes of order conditions.
- IC agent acts as a supplier of ingots to the User agent. Since ingots are produced in different weights, the processing time of casting stage differs in different orders.
- VD agent is responsible for controlling degassing process. The management of degassing stage involves determining the degassing parameters such as vacuum pressure and calculation of the processing time.
- LF agent is responsible for providing refined molten steel. This includes determining the amounts of additives required to be added in ladle stage to change the molten steel chemical properties to what the customer ordered.
- EAF agent acts as a supplier of molten steel to LF agent. It provides primary molten steel by melting scraps.
- Customer agent is responsible for making orders and negotiating with user agent to make contract.

The main modules of each agent include: self-knowledge, social-knowledge, domain knowledge, cooperation knowledge, inference, learning, communication interface, and user interface. Fig. 2.

5.1. User agent

User agent relates all other agents to know their status (active or inactive) in every moment to be able to manage new received orders and deal with real-time events.

Upon receiving a new order from customer, user agent checks the status of other agents to be sure that they are active and able to do the job. Every order should be assessed by user agent to know if it is feasible. A function is used for assessment. The assessment

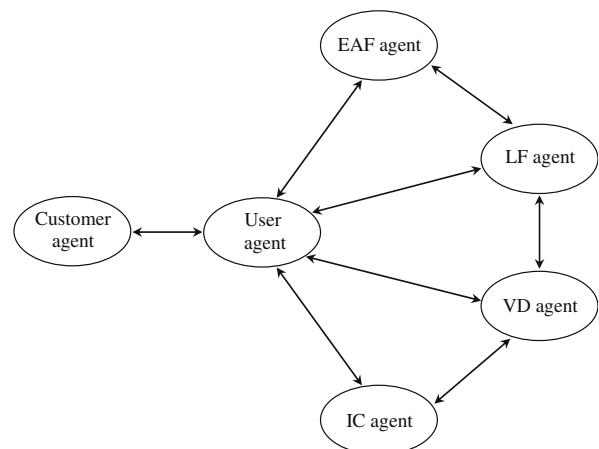


Fig. 1. Proposed multi-agent system.

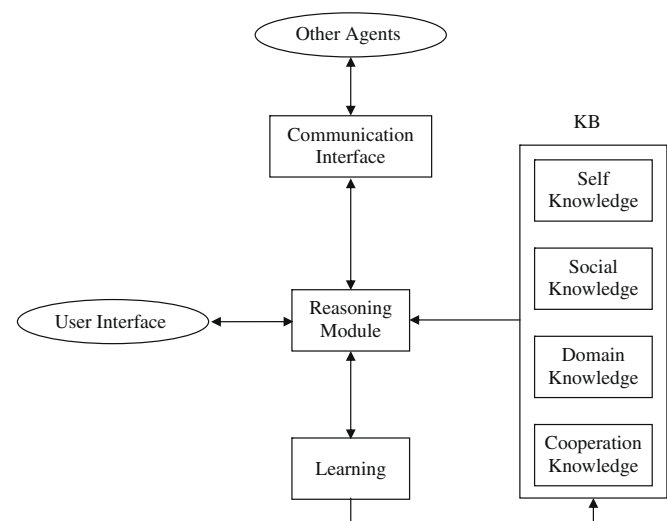


Fig. 2. Agent structure.

function depends on the amounts of additives and delay of delivery, as follows:

Assessment function = cost of additives
+ cost of delivery date differences

Feasibility of every received order is assessed using the above function. For feasible orders, a message containing “feasible” status will be sent to customer agent. Otherwise, the message implies infeasibility.

5.2. IC agent

There are many types of ingots. Ingots range in weight from 2 to 50 tons. Ingot shape and weight are selected to meet the specific

Table 1
VD agent FISs.

No.	Output	Inputs	MF	Number of rules	Error
1	PB _{vb}	T1 T2 H	Gaussian	11	0.027
2	db1	T1 T2 H	Gaussian	16	0.077
3	db2	T1 T2 H	Gaussian	13	0.02
4	Time	T1 T2 H	Gaussian	14	0.09

requirements of the product to be made. The casting processing time depends on the size of ingots, since the solidification time is different for various ingot sizes.

The main task of IC agent is determination of the amount of needed molten steel for the ingots requested by customer, and prediction of the total casting processing time for current order.

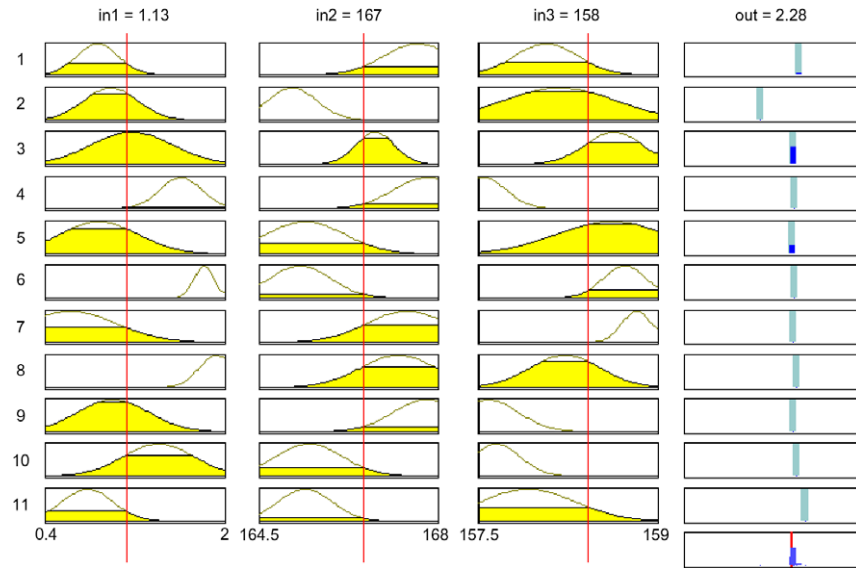
5.3. VD agent

In vacuum degassing (VD), molten steel is subjected to a vacuum for vacuum control, temperature control and degassing (hydrogen removal).

IC agent is responsible for degassing process and determination of degassing parameters, such as vacuum pressure, the amount of neutral gas (Argon), and the processing time. For this purpose, an adaptive neuro-fuzzy inference system (ANFIS) is used. The input data includes the amount of hydrogen, input and output temperature. The output data includes vacuum pressure, amount of neutral gas, and processing time. With these data, a MIMO (Multiple Input–Multiple Output) fuzzy system should be generated. For simplification, it is transferred into four MISO (Multiple Input–Single Output) fuzzy systems. For each of these MISOs a fuzzy rule base is generated using ANFIS. The hybrid approach is used for learning. These systems after training stage are shown in Table 1, where, the temperatures of input and output molten steel are defined as T1 and T2, and H defines the amount of required Hydrogen, and P_v, db1 and db2 define vacuum pressure, amount of neutral gas in stages 1 and 2, respectively.

1.	If <i>H</i> is very low and <i>T1</i> is very high and <i>T2</i> is rarely low then <i>Pv</i> is $(-0.1848 * H - 0.6485 * T1 + 1.8305 * T2 - 178.1581)$
2.	If <i>H</i> is rarely low and <i>T1</i> is very strongly low and <i>T2</i> is moderate then <i>Pv</i> is $(3.1852 * H - 4.1816 * T1 + 4.2968 * T2 + 7.7689)$
3.	If <i>H</i> is rarely high and <i>T1</i> is moderate and <i>T2</i> is very high then <i>Pv</i> is $(0.5049 * H + 0.9166 * T1 - 0.4582 * T2 - 78.2941)$
4.	If <i>H</i> is very high and <i>T1</i> is very strongly high and <i>T2</i> is very strongly low then <i>Pv</i> is $(0.1966 * H - 0.5228 * T1 + 0.3739 * T2 + 30.1955)$
5.	If <i>H</i> is low and <i>T1</i> is very low and <i>T2</i> is high then <i>Pv</i> is $(-0.3597 * H + 0.2353 * T1 + 0.3492 * T2 - 92.0205)$
6.	If <i>H</i> is strongly high and <i>T1</i> is strongly low and <i>T2</i> is strongly high then <i>Pv</i> is $(0.9346 * H + 0.3159 * T1 - 0.515 * T2 + 30.6456)$
7.	If <i>H</i> is very strongly low and <i>T1</i> is high and <i>T2</i> is very strongly high then <i>Pv</i> is $(-0.9718 * H - 0.0313 * T1 - 0.0824 * T2 + 21.7863)$
8.	If <i>H</i> is very strongly high and <i>T1</i> is rarely high and <i>T2</i> is rarely low then <i>Pv</i> is $(-0.0359 * H - 0.061 * T1 - 0.0968 * T2 + 28.5883)$
9.	If <i>H</i> is moderate and <i>T1</i> is strongly high and <i>T2</i> is strongly low then <i>Pv</i> is $(0.1442 * H - 0.2932 * T1 + 0.3026 * T2 + 3.2343)$
10.	If <i>H</i> is high and <i>T1</i> is rarely low and <i>T2</i> is very low then <i>Pv</i> is $(-3.1018 * H - 1.6691 * T1 + 1.8293 * T2 - 5.2197)$
11.	If <i>H</i> is strongly low and <i>T1</i> is low and <i>T2</i> is low then <i>Pv</i> is $(7.0911 * H + 0.8726 * T1 - 0.9245 * T2 - 2.0054)$

Fig. 3. P_v rule base.

Fig. 4. P_v membership functions.Table 2
LF agent FISs.

No.	Output	Number of inputs	MF	Number of rules	Error
1	CaO	6	Gaussian	23	0.27
2	FeSi	7	Gaussian	14	0.18
3	Al	8	Gaussian	15	0.072
4	FeMn	7	Gaussian	32	0.518
5	FeCr	9	Gaussian	15	0.177
6	Time	4	Gaussian	12	0.043

These rule bases are saved as knowledge base in VD agent. As an example, P_v rule base and membership functions are shown in Figs. 3 and 4, respectively.

5.4. LF agent

In ladle metallurgy, alloys are added to the molten steel, which is then reheated to produce the desired metallurgy.

LF agent is responsible for refining process and determination of its parameters, such as the amounts of additives, and the processing

1. If S2 is rarely low and Cr2 is moderate and Ni2 is high and Temperature is rarely high then Time is $(0.5138*S2 - 0.0357*Cr2 - 1.2426*Ni2 + 0.0843*Temperature - 106.6996)$
2. If S2 is moderate and Cr2 is very high and Ni2 is rarely low and Temperature is rarely low then Time is $(0.1389*S2 + 1.6269*Cr2 - 1.4204*Ni2 + 2.4173*Temperature - 4145.3457)$
3. If S2 is low and Cr2 is very low and Ni2 is strongly high and Temperature is very low then Time is $(-13.0052*S2 + 61.592*Cr2 - 296.1599*Ni2 - 0.7135*Temperature + 307.2859)$
4. If S2 is high and Cr2 is rarely high and Ni2 is low and Temperature is strongly high then Time is $(2.2872*S2 + 0.818*Cr2 + 0.3753*Ni2 + 0.1714*Temperature - 406.5699)$
5. If S2 is strongly low and Cr2 is very strongly high and Ni2 is very high and Temperature is high then Time is $(-1.4643*S2 - 1.1535*Cr2 + 0.9456*Ni2 + 0.5525*Temperature - 768.843)$
6. If S2 is strongly high and Cr2 is high and Ni2 is rarely high and Temperature is moderate then Time is $(3.0166*S2 - 0.3019*Cr2 - 0.8707*Ni2 - 1.2534*Temperature + 2057.7422)$
7. If S2 is very low and Cr2 is strongly high and Ni2 is moderate and Temperature is very strongly low then Time is $(0.1647*S2 + 2.7019*Cr2 + 0.4491*Ni2 + 0.1365*Temperature - 495.0239)$
8. If S2 is very strongly high and Cr2 is very strongly low and Ni2 is strongly low and Temperature is very rarely low then Time is $(28.3107*S2 + 35.9924*Cr2 - 35.407*Ni2 - 2.0716*Temperature + 12.542)$
9. If S2 is very rarely low and Cr2 is low and Ni2 is very low and Temperature is strongly low then Time is $(17.8234*S2 + 12.9095*Cr2 - 22.1442*Ni2 - 0.9092*Temperature + 72.6366)$
10. If S2 is rarely high and Cr2 is strongly low and Ni2 is very strongly high and Temperature is low then Time is $(-932.2819*S2 - 59.9852*Cr2 - 969.0433*Ni2 + 30.9102*Temperature - 47.7353)$
11. If S2 is very high and Cr2 is very rarely low and Ni2 is very strongly low and Temperature is very high then Time is $(-21.8419*S2 - 22.0158*Cr2 - 0.213*Ni2 + 1.6037*Temperature - 119.1887)$
12. If S2 is strongly low and Cr2 is rarely low and Ni2 is very rarely low and Temperature is very strongly high then Time is $(0.6518*S2 - 1.019*Cr2 + 0.2464*Ni2 - 0.4935*Temperature + 928.5418)$

Fig. 5. LFTIME rule base.

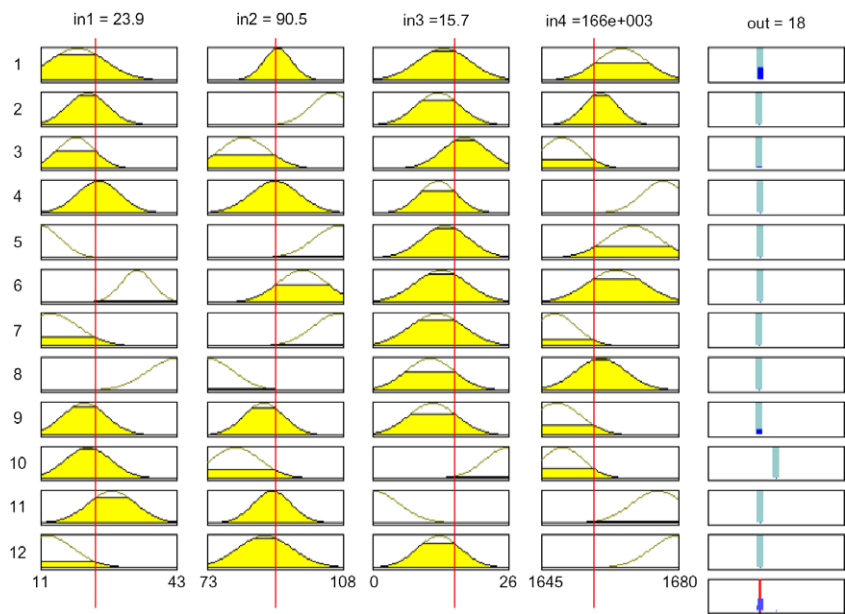


Fig. 6. LFTIME membership functions.

time. To do so, an adaptive neuro-fuzzy inference system (ANFIS) is used. The input data includes the chemical properties of input molten steel and required chemical properties of output molten steel. The output data includes the amounts of needed additives to change the chemical properties of molten steel, and processing time. With these data, a MIMO (Multiple Input–Multiple Output) fuzzy system should be generated. For simplification, MIMO is transferred into six MISO (Multiple Input–Single Output) fuzzy systems. After variable selection using Sugeno approach, for each of these MISOs a fuzzy rule base is generated using ANFIS. The hybrid approach is used for learning. These systems after training stage are shown in Table 2.

These rule bases are saved as knowledge base in LF agent. As an example, LFTIME rule base and membership functions are shown in Figs. 5 and 6, respectively.

5.5. Customer agent

The main task of customer agent is making orders and negotiation with user agent to make contract. It sends a primary order with required delivery time to user agent and receives the possible delivery time from it. If these two delivery times are different, customer agent should make decision about accepting the delivery time. This decision making is based on a utility function. The customer utility function is a function of differences between customer required properties and possible properties.

6. Proposed negotiation protocol

The individual agents present different objectives and constraints, and no agent possesses a global view of the entire agency, therefore cooperation is essential to allow the agents to adjust their local schedules in order to achieve global objectives.

The contract net protocol is a high level protocol for achieving efficient cooperation based on a market-like protocol. The contract net protocol has been extensively used for inter-agent cooperation. The latter include dynamic task allocation via self-bidding which leads to better agreements, it provides natural load balancing (as busy agents need not bid), agents can be introduced and removed dynamically, and it is a reliable mechanism for distributed control and failure recovery.

A negotiation protocol, based on the contract net protocol, is proposed for inter-agent cooperation. In the basic contract net protocol no agent takes any responsibilities before reaching mutual agreements on the tasks, which can cause the degradation of the global coordination performance. By attaching the commitment duration to the negotiation messages, the performance of the basic contract net protocol is improved especially for dynamic environments where desired tasks and available resources may be continuously changing. The commitment duration specifies the time windows by which the agents must respond to a given negotiation message.

The negotiation protocol is a three-step process. It begins after the Customer agent has initiated a session by sending a request message to the User agent to supply a set of ingots. The steps of the negotiation protocol are described below.

6.1. Task announcement

Upon receiving the request message from the Customer agent to supply a set of ingot orders, the User agent tries to generate the set of required ingots. Then, the User agent negotiates the production of ingots with the Ingot casting agent. The User agent issues an announcement message (User-announcement) to the IC agent to produce ingots according to the customer order.

Sender:	User agent
Receiver:	IC agent
Message Type:	Task-announcement
Message Task:	Produce-Ingots
Deadline:	Time by which the IC agent must respond with a bid
Content:	The description of chemical properties and dimension of ingots and its latest required readiness date.

Upon receiving the request message from the User agent to produce a set of ingot orders, the Ingot Casting agent tries to produce the set of required ingots. Then, the Ingot Casting agent negotiates the supply of molten steel with the VD agent. The Ingot Casting agent is-

sues an announcement message (IC-announcement) to the VD agent to supply the molten steel necessary to produce the ingots.

The IC-announcement message must specify the following essential information:

Sender:	IC agent
Receiver:	VD agent
Message Type:	Task-announcement
Message Task:	Supply-molten steel
Deadline:	Time by which the VD agent must respond with a bid
Content:	The description of chemical properties and amount of molten steel and its latest required readiness date.

Upon receiving the IC-announcement message, the VD agent analyses the content of the message. Then the negotiation at the next level begins for the purpose of ordering molten steel to LF agent. The VD agent sends the VD-announcement message to LF agent to request the providing molten steel, in which it adjusts the required readiness dates to take into account the Vacuum degassing (VD) processing time. For LF agent, the VD-announcement message must specify the following information:

Sender:	VD agent
Receiver:	LF agent
Message Type:	Task-announcement
Message Task:	Supply-molten steel
Deadline:	Time by which the LF agent must respond with a bid
Content:	The description of chemical properties and amount of molten steel and its latest required readiness date.

Upon receiving the VD-announcement message, the LF agent analyses the content of the message. Then the negotiation at the next level begins for the purpose of ordering molten steel to EAF agent. The LF agent sends the LF-announcement message to EAF agent to request the providing molten steel, in which it adjusts the required readiness dates to take into account the Ladle furnace (LF) processing time. The LF-announcement message must specify the following information:

Sender:	LF agent
Receiver:	EAF agent
Message Type:	Task-announcement
Message Task:	Produce-molten steel
Deadline:	Time by which the EAF agent must respond with a bid
Content:	The description of chemical properties and amount of molten steel and its latest required readiness date.

A failure to respond to a task-announcement message, before its specified deadline, is interpreted as meaning that the acquaintance cannot fulfill the task.

6.2. Bidding

A bid represents an offer to execute the task specified in the LF-announcement. EAF agent will inspect the LF-announcement mes-

sage and will decide whether or not it should respond with a bid, considering the quality of the EAF schedule resulting from adding the requested molten steel to its current engagements. If it chooses to respond, it will count its locally processing time and send an EAF-bid message to the LF agent. The EAF-bid message must specify the following information:

Sender:	EAF agent
Receiver:	LF agent
Message Type:	Bid
Message Task:	Produce-molten steel
Deadline:	Time by which the LF agent must respond with either a contract or a review of the bid.
Content:	The description of chemical properties and amount of molten steel and its proposed production date.

After receiving the bid, or once the bid-reception deadline has expired, the LF agent evaluates the EAF-bids received and selects the EAF agent with the best bid based on the earliest production dates proposed. The LF agent sends an LF-bid message to the VD agent, containing information about the readiness date of the molten steel to be produced by the LF agent. The readiness date of molten steel is the production date proposed by the EAF agent to which is added the LF processing time. The bid is used at the next negotiation level. The LF-bid message must specify the following information:

Sender:	LF agent
Receiver:	VD agent
Message Type:	Bid
Message Task:	Supply-molten steel
Deadline:	Time by which the VD agent must respond with either a contract or a review of the bid.
Content:	The description of chemical properties and amount of molten steel and its proposed production date.

After receiving the bid, or once the bid-reception deadline has expired, the VD agent evaluates the LF-bid received. The VD agent sends a VD-bid message to the Ingot Casting agent, containing information about the readiness date of the molten steel to be produced by the VD agent. The readiness date of molten steel is the production date proposed by the LF agent to which is added the VD processing time. The bid is used at the next negotiation level. The VD-bid message must specify the following information:

Sender:	VD agent
Receiver:	IC agent
Message Type:	Bid
Message Task:	Supply-molten steel
Deadline:	Time by which the IC agent must respond with either a contract or a review of the bid.
Content:	The description of chemical properties and amount of molten steel and its proposed production date.

After receiving the bid, or once the bid-reception deadline has expired, the IC agent evaluates the VD-bid received. The IC agent sends an IC-bid message to the User agent, containing information

about the readiness date of the ingots to be produced by the IC agent. The readiness date of ingots is the production date proposed by the VD agent to which is added the IC processing time. The bid is used at the next negotiation level. The IC-bid message must specify the following information:

Sender:	IC agent
Receiver:	User agent
Message Type:	Bid
Message Task:	Produce- Ingots
Deadline:	Time by which the User agent must respond with either a contract or a review of the bid.
Content:	The description of chemical properties and the number of ingots and its proposed production date.

After receiving the bid, or once the bid-reception deadline has expired, the User agent evaluates the IC-bid received. The User agent sends a User-bid message to the Customer agent, containing information about the readiness date of the ingots to be supplied by the User agent. The bid is used at the next negotiation level. The User agent informs Customer agent about feasibility of the order.

A failure to respond to a bid, before its specified deadline, is interpreted as meaning that the bid had been rejected.

6.3. Contracting

After Customer agent has received the bid from User agent, it examines the new readiness date (possible delivery time) proposed by user agent and must respond before the deadline with the following alternatives:

- Accept the bid and send a Customer-contract message to the User agent. The User agent, in turn, sends a User-contract message to the IC agent before the specified deadline. Then, the IC agent sends an IC-contract message to the VD agent before the deadline expires. The VD and LF agents send VD-contract and LF-contract respectively. The agents can then start to produce their special productions. The contracts messages must specify the following essential information:

Sender:	(Customer) (User Agent) (IC agent) (VD agent) (LF agent)
Receiver:	(User Agent) (IC Agent)(VD agent) (LF agent) (EAF agent)
Message Type:	Establish-contract
Message Task:	(Supply-Ingots) (Produce-Ingots) (Produce-molten steel)

- The IC agent restarts a re-negotiation session with a new announcement message to revise the bid. To do so, an additional allowance tolerance on their properties is applied. The flexibility of the molten steel properties means an attempt to satisfy the required readiness date of them. The re-negotiation process iterates cyclically until the specified number of bids are made. At each re-negotiation session the other agents adjust their local schedule using the new property ranges.

The negotiation algorithm for negotiation between user agent and customer agent is described in the next section.

7. Negotiation algorithm

Negotiation between user agent and customer agent continues until the specified number of orders is sent. The steps of algorithm include:

- Step 1. Sending an order to user agent;
- Step 2. Inferring the possible delivery time;
- Step 3. Checking the feasibility of the order by user agent;
- Step 4. Sending the answer to customer agent: if feasible then go to step (6); otherwise, go to step (5);
- Step 5. Informing customer agent about infeasibility; go to step (9);
- Step 6. Informing customer agent about feasibility and possible delivery time;
- Step 7. Checking the acceptability of order by customer agent; if acceptable then go to step (8); otherwise, go to step (9);
- Step 8. Comparing the current order with the best order; if its better then replace it;
- Step 9. Stop criteria. If true then go to step (11);
- Step 10. Search for a new offer; go to step (1);
- Step 11. Stop and make decision for contracting.

The stop criteria in step (9) include sending a certain number of orders to user agent, where this number is specified by the user.

Decision making in step (11) is conducted by assessing accepted candidates. The best order is selected to contract. If there is no order as best order, negotiation will finish without making contract.

The proposed negotiation algorithm provides a way for agents to reach an agreement by testing similar orders. In order to make a similar order, it's assumed that order properties are flexible in an accepted domain. Different combinations of property values will be checked to find the best one. Best order means the order which is feasible for user agent and has best utility value for customer agent. Using this algorithm more customer orders will lead to making contract.

It should be noted that, in this research, an Experimental simulation prototype was developed in Microsoft MATLAB7, where, six agents (EAF, LF, VD, IC, User and Customer) can cooperate. The inference systems of agents include adaptive neuro-fuzzy inference systems (ANFISs). To reach a reliable system, a hybrid learning algorithm was used which works based on error reduction approach in forward and backward passes. These acceptable errors can be read from right columns of Tables 1 and 2.

8. Conclusions and future works

This paper, presented a multi-agent system for electric arc furnace steel making processes. Each process of electric arc furnace steel making was assigned to an agent, which could work independently whilst coordinates and cooperates with other acquaintance agents. Adaptive neuro-fuzzy inference system was used to generate agents' knowledge bases. Contract net protocol was developed as negotiation protocol in the proposed multi-agent system.

The proposed system takes a global perspective to steel making processes and presents a mechanism for interaction of agents. Using fuzzy systems helps the multi-agent system to work more similar to real world. Applying adaptive neuro-fuzzy inference system, dedicates the learning and adaptation capabilities to agents.

In this paper it's assumed that there is one agent for every process, but in real world more than one agent can work on one process. Order making mechanism is supposed out of the scope of this research. Improvement of order making mechanism is a potential concept to work on.

References

- Cheng, C.-B., Henry Chan, C.-C., & Lin, K.-C. (2006). Intelligent agents for e-marketplace: Negotiation with issue trade-offs by fuzzy inference systems. *Decision Support Systems*, 42, 626–638.
- Chiu, S. C. (1997). Extracting fuzzy rules from data for function approximation and pattern classification. In D. Dubios, H. Prade, & R. R. Yager (Eds.), *Fuzzy information engineering: A guided tour of applications*. New York: John Wiley & sons.
- Choi, S. P. M., Liu, J., & Chan, S.-P. (2001). A genetic agent-based negotiation system. *Computer Networks*, 37, 195–204.
- Cowling, P. I., Ouelhadj, D., & Petrovic, S. (2003). A Multi-agent architecture for dynamic scheduling of steel hot rolling. *Journal of Intelligent Manufacturing*, 14, 457–470.
- Cowling, P. I., Ouelhadj, D., & Petrovic, S. (2004). Dynamic scheduling of steel casting and milling using multi-agents. *Production Planning and Control*, 15, 1–11 [special issue on the application of multi agent systems to Production Planning and Control].
- Dell'Orco, M., Giordano, R. (2002). Web community of agents for integrated logistics of industrial districts. In *IEEE proceeding of the 36th Hawaii international conference on system sciences*.
- Durfee, E., Lesser, V. R., & Corkill, D. D. (1989a). Cooperative distributed problem solving. In E. A. Feigenbaum, A. Barr, & P. R. Cohen (Eds.), *Handbook of artificial intelligence* (pp. 83–147). Addison-Wesley.
- Durfee, E., Lesser, V. R., & Corkill, D. D. (1989b). Trends in cooperative distributed problem solving. *IEEE Transactions on Knowledge and Data Engineering*, 1(1), 63–83.
- Faratin, P., Sierra, C., & Jennings, N. R. (1998). Negotiation decision functions for autonomous agents. *Robotics and Autonomous Systems*, 24(3–4), 159–182.
- Fatima, S., Wooldridge, M., & Jennings, N. R. (2004). An agenda based framework for multi-issues negotiation. *Artificial Intelligence*, 152(1), 1–45.
- Ferber, J. (1999). *Multi-agent systems: An introduction to distributed artificial intelligence*. London: Addison-Wesley.
- Ferber, J., & Ghallab, M. (1988). Problématique des univers multi-agents. In *Actes des Journées Nationales du PRC-IA, Cepadus -edition, Toulouse, France*.
- He, M., Jennings, N. R., & Leung, H. F. (2003). On agent-mediated electronic commerce. *IEEE Transactions on Knowledge and Data Engineering*, 15(4), 985–1003.
- Huang, P., & Sycara, K. (2002). A computational model for online agent negotiation. In *IEEE proceeding of the 35th Hawaii international conference on system sciences*.
- Jang, J. S. R. (1993). ANFIS: Adaptive-network-based fuzzy inference system. *IEEE Transactions on Systems, Man, and Cybernetics*, 23(3), 665–685.
- Jennings, N. R. (2000). On agent-based software engineering. *Artificial Intelligence*, 117(2), 277–296.
- Jennings, N. R., Sycara, K., & Wooldridge, M. (1998). A roadmap of agent research and development. *Autonomous Agents and Multi-Agent Systems*, 1(1), 7–38.
- Labidi, S., & Lejouad, W. (1993). *De l'intelligence artificielle au syst'emes multi-agents*. Writing technical report, INRIA, France.
- Lahdelma, R., & Rong, A. (2006). Scrap charge optimization using fuzzy chance constrained linear programming. In *New developments in artificial intelligence and the semantic web proceedings of the 12th Finnish artificial intelligence conference STeP*.
- Lin, C., & Lee, C. S. G. (1992). Neural network-based fuzzy logic control and decision system. *IEEE Transactions on Computers*, 40(12), 1320–1336.
- Lopes, F., Mamede, N., Novais, A. Q., & Coelho, H. (2001). Negotiation tactics for autonomous agents. *IEEE*.
- Maes, P. (1995). Artificial life meets entertainment: Life like autonomous agents. *Communication of the ACM*, 38(11), 108–114.
- Oliveira, E., Fischer, K., & Stepankova, O. (1998). Multi-agent systems: Which research for which applications. *Robotics and Autonomous Systems*, 7(1–2), 91–106.
- Ouelhadj, D., Petrovic, S., Cowling, P. I., & Meisels, A. (2004). Inter-agent cooperation and communication for agent-based robust dynamic scheduling in steel production. *Advanced Engineering Informatics*, 18, 161–172.
- Sadeghian, A. R., & Lavers, J. D. (2000). Recurrent neuro-fuzzy prediction for multi-step prediction of v-i characteristics of electric arc furnace. In *The ninth IEEE international conference on fuzzy system, fuzzy IEEE 2000* (Vol. 1, pp. 110–115).
- Shen, W., Norrie, D. H., & Barthes, J. P. A. (2001). *Multi-agent systems for concurrent intelligent design and manufacturing*. London: Taylor & Francis.
- Smith, R. (1980). The contract net protocol: High level communication and control in distributed problem solver. *IEEE Transactions on Computers*, 29(12), 1104–1113.
- Sven, J. et al. (2005). An agent based online system for the planning and observation of steel production. In *Proceedings of the international conference on autonomous agents, proceeding of the 4th international conference on autonomous agents and multi agent systems, 2005, AAMAS 05* (pp. 155–160).
- Takagi, T., & Sugeno, M. (1985). Fuzzy identification of systems and its application to modeling and control. *IEEE Transactions on Systems, Man and Cybernetics*, 15, 116–132.
- Wang, L. X., & Mendel, J. M. (1992). Back-propagation of fuzzy systems as nonlinear dynamic system identifiers. In *Proceedings IEEE international conference on fuzzy systems* (pp. 1409–1418).
- Weiss, G. (1999). *Multi-agent systems: A modern approach to distributed artificial intelligence*. Cambridge, Massachusetts, London, England: The MIT Press.
- Wooldridge, M. (2002). *An introduction to multi-agent systems*. Chichester, England: John Wiley & Sons Ltd.
- Wooldridge, M., & Jennings, N. R. (1994). Agent theories, architectures, and languages: A survey. In M. Wooldridge, & N. R., Jennings (Eds.), *Proceedings of ECAI-Workshop on Agent Theories, Architectures and Languages* (pp. 1–32). Amsterdam, The Netherlands.
- Wooldridge, M., & Jennings, N. R. (1995). Intelligent agents: Theory and practice. *Knowledge Engineering Review*, 10(2), 115–152.
- Yager, R. R. (1994). *Essentials of fuzzy modeling and control*. New York: Wiley.