Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 96 (2016) 531 – 539

Procedia Computer Science

20th International Conference on Knowledge Based and Intelligent Information and Engineering Systems, KES2016, 5-7 September 2016, York, United Kingdom

A multi-agent approach to computational optimization of metal forming processes

Jerzy Duda[a*], Andrzej Maciół[a], Piotr Maciół[b]

[a]*AGH University of Science and Technology, Faculty of Management, Department of Applied Computer Science, Gramatyka St. 10, Krakow, 30-067, Poland*

[b]*AGH University of Science and Technology, Faculty of Metals Engineering and Industrial Computer Science, Department of Applied Computer Science and Modelling, Czarnowiejska St. 66, Krakow, 30-054, Poland*

Abstract

In the paper we present a new original approach to a complex multi-phase process optimization problem that relies on dividing the optimization task into partial tasks related to the implementation of individual phases. To implement this concept, we propose a multi-agent approach. Its practical realization is shown on the example of manufacturing process for auto body parts made of the Advanced High Strength Steels (AHSS). Although it is a rather simple process consisting of only two phases, we assumed, however, that the results obtained will allow to extend further research to more complex problems. We present the operating principles of a multi-agent model, the flow of the messages between agents, and the architecture of the system. To ensure the proper speed of the whole system a simple and flexible multi-agent framework called Eve was used to develop a prototype of the system. Research performed, as well as preliminary tests have shown that a multi-agent approach can be successfully applied to reduce complexity of the whole optimization processes. Due to splitting a single, complex optimization process into several, partially independent optimization processes, delegating them to autonomous agents, and application of a knowledge-based reasoning system, significant advantage could be observed over to the solutions described so far in the literature.

Peer-review under responsibility of KES International

Keywords: Optimization; multi-agent systems; numerical modelling; steel rolling

* Corresponding author. *E-mail address:* jduda@zarz.agh.edu.pl

Peer-review under responsibility of KES International
doi:10.1016/j.procs.2016.08.230

1. Introduction

One of the most important applications of computational methods nowadays is modelling of physical phenomena. Such models can be applied for problems in physics, natural science (meteorology, seismology, etc.), material science and many others. Each of those domains has its own mathematical methods, dedicated for specific requirements. Sometimes even singular cases are solved with a very unique, dedicated models. Moreover, mathematical methods can be implemented with various numerical solutions. Numerical modelling is widely represented, among others, in manufacturing. Almost all branches of industry are currently supported with modelling. Unfortunately, numerical models usually refer to particular stages of a production chain and they are not useful for optimizing the whole chain. The opposite approach requires modelling of the whole production chain with integrated and coherent set of methods. The issue is further complicated by the fact that most manufacturing processes are heterogeneous, with significantly different stages. Optimization of the whole chain with coherent numerical model is theoretically possible, however, very time-consuming due to large number of independent variables [1]. Only simple processes can be efficiently optimized or models of particular operations must be significantly simplified, which limits reliability of such models and, consequently, the whole optimization. To the best of the authors knowledge, there are no published works describing multicriteria optimization of manufacturing processes with more than two distinct and significantly different operations.

Presented research is focused on manufacturing of metal products from a high-grade steel. Such production process consists of multiple operations, including significantly different methods of liquid and solid metal alloys processing. Single manufacturing process frequently includes more than a dozen technological operations. One of such processes is multi-operational forging with drawing of several forging steps and heat treatment. There are many numerical models developed and published for each step of this process [2]. However, each of the steps has different, frequently contradictory criteria of optimization. Similar issues have been identified for steel works scheduling. Some attempts to solve the scheduling problem with multi-agent systems have been described in [3] [4]. In this paper, the authors present a concept for optimizing manufacturing processes in a similar way.

Since there is no published research in this area, several basic, methodological issues must be solved initially. Our research is focused on the development of a computer-aiding tool able to solve complex optimization processes based on a multi-agent approach. The aim of our solution is to solve complex optimization problems without undue simplification of numerical models of sub-processes and within a reasonable computing time. The further goal is to improve parallelization chance of the optimization process. Furthermore, it is expected that agents, based on accumulated knowledge, will be able to dynamically choose the most suitable numerical models, depending on technological issues and expected reliability of simulation. Previous research proved that this approach can be successful, however only single-step problems were considered [2, 5]. Application of multi-agent environment will extend this capability to multi-step processes.

In our research we took a relatively simple case of the process consisting of two phases, assuming that the results obtained for that problem, will allow to extend further research to more complex problems.

2. Exemplary case

In this paper, optimization of manufacturing process for auto body parts made of the Advanced High Strength Steels (AHSS) is discussed. Detailed description of the process can be found in [6]. Due to the stage of research, a relatively simple, two-step problem is chosen, to simplify description of the methodology. The discussed process consists of multi-pass hot rolling (first step) and laminar, controlled cooling (second step). Hot rolling is aimed at reducing the thickness of the sheet, as well as preparing requested microstructure and thermal conditions for laminar cooling. Laminar cooling is aimed at obtaining required microstructure. Also from a modelling point of view of both steps are different. For hot rolling, a simplified model considering number of passes and cooling conditions between passes was applied by the authors. The simplified model was used instead of direct computations with Finite Element Method (FEM) to decrease the computing time. Empirical equations for phase transformations (according to Avrami equation) was used for modelling of microstructure evolution during laminar cooling. The sensitivity analysis was performed, with the phase composition for cooling and finishing rolling temperature and grain size for

rolling. Subsequently, reduced sets of optimized variables were specified: initial temperature, interpass times, heat exchange coefficients, rolling velocities (rolling) and cooling rates (cooling). Finally, the optimization was performed by applying various techniques, including methods inspired by nature.

Due to a simple structure of the problem and replacing of direct FEM model with a fast, simplified model for rolling, the application of a quasi-optimization with integrated, two-step model became possible. Authors of [6] proved that with the sensitivity analysis, the time to solve the optimization problem can be significantly reduced, however, at the expense of excluding of some independent variables from the optimization process. In this paper an alternative, multi-agent based approach is introduced.

3. Concept of problem solution

The concept of optimization of a complex multi-phase process relies on dividing the whole optimization task into partial tasks related to the implementation of individual production phases. For obvious reasons, optimization criteria for each phase are different and can be contradictory. Multi-agent simulation approach allows for a kind of 'handshake' between the results of individual solutions and in turn allows to reach a compromise solution, which would be the one acceptable for all agents and as close to the optimal solution as possible in particular circumstances.

In the analyzed example we decided that the optimization process will be implemented through three agents:

1. A representative of the customer (CustomerAgent), the purpose of whom is to determine the best phase composition after cooling to the room temperature possible to obtain under real production conditions;
2. An agent representing cooling process (CoolingAgent), whose task is to design the cooling process in such a way that a steel sheet brought to the room temperature has a phase composition acceptable for the Customer Agent (though not necessarily identical);
3. An agent representing rolling process (RollingAgent), whose task is to design the sheet rolling process so that it is possible to obtain the most favorable initial state of material beginning of the laminar cooling, allowing to achieve the objectives of Customer Agent.

4. Agent models

The workflow of the multi-agent system is described as an UML sequence diagram and depicted in Figure 1. As shown CustomerAgent initiates the whole simulation process and controls its flow.

In order to describe knowledge and behavior of the agents we used a popular belief-desire-intention (BDI) model, however the system does not follow the BDI framework at the operational level.

4.1. CustomerAgent

Beliefs

A. Parameters of the steel required by the customer

1. required volume fraction of martensite
2. required level of bainite
3. maximum permissible volume fraction of martensite
4. maximum permissible level of bainite
5. the step of volume fraction of martensite destending
6. the step of level of bainite destending
7. steel chemical composition

B. Current set of parameters obtained from the agents

1. volume fraction of martensite
2. level of bainite
3. status of the combination, i.e. whether it is OK for the customer, whether it is worth to be tried again or it should be rejected.

C. Optimization parameters

1. MaxIterationCounter – maximum number of dialogs with the processing agents
2. IterationCounter – current number of iteration

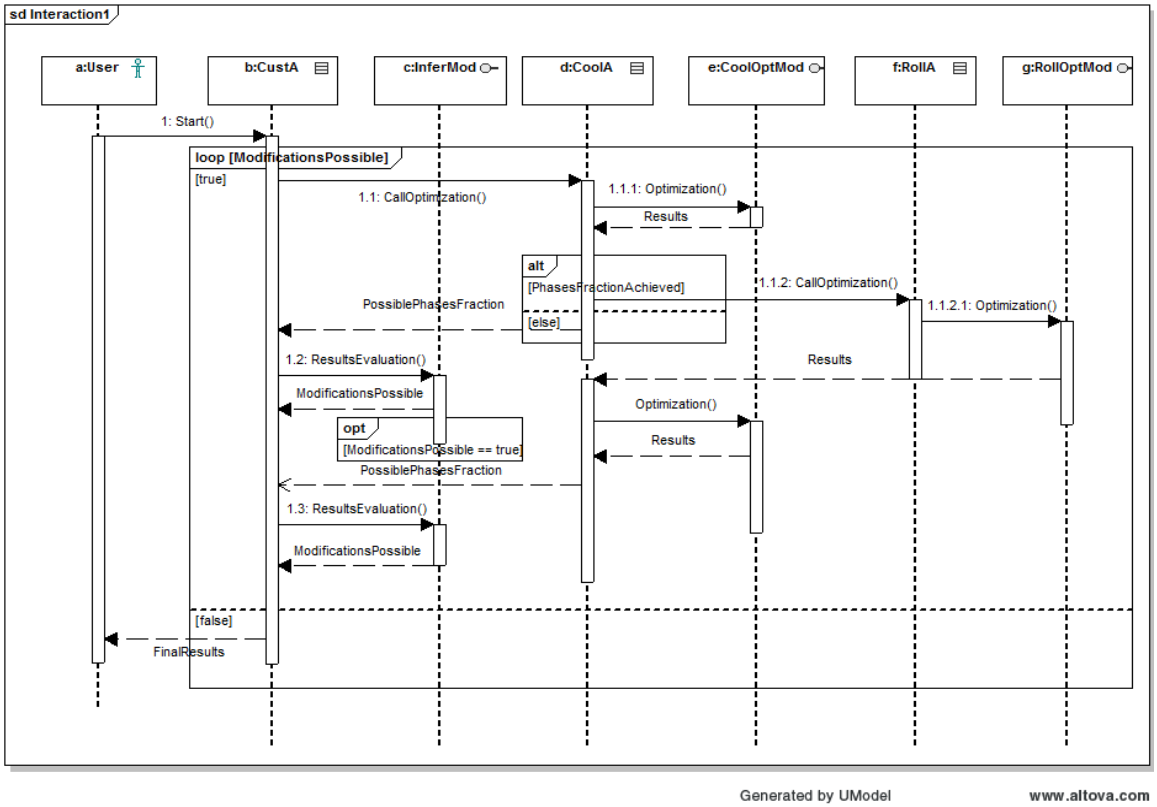

Fig 1. The flow of the messages between agents

D. Ruleset determining whether the parameters are acceptable for the customer are expressed in the following form:

```
RULE 1
  IF
    proposed by CoolingAgent volume_fraction_of_martensite <=
    required_volume_fraction_of_martensite AND proposed by CoolingAgent
    level_of_bainite <= required_level_of_bainite
  THEN
    Result = "OK"
RULE 2
  IF
    proposed by CoolingAgent volume_fraction_of_martensite >
    required_volume_fraction_of_martensite AND
    proposed by CoolingAgent level_of_bainite <= required_level_of_bainite
  AND
    IterationCounter < MaxIterationCounter
  THEN
```

```
volume_fraction_of_martensite = volume_fraction_of_martensite -
the_step_of_volume_fraction_of_martensite_destending,
Result = "Try again" ...
```

Desire

Best combination of parameters that are possible to obtain in given production circumstances.

Intentions

The agent realizes the following optimization Plan:

A. Send a required set of the following parameters to CoolingAgent:

1. required volume fraction of martensite,
2. required level of bainite ,
3. steel chemical composition.

B. Receive the message from CoolingAgent containing the parameters possible to achieve at the current stage of simulation.

C. If the current status of the simulation is "Try again" it resends the requirement to the CoolingAgent, otherwise the simulation ends

4.2. CoolingAgent

Beliefs

A. CoolingAgent uses a database to determine initial parameters of the cooling process. Such parameters can be stored either as deterministic values or as intervals.

B. Required parameter values received from CustomerAgent:

1. required volume fraction of martensite,
2. required level of bainite.

C. A set of externals methods able to determine the values of optimization variables in a reasonable time. For the preliminary experiments we chose evolutionary algorithm, but other heuristics can be used in the future as well. A main procedure for the heuristics looks as follows:

```
1. set optimization variables values based on data stored in a database
(e.g. as interval middles)
2. while StopCondition not met (e.g. goal achieving and/or maximum number
of iterations)
  3. run the modelling tool for cooling process
  4. if required values of parameters are met then StopCondition = true
  5. else modify value of the variables
6. end while
```

Desires

Set the values for the following parameters:

1. finishing rolling temperature T_f
2. heat transfer coefficient in the first stage of cooling α_1,
3. time of the first stage of cooling tc_1,
4. time of the cooling in air tc_2,
5. heat transfer coefficient in the third stage of cooling α_3,
6. volume fraction of martensite after rolling,
7. level of bainite after rolling.

Intentions

The CoolingAgent may realize two plans, one as the response for CustomerAgent request and one for the RollingAgent request.

I. Plan for CustomerAgent request

A. Receive the required parameters from CustomerAgent

1. required volume fraction of martensite
2. required level of bainite

B. Execute an external optimization method

C. If required parameters cannot be achieved send a message to the CustomerAgent along with the obtained values, otherwise send a message to the RollingAgent with the following parameters:

1. finishing rolling temperature T_f
2. volume fraction of martensite after rolling,
3. level of bainite after rolling.

II. Plan for RollingAgent request

A. Receive from RollingAgent a message with optimization results of the rolling process (independent on whether the required values have been met, as RollingAgent may obtain better or worse values in terms of various aspects, so it is necessary to recalculate the cooling process) along with possible to obtain values of finishing rolling temperature T_f, volume fraction of martensite after rolling, and level of bainite after rolling.

B. Execute the external optimization methods (as in the first plan) with additional constraints

1. finishing_rolling_temperature T_f, >= possible_finishing_rolling_temperature T_f
2. volume_fraction_of_martensite_after_rolling >= possible_volume_fraction_of_martensite_after_rolling,
3. level_of_bainite_after_rolling >= possible_level_of_bainite_after_rolling.

C. Send the optimization results to CustomerAgent.

4.3. Rolling agent

Beliefs

A. RollingAgent, similarly to CoolingAgent, uses a database to determine initial parameters of the rolling process. It contains the information about the relationship between the given parameters and decision parameters.

B. The following parameters received from CoolingAgent:

1. finishing rolling temperature T_f
2. required volume fraction of martensite after rolling
3. required level of bainite after rolling
4. steel chemical composition

C. A dedicated program [7] is used to achieve final parameters.

Desires

Values of the following parameters:

1. initial temperature T_0,
2. time intervals after pass 3 t_3,
3. time intervals after pass 4 t_4,
4. time intervals after pass 5 t_5.

Intentions

A. Receive parameters from CoolingAgent to charge the rolling optimization model. Additional constraints include calculated rolling forces which do not exceed limited value for the rolling mill and the additional criterion are the costs related to energy consumption calculated by rolling forces and the time of passes and minimized for the set of output parameters.

B. Send the results of optimization to CoolingAgent.

5. Architecture of the system

The main challenge for the proposed system is the speed, as the system has to support a real production process and decisions on how to set production parameters have to be made quickly. Traditionally Java-based frameworks are used to develop multi-agent systems, of which JADE (Java Agent Development Environment) is the most popular. There are also frameworks dedicated to BDI architecture like Procedural Reasoning System (PRS), JACK

Intelligent Agents or JADEX, just to name only a few popular ones. In order to perform initial experiments we chose Eve web-based agent framework [8] that allows to use either Java or JavaScript language to describe the agents and their behavior. The agents communicate with each other using simple JSON-RPC protocol over HTTP or XMPP network. In the Eve framework, contrary to the majority of multi-agent frameworks, agents are not continuously running as a thread on a server, but have to be triggered externally to execute some tasks. In our case an agent is triggered by an incoming message sent by other agent. Thanks to use of such a simple and flexible architecture we were able to use in memory SQLite database to store simulation parameters for each agent. In our experiments we did not notice any downgrade in performance due to the use of SQL database, but when the number of parameters will be much higher, it will be more reasonable to use of one of key-value databases like Redis. The agents are supported by an externally inference module. In our researches we used an original rule-based reasoning system developed in Faculty of Management of AGH University of Science and Technology, described i.a. in [9].

Rebit System is an example of Business Rules Management Systems (BRMS). It has a modular architecture that includes first of all a knowledgebase editor, a knowledge repository, and an inference client that incorporates two engines – one for triggering rules, and another one for performing workflow actions. All modules can be utilized either as desktop and web applications, or separately, as a part of SOA architecture services using SOAP protocol. What distinguishes the Rebit inference engine from other BRMS engines is that the inference process may be suspended for example when waiting for a user's interaction or for some computational task to be finished and resumed after it receives necessary facts. All the rules triggered before suspension are still valid. Another unique feature of the Rebit inference engine is the ability of controlling the inference process, what may reduce the number of necessary interactions with the user or with some external application (less number of facts are needed due to the optimized triggering of rules).

The inference module is run as a web service and any agent can call it. This can potentially create a bottleneck in the system, however, in the current version of the system, only CustomerAgent uses interference engine, once it gets a message from CoolingAgent, so the inference engine service is called rarely enough not to cause any downgrades in overall speed of the system. The overall architecture of the system is shown in Figure 2.

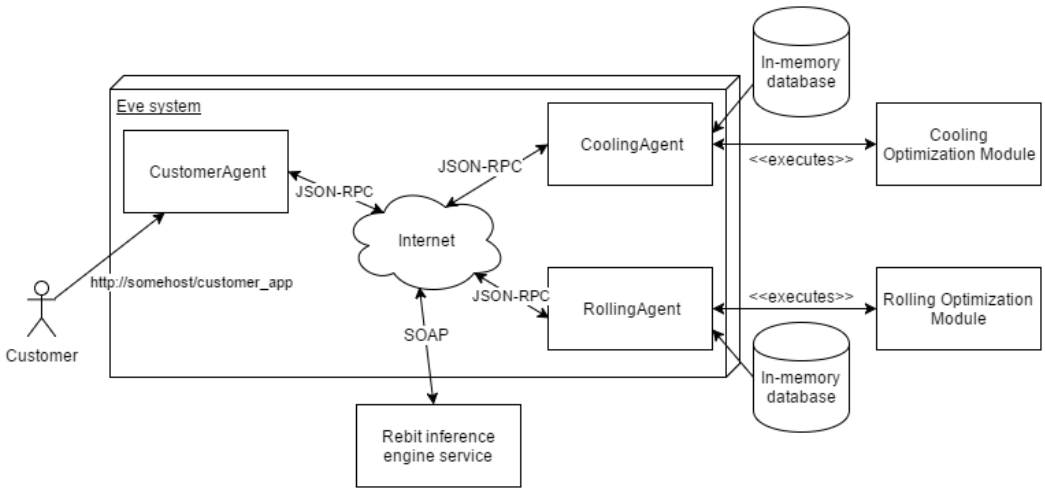

Fig. 2. Diagram of the system architecture.

The main drawback of the current system is the time necessary to solve cooling optimization model, what usually takes few minutes for a single run, however, in the future we plan to rewrite the optimization engine to use some distributed optimization system like the one described in [10].

The multi-agent system is currently in an early prototype phase. However, preliminary results met the expectations and confirmed the previous experience of the authors. It has been confirmed i.a. that application of the Rebit inference engine can significantly reduce the inference cycle (more than 20% comparing with an uncontrolled forward inference), as it was shown in [11]. The Eve system that has been used for our prototype solution also

showed adequate performance i.e. did not introduce any excessive delay especially when compared with the performance of optimization modules, however, for simulations of more complex production processes, a more sophisticated architecture may be required.

6. Conclusions and further works

In this paper a new approach to optimization of design of complex manufacturing process is presented. The research performed, as well as preliminary tests have shown that a multi-agent approach can be successfully applied to reduce complexity of optimization processes. Such solutions have not yet been applied in the area of computer-aided technology design. Although the aforementioned works [3] [4] concerning the planning of metallurgical production use the concept of multi-agent systems, but they significantly differ from our system in the way of knowledge representation used by the agents. There are no known works on the use of advanced knowledge modelling methods and innovative, highly efficient inference mechanisms for solving such complex decision problems.

To ensure the proper speed of the whole system a simple and flexible multi-agent framework called Eve has been used to develop a prototype of the system. Although the agents have been coded in Java, in the future version of the system they can be rewritten in JavaScript language allowing to use such techniques like Web Workers or Web Sockets to further improve overall system performance. Due to splitting of a single, complex optimization process into several, partially independent optimization processes, delegating them to autonomous agents and application of knowledge-based reasoning system, some advantages appeared. In solution presented in [6], calculation of goal function required solving of a model of full process. In presented solution, the goal function requires only a part of the model, that allows to apply more detailed (but more complex) models. Different optimization techniques can be applied for different agents, according to particular requirements and specific conditions. Negotiating allows to achieve a 'good-enough' solution for a customer (represented by an agent) in relatively short time. Simultaneously, requirements and constraints for each part of manufacturing process are fulfilled.

The prototype solution, presented in the paper has been designed for a particular problem of AHSS sheet rolling for automotive industry. The further step of the research is the verification of this approach for more complex problems, like e.g. multi-operational forging.

Acknowledgments

This work is supported by the NCN, project No. 2014/15/B/ST8/00187 and AGH University statutory research No. 11/11.200.327.

References

1. Rauch, Ł., Skiba, M., Kusiak, J.: Computer system dedicated to optimization of production processes and cycles in metal forming industry. Comput. Methods Mater. Sci. Vol. 14, No. 1, 3–12 (2014).
2. Macioł, P., Bureau, R., Sommitsch, C.: An Object-Oriented Analysis of Complex Numerical Models. Key Eng. Mater. 611-612, 1356–1363 (2014).
3. Mattik, I.: Integrated Scheduling of Continuous Casters and Hot Strip Mills: A Block Planning Application for the Steel Industry. Springer Fachmedien Wiesbaden (2013).
4. Cowling, P.I., Ouelhadj, D., Petrovic, S.: Dynamic scheduling of steel casting and milling using multi-agents. Prod. Plan. Control. 15, 178–188 (2004).
5. Rec, T., Macioł, A., Macioł, P.: Application of rule-based system to support FEM simulations in hot rolling process. Comput. Methods Mater. Sci. 11, 284–289 (2011).
6. Szeliga, D., Sztangret, Ł., Kusiak, J., Pietrzyk, M.: Optimization as a support for design of hot rolling technology of dual phase steel strips. AIP Conf. Proc. 1532, (2013).
7. Rauch, Ł., Bzowski, K., Bednarski, K., Kuziak, R., Radwański, K., Pietrzyk, M.: Komputerowy system wspomagający projektowanie technologii walcowania blach cienkich ze stali wielofazowych. Pr. Inst. Metal. Żelaza. 67, 129–135 (2015).
8. d. Jong, J., Stellingwerff, L., Pazienza, G.E.: Eve: A Novel Open-Source Web-Based Agent Platform. In: 2013 IEEE International Conference on Systems, Man, and Cybernetics. pp. 1537–1541 (2013).
9. Maciol, A., Maciol, P., Jedrusik, S., Lelito, J.: The new hybrid rule-based tool to evaluate processes in manufacturing. Int. J. Adv. Manuf. Technol. 79, 1733–1745 (2015).
10. Duda, J., Dłubacz, W.: Distributed evolutionary computing system based on web browsers with javascript. In: 11th International

Conference, PARA 2012, Helsinki, Finland, June 10-13, 2012. pp. 183–191. Applied Parallel and Scientific Computing. Springer Berlin Heidelberg (2013).

11. Baster, B., Maciol, A.: The Inference Efficiency Problem in Business and Technological Rules Management Systems. ICSOFT (1). (2010).