



Multi-site scheduling with fuzzy concepts

Jürgen Sauer *, Gerd Suelmann, Hans-Jürgen Appellath

*University of Oldenburg, Department of Computer Science, Escherweg 2, D-26121 Oldenburg,
Germany*

Received 1 September 1997; accepted 1 December 1997

Abstract

The objective of multi-site scheduling is to support the scheduling activities of a global scheduler and schedulers in distributed production plants in a cooperative way. A global schedule generated on a global level must be translated into detailed schedules as part of the local scheduling process. In case of disturbance, feedback between the local and global levels is essential. Global level data are normally aggregated, imprecise, or estimated. Previous methods focused on local production sites, in most cases without coordination. In this work we present an approach that considers the adequate modeling and processing of imprecise data for global level scheduling within a multi-site scheduling system based on fuzzy concepts. One of the goals is to create a robust prescription for the local scheduling systems which helps to reduce the effort of coordination and rescheduling. © 1998 Elsevier Science Inc. All rights reserved.

Keywords: Multi-site scheduling; Knowledge-based systems; Fuzzy systems

1. Introduction

Scheduling integrates different tasks of production planning and control (PPC) systems to plan and carry out the manufacturing of products on the available resources respecting time and cost restrictions. It can be assigned to the phases resource/material management, time management, job release, and production control of PPC.

* Corresponding author. E-mail: sauer@informatik.uni-oldenburg.de.

In most cases scheduling problems are addressed for single production centers or specific areas of responsibility [1]. To schedule the production, predictive techniques are employed which create schedules in advance for a given time period [2]. Reactive techniques [3] are used to adjust the schedule to actual conditions caused by disturbances in the scheduling environment. If we look at enterprises that produce at different production sites and must transport intermediate products between two locations, we have to consider new aspects that are to be solved by an adequate approach.

Fig. 1 illustrates the hierarchical two-level structure of multi-site scheduling reflecting the organizational structure to be found in business. On the global level requirements are generated for intermediate products (IPx) manufactured in individual locations. Local scheduling (at individual locations) deals with the transformation into concrete production schedules which represent the assignment of operations to machines. On both levels predictive as well as reactive problems are addressed, not only to generate schedules but also to adapt them to the actual situation in the production process.

Due to the distribution of production processes to different plants some specific problems arise which are:

- Complex interdependences between production processes that are performed in different plants, e.g.
 - temporal relations between intermediate and final products, e.g. if an intermediate product that is manufactured in plant A is needed in plant B,
 - the same item can be manufactured in different plants (possibly at different costs),
 - the transport of parts between different plants needs transportation capacities and is time- and cost-intensive.
- The type of information used for global scheduling, because generalized data are normally used instead of precise data, these are e.g.

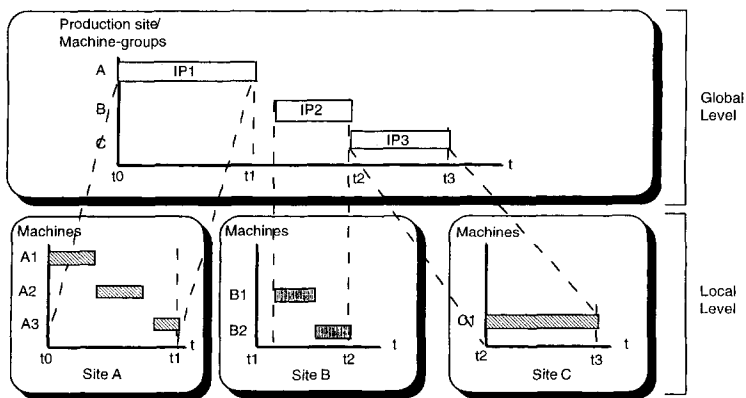


Fig. 1. Global and local scheduling.

- capacity information referring to machine groups instead of single machines,
- information on the duration of manufacturing processes for intermediate or final products that are often estimated values.
- Integration of existing (local) scheduling systems for individual plants that accomplish the local realization of global requirements.
- The necessity of coordination of decentralized scheduling activities for all plants within one enterprise since several levels of scheduling with their specific scheduling systems have to work cooperatively in a dynamic distributed manufacturing environment.
- The uncertainty about the actual “state” in individual plants.

In this paper an architecture for a multi-site scheduling system is presented that integrates fuzzy techniques to deal with the imprecise data especially found on the global scheduling level.

2. MUST: A multi-site scheduling system

In the project MUST (Multi-Site Scheduling System) [4], to which this work belongs, an approach is implemented that integrates predictive and reactive components for both global and local scheduling. The objective of this approach is the reduction of complexity of distributed scheduling and improving the quality of the solution at the same time. Fig. 2 shows the architecture of the MUST-system. On the global level both generating the requirements for the local plants (global/master schedule) and reactive correction of this plan can be accomplished. New components to deal with these tasks were developed within

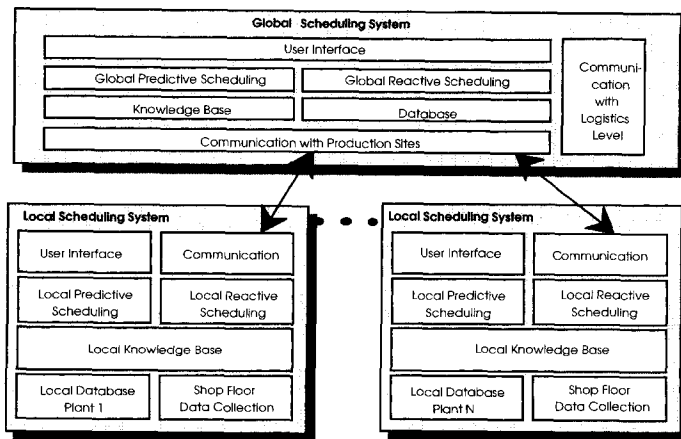


Fig. 2. The MUST system architecture.

the project. They include a global leitstand [5] which, according to Fig. 1, shows the distribution of jobs to local production sites on the global scheduling level and solves conflicts during the scheduling process with a heuristic reactive component, whereby the system tries to solve conflicts as locally as possible.

On the local scheduling level previously existing techniques of local/reactive scheduling can be integrated, e.g. from [6]. The communication between the individual levels has a specific importance, since it enables the coordination of the different scheduling levels. In this sense, at least the following items of information are sent.

- From the global to the local level:
 - the global schedule consisting of information on internal orders, affiliated intermediate products, machine groups to use, time windows that should (possibly) be met, and required quantities of intermediate products,
 - unexpected events that effect the local level (e.g. the cancellation of an order).
- From the local to the global level:
 - the local realization of the global requirements with information on internal orders, affiliated intermediate products, start and end time of all locally scheduled activities and used machine groups,
 - appearance of failure events,
 - suggestions for possible local rescheduling.

A typical multi-site scheduling procedure then looks like:

1. A global-level schedule with an initial distribution of internal orders to local production sites is generated (*global predictive scheduling*).
2. Based on the global schedule, the local plants draw up their detailed local production schedules (*local predictive scheduling*).
3. In case of local disturbances, the local reactive scheduler first tries to remedy them locally by interactive repair (*local reactive scheduling*).
4. If problems cannot be solved on the local level or the modified local schedule influences other local schedules (inter-plant dependencies), the global level has to be called again. Global scheduling can then cause a redistribution of internal orders to local plants and adapt the global schedule (*global reactive scheduling*).
5. The local plants adjust to the changes in the global schedule.

In order to achieve congruence of the global schedule and its local transpositions, steps 3–5 might be done more than once to maintain consistency.

3. The global scheduling problem

The scheduling problem on the global level consists of the transformation of incoming external orders for final products to internal orders for intermediate products and the assignment of the intermediates to machine groups of individ-

ual production sites respecting given due dates and additional objectives, for example minimization of transportation costs. The second task is the adjustment of the global schedule due to disturbances in the global scheduling environment which may be caused by events from the local level, e.g. machine groups are not available, as well as from the external environment, e.g. order cancellations.

A scheduling problem can be modeled by a five-tuple (R, P, O, HC, SC) [7], where R denotes the set of required resources, P the set of producible products, O the set of actual orders, and HC and SC stand for the sets of hard and soft constraints, respectively. Applied to global scheduling, this describes the following:

- Resources e.g. the groups of apparatuses, transportation times and costs between production sites, and the amount of all materials in a production plant.
- A set of final products, where each can be produced in form of different variants. Every variant consists of several intermediate products that can be manufactured using different resources with varying durations and capacity loads. Each produced item requires a specific amount of materials and tools.
- Additionally, the intermediate products are limited in their order of production by a precedence relation.

External orders for the manufacturing of an individual product out of the set of final products coupled with quantity, start time, due date, and priority.

- Global hard constraints that must be met, e.g.
 - all external orders must be planned,
 - exactly one variant with all its intermediate products must be selected for every external order,
 - intermediate products cannot be processed before the start time of the corresponding external order,
 - the order of intermediate products according to the precedence relation must be maintained.

Depending on the specific problem, additional hard constraints are possible.

- Global soft constraints (reflecting the scheduling objectives) that can be partially violated, e.g.
 - the given due date of a product should not be exceeded by any of its intermediate products,
 - transportation times and costs should be minimized,
 - all production sites should be used equally,
 - the solution of the global scheduling problem should be as robust as possible, i.e. it should give enough flexibility for a local scheduler to react to local disturbances without affecting the other sites. This can be achieved amongst others by using buffer times in the time windows for local production and by trying to optimize the load balancing on the machine groups.
 - the requirement for material should be minimal.

The solution of the global scheduling problem is a global schedule, which determines the requirements for the local scheduling level in terms of temporal assignments of intermediate products to machine groups (production plants) as shown in Fig. 1.

Many of the data used on the global scheduling level are aggregated, e.g. machine groups instead of individual machines, or only imprecisely known at the time of schedule generation, e.g. estimated durations and capacities, in order to keep the scheduling problem comprehensible and manageable. Events on the local scheduling level are not predictable, but they should have as low impact on future rescheduling as possible. Transportation times are very difficult to determine, since they largely depend on the actual situation. Despite the large amount of imprecise data, the schedule should be as “good” as possible. The heuristics that are used so far are not capable of reaching very good results because of the imprecision of data. In most cases, the uncertainty is detoured by using a discrete value instead or reducing the model, e.g. by neglecting transportation times.

To support this functionality, we need concepts for specification and processing of the fuzzy knowledge of the global scheduling level. In the following, we therefore present an approach that employs concepts of fuzzy logic [8,9] to handle global predictive scheduling. After an overview of previous approaches we describe the modeling of fuzzy data in the global scheduling domain, their application with the help of rules and inference strategies, and the implementation as a fuzzy controller.

4. Previous approaches to fuzzy scheduling

Fuzzy logic allows the modeling of imprecise scheduling knowledge with linguistic variables defined by membership functions showing the degree of preciseness of the data and the reasoning about the imprecise data by using fuzzy rules. This is a means of representation and processing that is transparent and comprehensible for the user.

While the presented fuzzy-based approach is innovative within the field of multi-site scheduling, there already exist several approaches to solve general or local scheduling problems by techniques founded on fuzzy set theory and fuzzy logic [10–13]. Most of the approaches to solve scheduling problems use conventional (from operations research) or knowledge-based methods [1,2]. The latter are often based on simple heuristics (SPT, EDD, FIFO etc.) that provide sufficient solutions with respect to their evaluation functions. Widely used are the concepts of constraints and heuristics (in the form of rules) for both representation and solution of scheduling problems [1,2]. On the level of global scheduling, there exist only few approaches to date. Traditional PPC-systems solve the distribution problem implicitly because materials or or-

ders (jobs) are allocated to specific machine groups and therefore production sites. Mainly, there are only simple methods in this area that use rough estimates and have no feedback mechanism. More recent approaches, which could also be applied to global scheduling deploy multi-level concepts [2,4], e.g. with communicating agents [14].

5. Fuzzy global scheduling

To solve the global scheduling problem with a fuzzy-based system (controller) the following sub-problems need to be addressed [10,15]:

- modeling and transformation of scheduling data into a knowledge representation that can be handled by a fuzzy controller (fuzzification),
- processing of the fuzzy scheduling knowledge towards a decision by means of given rules and integration of fuzzy arithmetic to deal with imprecise or vague data and
- transformation of the fuzzy scheduling decision into crisp scheduling data (defuzzification).

Fig. 3 visualizes the schematic architecture of the fuzzy controller we use to solve the global predictive scheduling tasks. The data for the controller is made available in an appropriate format by fuzzification. The rule bases together with the inference strategy determine the method for generating a predictive global schedule. The controller is designed so generally that it can be also used for the solution of other scheduling problems, e.g. job shop scheduling problems or local scheduling problems, just by changing the data, the rules and the control strategy. Therefore we provide functionality for an easy change or extension of the fuzzy knowledge base.

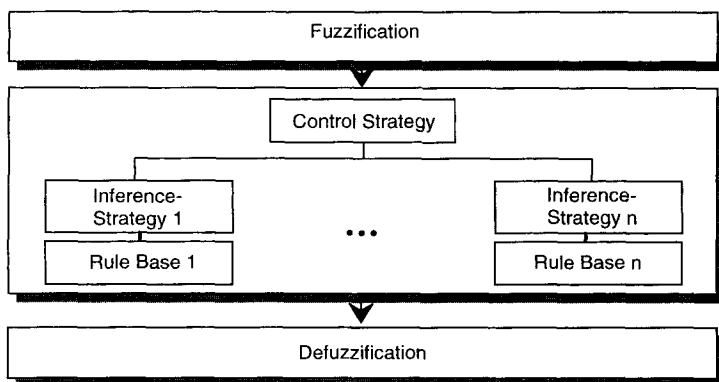


Fig. 3. Structure of the fuzzy controller.

5.1. Modeling of fuzzy data in the global scheduling domain

In a first step the scheduling knowledge has to be classified and reduced to its main features to transform the scheduling data into a format that can be understood by a fuzzy controller. The notion of linguistic variables is used to describe the relevant features of the scheduling data qualitatively. In principle, all data that are used in the model can be transformed into linguistic variables, but in many cases a discrete representation is sufficient. In the global scheduling domain, we describe the scheduling objects by the following features.

- *Resources*: By capacity of machine groups, machine utilization, fuzzy transportation costs and times, consumption of materials and wear and tear of tools.
- *Products*: Product capacity and time consumption for a product.
- *External orders*: Time demand and priority.

To allow for further modeling of distinctions between individual orders, *dates* and *priority of an external order* are represented by linguistic variables as well as *hard* and *soft constraints*. All other features are characterized by crisp values.

The following overview shows some of the modeled scheduling features and denotes the possible values of the linguistic variables. In most cases, triangular or trapezoidal membership functions are used, because they can be implemented efficiently and the linear growth or decline of gradual membership regarding a value of maximum membership can be matched simply. For reasons of clearness we decided to have a maximum of six values for each linguistic variable.

- Capacity of machine groups: {very low, low, normal, high, very high} (see Fig. 4). Since the machines of a group are differently loaded depending on a processed order, a free capacity in percent is given for every time unit that denotes possible additional load.

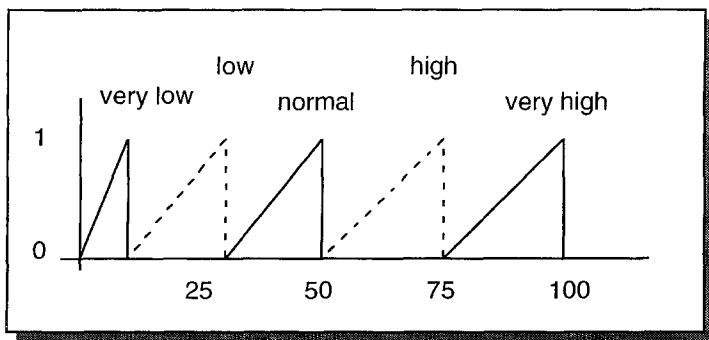


Fig. 4. Membership functions I.

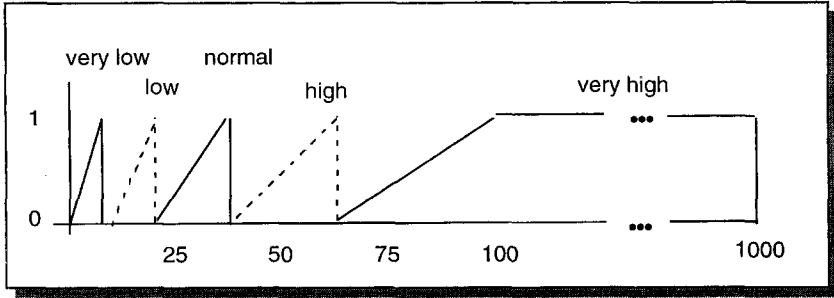


Fig. 5. Membership functions 2.

- Machine utilization: {very low, low, medium, high, very high} (see Fig. 5). Depending on the set of orders a rough estimation of the expected load within a time period can be done.
- Transportation times and costs: approximate values. These values describe the effort that is necessary for the transport from one production site to another. The uncertainty about times and costs are caused by factors that are not known a priori, e.g. uploading time, downloading time, transportation time, traffic situation, failure of the transportation system, energy costs, wages, service costs. Normally, the times and costs vary only within a small region and differ seriously only in exceptional cases. Therefore, an approximate value is used for describing these data.
- Product capacity: {very low, low, medium, high, very high}. Every product or the affiliated intermediate products require specific amounts of time, capacity, and materials on each machine group. The capacity demand of products is described as the capacities of the involved machine groups.
- Time demand: {very low, low, normal, high, very high}. An external order has parameters for the lot size, priority, start time and due date. Sometimes the time demand of the production does not match the chosen time interval. Thus, the time information supports the classification of the importance of an order and the time consumption is formulated relatively to existing orders.
- Priority: {none, low, normal, high, very high}. The (discrete) priority of an external order is transformed into a linguistic value.
- Constraints: {met} (see Fig. 6). The grade of fulfilment of hard and soft constraints is evaluated by linguistic terms. Some hard constraints are treated exceptionally and can be either true or false. The example shows the membership function for the soft constraint “minimal transportation costs” where a 10% exceed is allowable.
- Date: {already, now, soon, later, not yet relevant}. The feature date evaluates the distance of an order from the recent point of time. This is a “dynamic” feature that needs to be newly determined at the beginning of every run.

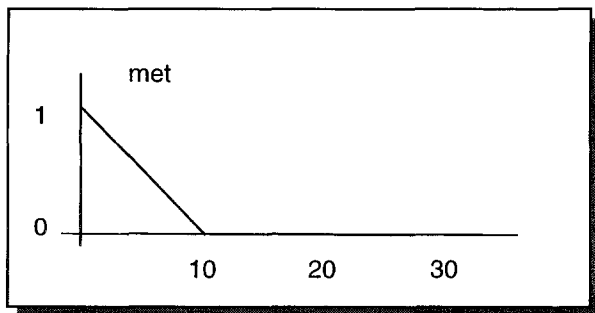


Fig. 6. Membership functions 3.

- Importance: {unimportant, less important, normal, important, very important}. The importance of an order depends on several features, e.g. date, product, time demand. This dependency is described by appropriate rules.

5.2. Processing fuzzy data in global scheduling

To process fuzzy data, a scheduling method must be applied in a fuzzy controller. Depending on specific problem situations and objectives different scheduling strategies may be applicable. To support the flexible use of several fuzzy scheduling strategies a dynamic modeling of fuzzy scheduling knowledge should be possible. Therefore we use a general processing model adopted from the decision making process of human schedulers and previous systems like ISIS [1,6]. It combines heuristic scheduling strategies, problem decomposition and the processing of fuzzy data and rules. The approach uses the following default strategy which is based on an order-based problem decomposition.

The system (like a human expert) starts with a coarse examination of the orders to be scheduled. It first looks for “difficult” orders that should be manufactured first so that the objective, for example meeting the due dates, can be reached. To do this, the orders are categorized with respect to a specific evaluation scheme, e.g. by using the estimated duration of processing an order based on its quantity or the due dates to first select orders that will be acute “soon”. This yields a preliminary order of jobs that is used in the next scheduling step where an appropriate machine group and time window are assigned to each order for an intermediate product. Here rules are applied that for example optimize load balancing or avoid bottleneck resources.

The processing model builds on a two-level hierarchical concept. On the upper level, a control strategy, e.g. the default strategy described above, determines the order of inference strategies with their rule bases and with that the stepwise solution of the scheduling problem. It therefore steers the fuzzy controller.

The individual inference strategies control the processing of their input data by the appropriate rule bases and create output data that are either discrete

(partial) solutions to the scheduling problem or themselves input data for further processing steps. The rules determine the order of processing and the selection of proper parameters (times, machine groups, capacities, variants). The permissible parameters of the rules are features of discrete data that are characterized by their linguistic values and some discrete data. The inference strategy combines the knowledge in the rules with the discrete data. It is independent from the actual knowledge in the rules and therefore any choice of a solution algorithm (in the form of rules) is possible.

As an example, we present the control strategy and the rule-bases for a problem scenario with the objectives of preferring valuable products, optimizing machine utilization, and meeting all due dates. The description of the default control strategy looks like:

```
sequence(IN External_jobs; OUT External_jobs);
/*yields sequence*/
schedule(IN External_jobs; OUT Plan);
/*yields discrete plan*/
material_demand(IN Plan; OUT Disposition);
```

The default control strategy with three steps implies that we have to create three corresponding rule bases for the three subsequent tasks of the fuzzy scheduling process.

The sequence (*sequence*) is generated by rules that characterize the orders with regard to their importance. The more important an order is, the earlier it is scheduled. Within this step all orders are rated by all rules of the belonging rule base and then sorted by the resulting importance. The rule based for the first step looks like:

```
/* rule base to determine the importance of orders*/
IF Time_demand(complete) FUZZY_OR Date(complete)
THEN Importance(important);
/* in the PROLOG implementation this looks like:
fuzzy_or([time_demand(complete), date(complete)],
[importance(important)]) */
IF Product(EP05) FUZZY_OR Product(EP01) FUZZY_OR
Date(complete)
THEN Importance(very important);
```

In the second step the orders are scheduled (*schedule*) according to their position in the sequence by rules reflecting hard and soft constraints as well as other heuristics. For every order a set of possible temporal assignments to machine groups is calculated and rated by the rules of this rule base. The position rated maximum denotes the favourable machine group and is scheduled. Here

the rules look like (soft constraint 8 checks the possibility of scheduling as early as possible and soft constraint 3 tries to optimize load balancing):

```

/* rules to determine the best scheduling position */
IF Machine_Group(MG03) FUZZY_OR Soft_Constraint_8(very
good)
  THEN Importance(very important);
IF Machine_Group(MG03) FUZZY_OR Soft_Constraint_8(good)
  THEN Importance(important);
IF Soft_Constraint_8(medium)
  THEN Importance(normal);
IF Soft_Constraint_8(bad)
  THEN Importance(unimportant);
IF Soft_Constraint_8(very bad)
  THEN Importance(very unimportant);
IF Soft_Constraint_3(met) FUZZY_OR Soft_Constraint_8(met)
  THEN Importance(very important);

```

The demand for material (*material_demand*) is computed in the third step by summation of all products in the form of the given values of the linguistic variables. This yields the disposition that is posed in the model. The rules are:

```

IF Load_on_Date(normal)
  THEN Importance(not very important);
IF DueDate(just_in_time) FUZZY_AND Load_on_Date(normal)
  THEN Importance(normal).

```

This concept permits a segregation of the knowledge bases for each action and with that a directed and independent update of (sub-)strategies or rules by the user. The distributed knowledge management can also be used to enhance the performance of the fuzzy controller, a feature that is important for quick decision making.

5.3. Implementation of the fuzzy controller for global scheduling

The presented approach for global scheduling is implemented by a fuzzy controller that is applicable to several scheduling problems. This is obtained by a largely general architecture in which one can model the specific scheduling problem by describing the fuzzy data and rules using an adequate editor. This supports the representation and use of different scheduling strategies by only changing the fuzzy rule set. New strategies can be realized by changing the following:

- the basic strategy with the belonging inference strategies,
- the rule bases of the inference strategies,

- the variables and their membership functions.

Therefore several other scheduling problems, e.g. job shop or flow shop problems may be tackled as well. The implemented system consists of a graphical user interface including the editor, the fuzzy controller, and the knowledge base (Fig. 7 gives a view of the user interface of the implemented prototype).

The *Editor* window supports the input of the needed rules and parameters similar to the example mentioned above. It allows the update or creation of rules for the rule bases, the generation of the control strategy out of given procedures, and the setting and creation of linguistic values and membership functions. The membership functions and operators are implemented as they are usually defined [10]. From the set of implemented functions the user can choose interactively:

- membership functions, e.g. triangular, trapezoidal, Pi-, S-, or Z-functions as well as discrete value/grade assignments
- fuzzy operators (in the rules)
 - associative operators, e.g. Hamacher product/sum, maximum, minimum, Yager,
 - non-associative operators, e.g. fuzzy-AND, fuzzy-OR, average, geometric mean,
 - comparison operators, e.g. greater, equal, less.

The results of the scheduling process are shown in the windows *Gantt-Chart* and *summary of results* (Zusammenfassung der Ergebnisse).

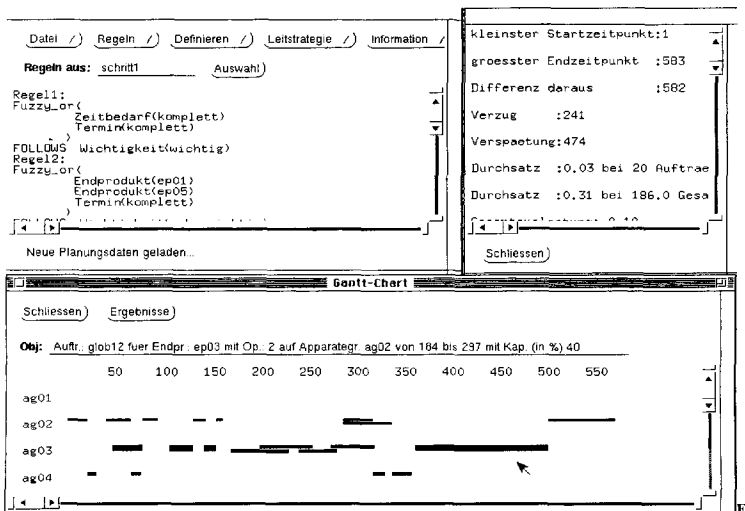


Fig. 7. User interface.

The *fuzzy controller* accesses exclusively the controller data that are generated using the editor and evaluates its rule bases based on actual scheduling data according to the selected control strategy. The result is a global schedule which is presented as a gantt-chart.

The *knowledge base* contains additional information to solve scheduling problems, i.e. data on the scheduling environment, available operators, linguistic variables with corresponding values and membership functions.

The design was realized in *Quintus Prolog* on a *SUN* workstation. Prolog was chosen for the implementation of all tasks because of the easy integration of fuzzy logic in the programming language and the simple handling of necessary data for scheduling and the fuzzy controller.

5.4. Comparison to other algorithms

To allow for comparison of the realized concept and conventional algorithms, some other methods that were chosen considering different criteria were also implemented. Among others we looked at order-based strategies using EDD (earliest due date), SPT (shortest processing time), LPT (longest processing time), and FIFO (first in first out) rules for selecting orders to be scheduled. As sample data we generated different sets of 20 orders that represented different loads in the production, e.g. normal load, large lot sizes per order, and heavy alternation in the size of order sets. We constructed rule bases for each problem that were tailored for the specific load characteristics (RB1–RB3). Table I presents a part of the findings. The number of rules within a rule base is denoted in brackets.

Only the two best results are shown for the simple competing methods. Evaluation criteria are delay and lateness of the orders and the solution time for the scheduling problem. For the design of the conventional algorithms specific attention was paid to the similarity to mechanisms of the fuzzy controller. Therefore, the functions of the fuzzy controller were directly transformed for the scheduling algorithms to facilitate the generation of planning alternatives. Thus, aside from the shortening of fuzzy-logical computations and compari-

Table I
Comparison with other methods

Scenario	Normal load			Large lot sizes			Heavy alternation		
	RB 1 (10)	FIFO	EDD	RB 2 (7)	EDD	SPT	RB 3 (9)	LIFO	SPT
Delay	474	1262	891	39	51	39	1581	2177	1780
Lateness	241	886	561	-627	-649	-627	539	1350	963
Runtime (s)	40	18	15	61	11	16	48	15	22

sons, the algorithms correspond exactly to the functions that the controller is capable of. This guarantees that improvements of the fuzzy controller compared to other scheduling algorithms are only caused by the controller data and the application of the strategies.

For our sample problems it was always possible to conceive a rule base that was equivalent in solution performance to the conventional heuristics. Furthermore, an additional improvement can be achieved by integration of greater number of features and the observation of production- or plant-specific isolated cases, at the cost of a higher running time.

6. Conclusion

The presented approach to global scheduling combines heuristic scheduling strategies and fuzzy concepts. The fuzzy controller implementing the concept can also be used for other scheduling problems by changing the fuzzy knowledge base. It is designed as a component of the multi-site scheduling system MUST which consists of a global component in which the tasks of global scheduling are resolved and different local components at the production sites that take care of the predictive and reactive scheduling actions.

The global level consists of a graphical user interface that offers the user a global leitstand visualizing the global schedule and important information for carrying out global activities. The component for global predictive scheduling has been presented in this paper, the global reactive area is under investigation, e.g. with the use of heuristics or constraint relaxation.

The systems on the lower level basically consist of already known and implemented components like user interface, database, and knowledge-based techniques for predictive and reactive scheduling. Existing components can be integrated, e.g. heuristic [6], genetic algorithms [16], or heuristic reactive methods [17]. Additionally, a component needs to be introduced that manages communication between global and local level. This can be done in different ways. At the moment, a blackboard architecture is implemented based on which further investigations are to follow.

References

- [1] M. Zweben, M.S. Fox, *Intelligent Scheduling*, Morgan Kaufman, Los Altos, CA, 1994.
- [2] S.F. Smith, Knowledge-based production management: Approaches, results and prospects, *Production Planning and Control* 3 (4) (1992).
- [3] R. Kerr, E. Szelke, *Artificial Intelligence in Reactive Scheduling*, Chapman & Hall, London, 1995.
- [4] R. Bruns, J. Sauer, Knowledge-based multi-site coordination and scheduling, in: R.D. Schraft et al. (Eds.), *Flexible Automation and Intelligent Manufacturing 1995*, Begell House, 1995.

- [5] D. Lemmermann, *Globaler Leitstand*, Diploma Thesis, Universität Oldenburg, FB Informatik, 1995.
- [6] J. Sauer, R. Bruns, Knowledge-based scheduling systems in industry and medicine, *IEEE Expert* (1997).
- [7] J. Sauer, Meta-scheduling using dynamic scheduling knowledge, in: J. Dorn, K.A. Froeschl (Eds.), *Scheduling of Production Processes*, Ellis Horwood, Chichester, UK, 1993.
- [8] L.A. Zadeh, Fuzzy sets, *Information and control* 8 (1965).
- [9] L.A. Zadeh, The concept of a linguistic variable and its application to approximate reasoning, *Information Sciences* 8 (1975).
- [10] W. Slany, Scheduling as a fuzzy multiple criteria optimization problem, *Fuzzy Sets and Systems* 78 (1996).
- [11] R.M. Kerr, R.N. Walker, A job-shop scheduling system based on fuzzy arithmetic, *Proceeding of the Third International Conference on Expert Systems and the Leading Edge in Production and Operations Management*, Hilton Head Island, SC, 1989.
- [12] C.S. McCahon, E.S. Lee, Fuzzy job sequencing for a flow job, *European Journal of Operational Research* 62 (1992).
- [13] I.B. Türksen, Fuzzy logic-based expert systems for operations management, in: C.Y. Suen, R. Shinghal (Eds.), *Operational Expert System Applications in Canada*, Pergamon Press, Oxford, 1991.
- [14] J.S. Liu, K.P. Sycara, Distributed Scheduling through Cooperating Specialists, *Proceeding of the IJCAI-93 Workshop on Knowledge-Based Production Planning, Scheduling, and Control*, Chambery, France, 1993.
- [15] B. Kosko, *Neural Networks and Fuzzy Systems – A Dynamical Systems Approach to Machine Intelligence*, Prentice-Hall, Englewood Cliffs, NJ, 1992.
- [16] R. Bruns, Direct chromosome representation and advanced genetic operators for production scheduling, in: S. Forrest (Ed.), *Proceeding of the Fifth International Conference on Genetic Algorithms*, Morgan Kaufmann, San Mateo, CA, 1993.
- [17] H. Henseler, REAKTION: A system for event independent reactive scheduling, in: R. Kerr, E. Szelke (Eds.), *Artificial Intelligence in Reactive Scheduling*, Chapman & Hall, London, 1995.