Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 19 (2014) 180 – 185

www.elsevier.com/locate/procedia

Robust Manufacturing Conference (RoMaC 2014)

Distributed key figure optimization approaches for global goal coordination in multi-agent systems for production control

Florian Pantke[a,*], Otthein Herzog[a]

[a]*Center for Computing and Communication Technologies (TZI), University of Bremen, 28359, Germany*

[*] Corresponding author. Tel.: +49-421-218-64000; fax: +49-421-218-64047. *E-mail address:* fpantke@tzi.de

Abstract

Performance measurement by means of key performance indicators (KPIs) is a widespread device for communicating qualitative and quantitative business objectives throughout an organization and monitoring their achievement. It seems desirable to also apply this concept to dependably influencing the operational behaviour of distributed agent controlled processes, which have progressively gained in importance in production logistics and control over recent years as compared to centralized control methods. However, planning and scheduling with numeric goal systems at present still poses a challenging task within the field of computational intelligence.

In this paper, a framework is presented that enables global coordination of agents in a multi-agent system through user-configurable numeric key performance indicators and associated objectives. A practical approach to distributed control based on periodical mathematical optimization of the defined KPI goal system is described that focuses on how automatic key figure aggregation over changing groups and hierarchies of agents and other business objects can be appropriately managed when proactive modification of these organizational structures, e.g., agents deliberately forming and disbanding groups, is an integral part of the control problem. Five variants of this approach are compared in a case study with a simulated shop floor where incoming manufacturing orders need to be assigned to different machine tools.

Selection and peer-review under responsibility of the International Scientific Committee of “RoMaC 2014” in the person of the Conference Chair Prof. Dr.-Ing. Katja Windt.

Keywords: Multi-agent system, production control, key performance indicators, mathematical optimization, planning and scheduling, dynamic reorganization.

1. Introduction

In production logistics and control, distributed autonomous multi-agent planning has progressively gained in importance over recent years as compared to centralized control methods. This is largely attributable to an increased need for flexibility within the respective processes, e.g., stemming from a continuing trend towards higher degrees of product customization in conjunction with smaller lot counts and preference of just-in-time production and delivery, which classical approaches often cannot satisfy sufficiently [1–3]. Despite their proclaimed autonomy, agent based control approaches still must be able to adhere to various business goals and, hence, should be able to dependably adjust to changing qualitative and quantitative objectives defined by human decision makers. In business organizations, performance measurement by means of numeric key performance indicators (KPIs) and associated objectives is a widespread device for communicating goals throughout the organization and monitoring their achievement [4]. It therefore seems desirable to also apply this concept to influencing the operational behaviour of agent controlled processes [5]. However, planning and scheduling with numeric goal systems at present still poses a challenging task within the field of computational intelligence. This holds for centralized control and even more for distributed agent systems, as the latter require additional mechanisms for proper inter-agent coordination to jointly ensure an acceptable global system state. Especially in cases where dynamic reorganization is an integral part of the control problem to be solved, with agents deliberately forming and disbanding groups and rearranging in certain other ways, distributed control approaches are called for, which explicitly consider future change of the organizational relationships between the agents in their planning and optimization procedures.

In this paper, a framework is presented that enables local and global control and coordination of agents in a multi-agent system through user-configurable key performance indicators. Agents locally assess key figure values from their scope of visibility, e.g., in the context of production data acquisition, and exchange this information to construct a global view on the system. Given a set of user-defined objectives, key figure target values are determined in a distributed mathematical optimization process at run-time and then shared among the agents as

Selection and peer-review under responsibility of the International Scientific Committee of “RoMaC 2014” in the person of the Conference Chair Prof. Dr.-Ing. Katja Windt.
doi:10.1016/j.procir.2014.05.022

input into their local planning processes. A practical approach to this distributed optimization and planning task is described. It focuses on how automatic key figure aggregation over changing groups and hierarchies can be appropriately handled when deliberate and frequent modification of the agent organization is a fundamental aspect of the control and coordination problem at hand. Five variants of this method with different handling of the organizational structure during optimization are compared in a case study with a simulated shop floor where incoming customer orders need to be assigned to different machine tools.

The presented framework, which has been implemented as a Java software library for multi-agent systems, constitutes an important step towards the effective and dependable integration of business KPI systems into autonomous agent controlled processes with highly dynamic inter-agent relationships. In practice, the emergence of unanticipated, chaotic, and sometimes even economically unfavourable behaviour often cannot be entirely ruled out for such processes due to an intended loose coupling of the agents. The KPI agent control framework aims at minimizing this risk by offering the system user a powerful and flexible way of defining goal specifications and performance measures for steering and monitoring the exhibited run-time agent behaviour. While the planning component used in the experiments has been implemented for the examined production control scenario only, the Java KPI library as well as the coordination strategies discussed in this paper support the definition of arbitrary key figures in a multitude of different multi-agent application domains where KPI based control is desired.

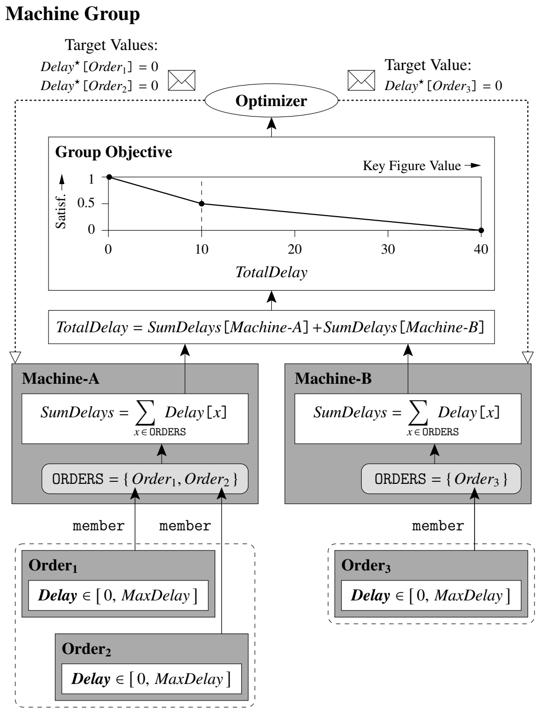

Fig. 1. Coordination by centralized group optimization.

2. Key figure based agent control and coordination

Real-world logistic planning and control problems usually require the consideration of multiple qualitative and quantitative objectives, often with subsets of them mutually conflicting at the local operational or global organizational level (e.g., maximization of machine utilization versus minimization of throughput time in shop floor control problems). The solution of such problems therefore involves the identification of economically suitable trade-offs between these different goals by utilizing techniques from the field of operations research, such as non-linear multi-criteria optimization [6–8]. Because a problem might, in general, have multiple optima, its distribution among several autonomously-acting agents must ensure that all agents jointly aim at the same global system state. Would each agent strive for a different optimum with its local actions, then a globally suboptimal state might be the result.

Fig. 1 illustrates our basic approach to multi-agent coordination by means of a numeric goal system modelled as a computational graph over a hierarchy of production resources, each one represented by an agent. In our framework, each agent periodically captures time-stamped values of a fixed set of measurands from its local scope of visibility, which it can directly influence with its own actions. Based on these atomic inputs, sets of composed key figures, which aggregate the measurands as well as other composed key figures through sequences of basic arithmetic operations and statistic functions, can be customized by the system user at run-time by specifying their textual computation formulae. Automatic aggregation of key figures over resource groups whose sets of members change in the course of time is possible. Objectives for select key figures are declared by means of objective functions that map the respective key figure value to a satisfaction level in the normalized range $[0, 1] \subset \mathbb{R}$. In the current software implementation, the objective functions are specified by the user as piecewise-linear functions (PWLFs), which allow for easy GUI based modelling while simultaneously enforcing clamping to the permitted range in an intuitive way. The entirety of all configured key figure objectives defines a multi-criteria optimization problem in form of a vector-valued function, for which a Pareto optimum is sought. Single points of the Pareto frontier can be determined by feeding a suitable scalarization of the component functions, e.g., their weighted average, into a general mathematical optimizer for maximization [8]. The result is a set of target values for all atomic measurands on which the objective function depends. Because this set consistently identifies a single Pareto optimum of the global goal set (preferably the one closest to the current measurand values according to some suitable distance function), it is distributed among the agents to be used as input into their local action planning processes. As a consequence, given that each planning process is sufficiently successful in actually reaching the target values, the action sequences decentrally planned and executed on their basis by each individual agent then aim at the same global system state.

2.1. Centralized variant

While the framework advocates action planning and execution to always happen in concurrent and decentralized fashion at each individual agent (due to the common intrinsic characteristics of multi-agent systems [9]), target value generation and distribution may either be performed centralized by a single global coordination agent or in a hierarchically distributed top-down approach.

An example of the centralized variant, which represents the original undecomposed problem formulation, is depicted in Fig. 1. It shows a pair of machine tools *Machine-A* and *Machine-B* situated on a shop floor with three manufacturing orders $Order_1$ to $Order_3$ assigned to their two production queues. For each order, the current delay of delivery (e.g., given in unit time slots) is tracked with an atomic measurand by the name of *Delay*, which can directly be influenced (indicated by bold lettering) by the production schedule generated and executed by the respective machine agent. Each of the latter manages a local group `ORDERS` of enqueued unfinished orders, whose membership status changes with each newly enqueued as well as completed order. The production schedule comprises actions such as enqueueing a yet unassigned order, producing or cancelling an enqueued order, performing maintenance, or idling. In the displayed computation formulae, key figures defined at agents further down the organizational hierarchy are referenced by following their name with the name of the specific agent enclosed in square brackets. This way, the user-defined composed key figure *SumDelays* computes the sum of all delays over the local group of orders at each machine. A level further up, the machine group key figure *TotalDelays* adds the delays on both machines to assess the total delay of all orders currently assigned to the machine pair. A single objective function is defined for it, which operationalizes a desired minimization of the total delay. It has a unique global maximum, given by $Delay^\star = 0$ for all three orders.

In this centralized approach, the entire computational graph of the objective function $f(Delay[Order_1], Delay[Order_2], Delay[Order_3])$ is represented, computed, and optimized at a single global coordination agent, which periodically sends its generated target values to the agents further down the hierarchy (e.g., via message passing). The two machine tool agents then in turn use the received target values as numeric goals in their local planning and scheduling processes and thereby jointly attempt to minimize *TotalDelays*.

2.2. Distributed black-box variant

As the computational graph of the organization-wide KPI system may get relatively large in real-world applications, its centralized evaluation and optimization may become infeasible due to increased problem complexity as well as other practical constraints, such as high spatial distribution and given communication bandwidth limitations. For this reason, the framework allows agents to be treated as black-boxes by hiding the computation rules of their composed key figures, i.e., their local computational subgraphs, from other agents further up in the hierarchy. In this distributed approach, with respect to optimization, all key figures whose values are imported from other agents are locally treated as if they were atomic measurands, i.e., they become leaves of the local computational graph representation. Each agent publishes only the names of its locally assessed key figures as well as their current values and possible ranges to the other agents in the system. To this end, the ranges of the composed key figures are determined with interval arithmetic [10,11] from the possible ranges of the leaves of the local graph (i.e., atomic measurands and imported key figures). A distributed optimization process, in which each agent generates

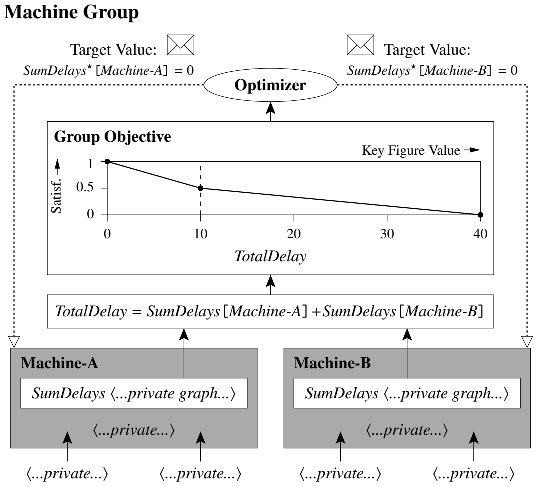

Fig. 2. Coordination by distributed black-box optimization.

target values for the leaves of its local computational graph on a regular basis, progresses top-down through the organization. The target values are sent down to the agent's direct children in the agent hierarchy, where they are integrated as key figure objective PWLFs into the local optimization and planning processes. The resulting target values are, again, sent down to the respective children and so forth. At each agent, the last published possible ranges of the imported key figures are used as constraints to the optimization problem. Fig. 2 shows such a black-box version of the computational graph from Fig. 1.

This modification can significantly reduce the size of the locally managed graphs and make the global optimization problem tractable in the first place. However, it may cause the agents to generate practically unachievable target values for imported key figures due to the hidden dependencies between them. Also, as a result of the dependency problem discussed in the interval arithmetic literature, the occurrence of interval widening during key figure range computation may lead to such behaviour. Consequently, depending on the concrete use case, global system performance in terms of KPI objectives may turn out significantly worse than with centralized optimization.

3. Limitations with respect to dynamic reorganization

Both the centralized and distributed approach to KPI based coordination do not work well in cases where the organizational structure over which the key figures are collected changes over time. Fig. 3 exemplifies this fact with the scenario of two milling machines that assess the average cutting volume (given in cm^3) of the milling jobs assigned to them. Two objectives, aiming at one machine preferring a low and the other a high average cutting volume via suitable order selection, are defined at the global level. Their scalarization has two global optima (*Machine-A* enqueueing orders with cutting volumes close to 1,000 cm^3 and *Machine-B* orders close to 25,000 cm^3, and vice versa) and therefore requires global agent coordination.

Assuming that the milling blank size for each order is fixed once the latter enters the system, the *CuttingVolume* key figure of the orders cannot be influenced by the planned production schedule. Hence, it does not constitute a modifiable atomic input to the optimization problem but a constant (denoted by the equals sign in the respective leaves instead of an interval specification). As a result, the objective function f becomes nullary, i.e., a constant expression, and the optimization problem vanishes. No target values can be generated with the basic approach described above. However, it is apparent that the average cutting volume at each machine can, indeed, be influenced by the production schedule, namely by different order-to-machine assignments—i.e., by dynamic reorganization within the computational graph.

Fig. 4 illustrates how this can be achieved. The ternary logic predicate instance member(*Machine-A*, ORDERS, $Order_2$) represents the fact that the entity $Order_2$ is a member of the group ORDERS of agent *Machine-A* in the current state of the graph. The expression gt(L) yields a graph transformation that modifies the graph in such a way that the given positive or negative literal L holds for the transformation result.

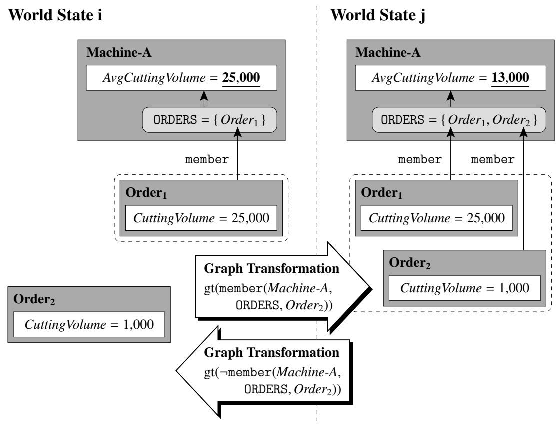

Fig. 4. Influence of dynamic reorganization on aggregated key figure values.

4. Optimizing the organizational structure

For proper, globally coordinated operationalization of the key figure objectives defined in Fig. 3, the possible future graph transformations that are triggered by the agents' executable actions (such as enqueueing, producing, or cancelling a manufacturing order) must be considered not only in the local planning processes but already in the preceding mathematical optimization procedure for target value generation and distribution. To this end, it seems, the optimizer must be able to handle not only a single objective function but many transformed variations of it, finally selecting the ones with the highest maximum value.

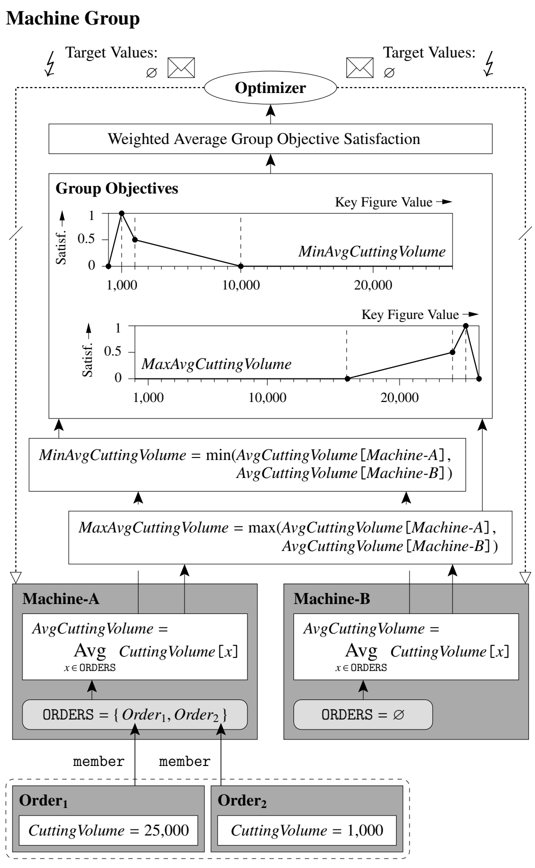

Fig. 3. Optimization problem depending on constant inputs only.

However, there exist standard optimization algorithms like Differential Evolution [7], which are relatively robust towards chaotic, highly multivariate, non-linear objective functions that are not continuously differentiable at all points. Given such a method, optimization of several thousand different functions in succession or even the development of specialized mixed continuous-numeric and graph-optimizing methods usually is not necessary. Instead, the possible computational graph variants can be combined into a single objective function by adding further numeric input variables that control the organizational structure. Fig. 5 shows how such decision nodes are integrated into the computational graph of the first machine, which has been extended with two additional order groups. While ORDERS, as before, holds the unfinished orders enqueued for production, all completed and delivered orders are kept in SOLD_ORDERS, and all orders that were cancelled in the past (e.g., due to machine failures, failed quality tests, or deadline troubles) in CANCELLED_ORDERS. Orders that are newly assigned to the machine by the generated production schedule start off in ORDERS and finally end up either in SOLD_ORDERS or CANCELLED_ORDERS. At all times, no order is contained in more than one distinct group.

In addition to the orders already in the system, a certain number of future incoming jobs $NewOrder_i$ can be added, setting their numeric properties (e.g., cutting volume, contract price, remaining time to deadline etc.) to the empirically expectable interval ranges and using them as atomic inputs into the optimization problem. As the assignment of these placeholder jobs to particular machines is still optional, they do not start as a member of any group. A decision node, behaving like a discrete switch, determines their future final state in the simulation as it will be considered in the optimization process.

Each decision node $Decision_i$, $1 \leq i \leq M$, partitions its normalized range $[0, 1] \subset \mathbb{R}$ into N_i individual subintervals of equal width and, by means of the mapping dec_i, relates its real-numbered input value to one of N_i associated graph transforma-

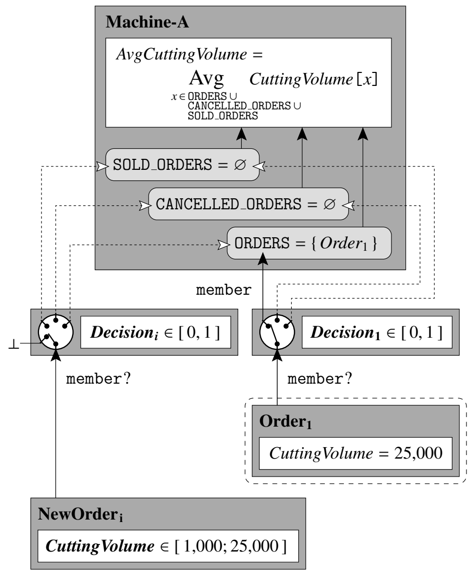

Fig. 5. Optimization of the organizational graph structure with decision nodes.

tions $\mathrm{gt}_{i,k} \in \mathrm{T}_i$, $1 \leq k \leq N_i$, based on the intersected interval:

$$\mathrm{dec}_i \quad : \quad [\,0,\,1\,] \rightarrow \mathrm{T}_i = \{\,\mathrm{gt}_{i,1},\,\ldots,\,\mathrm{gt}_{i,N_i}\,\}$$

$$\mathrm{dec}_i\,(\mathit{Decision}_i) \quad = \quad \begin{cases} \mathrm{gt}_{i,1} & \text{if } \mathit{Decision}_i \in \left[\,0,\ \frac{1}{N_i}\right) \\ \vdots & \\ \mathrm{gt}_{i,N_i-1} & \text{if } \mathit{Decision}_i \in \left[\,\frac{N_i-2}{N_i},\ \frac{N_i-1}{N_i}\right) \\ \mathrm{gt}_{i,N_i} & \text{otherwise.} \end{cases}$$

When the objective function f is evaluated at a certain point $\mathbf{x} = (x_1,\,\ldots,\,x_n\,;\,\mathit{Decision}_1,\,\ldots,\,\mathit{Decision}_M)$, the sequence $(\mathrm{dec}_1(\mathit{Decision}_1),\,\ldots,\,\mathrm{dec}_M(\mathit{Decision}_M))$ of all graph transformations selected by the decision nodes is applied to the computational graph before the latter is traversed for determination of the resulting scalarization function value at $\mathbf{x}$. Tables 1 and 2 display how the decision nodes in Fig. 5 influence the group membership status of the already existing as well as temporarily inserted placeholder orders during function evaluation.

The possible interval range $Y \subset \mathbb{R}$ of f can be computed by evaluating its transformed interval extension $F_{\mathbf{gt}}$ over all combinatorially constructible graph transformation sequences $\mathbf{gt}$ of the decision nodes:

$$Y \quad = \quad \mathrm{interval\_hull}\Bigg(\bigcup_{\mathbf{gt}\,\in\,\mathrm{T}_1\times\cdots\times\mathrm{T}_M} F_{\mathbf{gt}}(\mathbf{X})\Bigg),$$

where $\mathbf{X} = (X_1,\,\ldots,\,X_n)$ denotes the interval vector that specifies the current input ranges of the graph leaves, excluding the decision nodes.

Table 1. Membership of existing Order_1 depending on decision node value.

$\mathrm{Decision}_1$ interval	ORDERS	CANCELLED_ORDERS	SOLD_ORDERS
[0, 1/3)	X		
[1/3, 2/3)		X	
[2/3, 1]			X

Table 2. Membership of $\mathit{NewOrder}_i$ depending on decision node value.

$\mathrm{Decision}_i$ interval	ORDERS	CANCELLED_ORDERS	SOLD_ORDERS
[0, 1/4)			
[1/4, 1/2)			X
[1/2, 3/4)		X	
[3/4, 1]	X		

With the aid of decision nodes, the numeric optimizer is able to determine the best future organizational structure of the multi-agent system within the search space spanned by the inserted decision nodes and placeholder objects and to generate target values for all relevant key figures of the objective function for distribution within the system. The actual action sequences for jointly reaching such an optimal system state are then to be found by the individual agents in their local planning processes. If a composed key figure aggregates over a group whose membership status is influenced by a decision node, then a target value will be distributed for that key figure instead of any of the nodes in its subgraph. Also, no target values are distributed for decision nodes and placeholder objects, as those are only temporarily existent in the local computational graph for the time of the local optimization process and therefore are not visible to other agents.

5. Evaluation

The KPI control framework has been implemented as a Java software library and integrated into the existing IntaPS multi-agent system for production planning and control on simulated shop floors [12]. Using the pair of global key figure objectives shown in Fig. 3, plus one local KPI goal per machine maximizing the monetary value earned with the production and delivery of manufactured orders, six experiments (CD, BD, CS, BS, BSM, and O) with different KPI control modes were conducted in IntaPS. Each experiment involved the two independent agent controlled milling machines *Machine-A* and *Machine-B* and consisted of 50 simulation runs over 20 discrete production time slots. The shop floor's spatial layout was neither modelled nor considered in any way. In each time slot, six new milling jobs randomly selected from two different templates (one with a cutting volume of 1,000 cm^3, the other with 25,000 cm^3) were instantiated for enqueueing or rejection by the two machine tool agents. All orders require a single manufacturing step only. A mixed combinatorial-numeric planner implemented specifically for the IntaPS manufacturing scenario was used by the two agents to solve this scheduling problem. The general structure and inherent difficulties of the latter are presented and discussed in depth in a different article [13]. If key figure target values were available in the respective KPI control mode, they were considered in the planning process.

The experiment CD employed centralized optimization (cf. Sec. 2.1) with decision nodes, and experiment BD the distributed black-box approach (cf. Sec. 2.2), also with decision nodes. In contrast, no decision nodes were used in the KPI optimization processes of experiments CS (centralized optimization) and BS (distributed black-box optimization), leading to the problems concerning dynamic reorganization discussed in Sec. 3. In the additional control experiment O, the two group

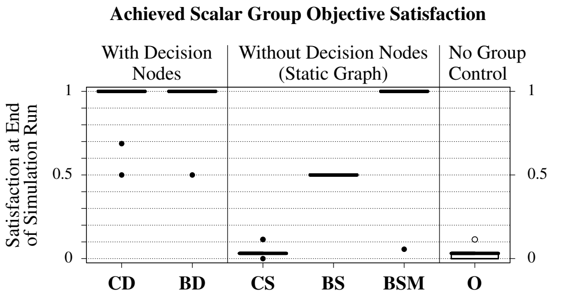

Fig. 6. Statistic of the scalar group objective satisfaction over 50 simulation runs.

objectives regarding the minimum and maximum average cutting volume were removed from the optimization and planning problem, only measuring their satisfaction, i.e., without actual operationalization of these goals.

Fig. 6 shows a box-and-whisker diagram of the scalar group objective satisfaction value reached at the end of each of the 50 randomized simulation runs per experiment. In the plot, almost all boxes visualizing the interquartile range collapse to plain bars, with whiskers being non-existent. This indicates the fact, that far more than 50% of the simulation runs of each experiment, despite each having a different random sequence of incoming orders, ended with exactly the same scalar satisfaction value. Because the two global optima (cf. Sec. 3) are the same in all experiments, the KPI optimization process generated very similar target value sets throughout all runs of each experiment—with different success with respect to optimality, depending on the experiment's KPI control strategy. In the majority of cases, the IntaPS planner was able to actually reach the distributed target values (where available) by generating a suitable production schedule for the randomized order sets.

As can be seen in the plot, the two decision node based KPI coordination strategies (CD, BD) were quite successful in operationalizing the global goal set, whereas centralized optimization without decision nodes (CS) completely failed to generate any target values related to the orders' cutting volume. As a consequence, the machines selected both order types with equal probability, resulting in an average cutting volume around 13,000 cm^3 and a scalar satisfaction value close to 0. The behaviour exhibited by the agents in experiment CS and the control experiment O is therefore the same: The group objectives stay completely unoperationalized.

In the black-box approach without decision nodes (BS), both machines enqueued only orders with a cutting volume of 1,000 cm^3 because the group coordination agent generated too small target values for *AvgCuttingVolume*. This is attributable to the fact that no proper information about the future possible range of that key figure was available to that agent due to the static structure of the computational graph during range computation. Therefore, only one of the two group objectives was reached. This can be fixed by explicitly specifying the correct interval as user input into the system, i.e., as prior domain-specific knowledge. This was done in the BSM variant of experiment BS and resulted in a goal satisfaction similar to the decision node based approaches. However, such semiautomatic control with manual range hinting is usually not an option in larger real-world applications, as it is rather prone to human error and cannot be considered reliable enough with respect to unpredicted future change within the agent environment.

6. Conclusion and outlook

In this paper, a novel framework was presented that enables dependable agent control and coordination in multi-agent systems through user-configurable key performance indicators. The performance of four fully automatic KPI based control approaches and one semiautomatic variant was compared in a minimal production control scenario. Among the fully automatic approaches, the two that integrated possible reorganization of the agents into the objective function and thereby considered future change of the organizational structure in the KPI optimization process achieved the best system-wide goal operationalization. This was accomplished with a standard mathematical optimization module, i.e., without the development of specialized methods for mixed numeric and graph optimization.

The framework, which has been implemented as a Java software library, constitutes an important step towards the effective and robust integration of business KPI systems into autonomous agent controlled business processes with highly dynamic interagent relationships. Outside production control, the library can be used in many different application domains where KPI based control is desired. The extension of our experimental system to process chains across multiple machines and its application and evaluation in large real-world production scenarios, including a still necessary improvement of the domain-specific planning module's performance for this purpose, remains future work.

References

[1] Windt, K., Becker, T., Jeken, O., Gelessus, A.. A classification pattern for autonomous control methods in logistics. Logistics Research 2010;2(2):109–120.

[2] Scholz-Reiter, B., Windt, K., Liu, H.. Modelling dynamic bottlenecks in production networks. Int J Comp Integ M 2011;24(5):391–404.

[3] Schuldt, A.. Multiagent coordination enabling autonomous logistics. KI 2012;26(1):91–94.

[4] Franceschini, F., Galetto, M., Maisano, D.. Management by Measurement – Designing Key Indicators and Performance Measurement Systems. Springer; 2007.

[5] Pantke, F.. Intelligent agent control and coordination with user-configurable key performance indicators. In: Kreowski, H.J., Scholz-Reiter, B., Thoben, K.D., editors. Dynamics in Logistics. Springer; 2011, p. 145–159.

[6] Avriel, M.. Nonlinear Programming: Analysis and Methods. 1st ed.; Englewood Cliffs, New Jersey: Prentice Hall; 1976.

[7] Price, K.V., Storn, R.M., Lampinen, J.A.. Differential Evolution – A Practical Approach to Global Optimization. 1st ed.; Springer; 2005.

[8] Ehrgott, M.. Multicriteria Optimization. 2nd ed.; Springer; 2005.

[9] Müller, H.J.. Towards agent systems engineering. International Journal on Data and Knowledge Engineering, Special Issue on Distributed Expertise 1997;23(3):217–245.

[10] Moore, R.E., Kearfott, R.B., Cloud, M.J.. Introduction to Interval Analysis. Philadelphia: SIAM; 2009.

[11] Brönnimann, H., Melquiond, G., Pion, S.. The design of the Boost interval arithmetic library. Theor Comput Sci 2006;351(1):111–118.

[12] Lorenzen, L., Scholz, T., Timm, I.J., Rudzio, H., Woelk, P., Denkena, B., et al. Integrated process planning and production control. In: Kirn, S., Herzog, O., Lockemann, P., Spaniol, O., editors. Multiagent Engineering – Theory and Application in Enterprises. Springer; 2006, p. 91–114.

[13] Pantke, F., Edelkamp, S., Herzog, O.. Combinatorial planning with numerical parameter optimization for local control in multi-agent systems. In: 2nd International Conference on System-Integrated Intelligence (SysInt 2014). Procedia Technology; Elsevier; 2014, Accepted for publication.