Coordination of Distributed Control Systems

Andreas Kroll, Hanns Sommer
Universität Kassel, Fachgebiet Mess- und Regelungstechnik
D-34109 Kassel, Mönchebergstr. 7
andreas.kroll@mrt.uni-kassel.de
hanns.sommer@mrt.uni-kassel.de

Abstract A general heuristic methodology to specify coordination algorithms for distributed systems is presented. The methodology is summarised in six fuzzy rules. To translate these rules into effective tractable algorithms, two methods will be examined: via a local coordination and, in terms of Max Weber: an 'ideal behaviour'. We demonstrate that thereby principally different types of systems are obtained. Often, coordination can only be achieved by the second method and the behaviour of those systems shows phenomena that are impossible in locally coordinated systems: downward causation and strong anticipation.
Keywords distributed control systems, local information, global information, coordination, anticipatory systems

1 Introduction

Distributed control means the control of a plant by many agents under the restriction that none of them disposes of the whole system information. There exist many reasons that may be responsible for the restricted access of the agents to the complete system information: memory restrictions, long transmission times or exponentially increasing computation times. Examples of distributed control systems are omnipresent in our technical world: resource allocation systems, energy distribution systems, cell structures, animal swarms, systems of cellular automata, traffic systems and large production facilities.

In distributed control systems, it can be distinguished between two types of information and the appropriate languages. Local information and a local control language corresponds to all events in an agent's neighbourhood, his actions and their direct consequences, whereas global information is related to the coordination of the whole system and cannot take into account the special situations of single agents. We shall demonstrate that there does not exist a precise translation between these two languages in general. (A result that corresponds to Quine's untranslatability thesis [1]). The practical importance of this fact will be demonstrated by examples.

In the centre of our presentation is a translation method between the local and the global languages that is based on Max Weber's idea of 'ideal types'. We examine the advantages and restrictions of this method in the framework of a general discussion of the principal properties of various architectures of distributed control systems.

International Journal of Computing Anticipatory Systems, Volume 30, 2014
Edited by D. M. Dubois, CHAOS, Liège, Belgium, ISSN 1373-5411 ISBN 2-930396-19-9

Experimental results will be obtained from a simulation of a supervision system which is constituted by a set of flexible robots. Even though, this system is relatively simple it is adequate to highlight the capabilities of different architectures.

We think that our discussion contributes to Daniel M. Dubois' theory of weak and strong anticipation [2]. As weak anticipation corresponds to a local language and strong anticipation to a global language, the qualitatively different behaviours of systems with local or global constraints, that are demonstrated in our theoretical results and computer experiments, explain also differences between weak and strong anticipatory systems.

2 Formalisms for Describing Distributed Control Systems

In the following we present a general formalism for the description of distributed control systems:

Definition (Distributed control): A distributed control system is defined by a set of equations:

$$\vec{X}_k(t+1) = \vec{f}_k(\vec{X}_k(t), \vec{U}_k(t)), \text{ with } \vec{X}_k \in \mathcal{R}^n, \vec{U}_k \in \mathcal{R}^p, k = 1, \ldots, K. \quad (1)$$

and a set of constraints:

$$\vec{g}_k(\vec{X}(t)) = \vec{h}_k(t) \text{ with } \vec{X} = (\vec{X}_1, \ldots, \vec{X}_K)^T, \vec{g}_k, \vec{h}_k \in \mathcal{R}^q, k = 1, \ldots, K. \quad (2)$$

Let $Var(f)$ denote the set of variables of a function f and $\mathcal{X}_k$ the space where $\vec{X}_k$ is defined. In the sequel we use the notions:

- A constraint is called strongly local iff $Var(\vec{g}_k) \subseteq \mathcal{X}_k$.
- A local constraint is one in which the number of equations K is not a parameter, or in other words:
  There is a fixed integer $M << K$ such that $Var(\vec{g}_k) \subseteq \bigcup_{i \in M_k} \mathcal{X}_i$ for a subset M_k of $\{1, \ldots, K\}$ with less than M elements.
- A global constraint is one which satisfies the equation:

  $$\vec{g}_k((\vec{X}_1, \ldots, \vec{X}_K)^T) = \vec{g}_k((\vec{X}_{\sigma(1)}, \ldots, \vec{X}_{\sigma(K)})^T$$

  for any bijective mapping $\sigma : \{1, \ldots, K\} \to \{1, \ldots, K\}$.
- The dependence graph of a distributed control system is defined by the tuple $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V} = \{1, \ldots, K\}$ denotes the set of nodes and
  $\mathcal{E} = \{(k, l) \in \mathcal{V} \times \mathcal{V} \mid$ there is a local constraint $\vec{g}_l(\vec{X}(t)) = \vec{h}_l(t)$ such that: $\mathcal{X}_k \cap Var(\vec{g}_l) \neq 0\}$ specifies the set of edges.

Many seemingly very different configurations can be described by the given formalism. Equation (1) describes the influence an agent has in subsystem k of a plant and equation (2) the exigencies he has to maintain by an adequate selection of the control function U_k. The main problem in distributed control consists in an adequate coordination between the various agents. Normally there exists no efficient, tractable algorithm to accomplish this coordination task and it must be searched for a best feasible approximation. In the language of the local agents it is often even not possible to formulate global constraints exactly but only in an imprecise translation. Global information may be so complex that it cannot be stored in the agent's memory and it is possible that global ideas are not available in the agents' language. The minimal storage size in which an information can be stored defines its Kolmogoroff-complexity.

To design the overall control of a distributed system, we have to specify coordination rules whose compliance by the individual agents circumvents strong mutual disturbances. The following methods will be considered:

Definition (Coordination methods for distributed systems):

I **The omnipotent supervisor:** All system information is collected by an omnipotent supervisor which calculates the optimal action for each agent in every time instant and commands the actors to execute these actions.

II **Local coordination:** Agents are informed by their neighbours and take this information into consideration when they select their actions. Most efficiently the topology of the agent set is specified with respect to the dependency graph. The neighbourhood N_i of agent i is defined: $N_i := \{j \mid (i,j) \in \mathcal{E} \text{ or } (j,i) \in \mathcal{E}\}$.

III **Max Weber's ideal behaviour:** The agents create a common knowledge which helps them to coordinate their actions. It may be possible that this knowledge contains globally formulated ideas (global constraints) that cannot be translated exactly in the individual agent's language.

To coordinate the actions of the agents with method (III), there exist typical mathematical expressions that had been used by other authors:

III.1 Mean values $\omega := \frac{1}{K}\sum_{k=1}^{K}\omega_k$, where ω_k is calculated by agent k.

III.2 Relatedness values $z := \sum_{k,j=1}^{K} c_{k,i}x_{k,i}$, where $x_{k,j}$ describes the relatedness between agent k and agent j.

III.3 The potential $f(\vec{X}) := \sum_{k=1}^{K}\exp(\frac{-\|\vec{X}-\vec{X}_k(t)\|}{\Sigma})$ or $f(\vec{X}) := \sum_{k=1}^{K}\frac{1}{-\|\vec{X}-\vec{X}_k(t)\|}$. This potential is often used by agents that have to maximise or minimise their distance to the others, but it is normally not available because its Kolmogoroff-complexity depends on K.

Methods (II) and (III) can be realised in very different ways and it is also possible to combine these methods. In the next section, we present a theory based design approach to realise these methods for various given distributed control systems .

3 A Coordination Logic for Distributed Control Systems

The problem of coordination is ubiquitous in distributed systems and one of the main challenges of modern engineering. Today it is no longer possible for engineers to find technical solutions simply by collecting ideas, evaluating these ideas and selecting the best for the final approach. Such a strategy fails, because it is impossible to evaluate ideas by themselves without taking their interaction with the final overall approach into consideration. Engineers are never sure that they have found the most important ideas and they cannot estimate the mutual influences which different methods cause against each other.

There is an important lesson, engineers have learned from software engineering:

- The design of a complex system calls for a methodology which enables a controlled, complete and evaluable creation of solution ideas. These ideas should be coordinated from the very beginning by basic concepts.

In the following, we present a methodology by which the basic concepts for the coordination of distributed systems can be found. Philip Anderson's statement (1972) 'More is different!' tells us that large, complex systems cannot be understood as a collection of simple parts. The most important effect that distinguishes conventional systems from complex systems is caused by couplings that may emerge in complex systems over large parts of the whole configuration. An examination of disasters in engineering systems (i.e. the catastrophe of Tschernobyl) demonstrates that it is almost impossible to recognise the emergence of all possible couplings in advance. The idea we trace here to cope with this problem consists in a search for new additional constraints, aside from those of equation (2), which guarantee a disturbance free coordination between the agents under minimal restrictions of their individual freedom of activity. As normally these additional constraints cannot be formulated in local terms and in the language of the individual agents, to realise them, method (II) and (III) of section (2) will be used.

3.1 A search for basic principles to avoid disturbances caused by coupling effects

As it is impossible to find something that is incognisable, we need a plan to make additional coordinating constraints visible. This objective will be achieved by the following strategy:

A: Present general coupling effects in the most simple description format.

B: Find all simple coordination methods that are possible in this descriptions.

C: Translate this methods into general rules for disturbance avoidance in distributed systems.

A: Characterisation of coupling and decoupling effects

To specify unambiguously the meaning of coupling effects, we postulate the

Thesis: Instances of constraint satisfaction problems constitute the basic patterns of decoupling problems.
Coupling and decoupling are exemplarily characterised by instances of constraint satisfaction problems (CSP) and by value assignments (solutions) that satisfy these coupling constraints.

Definition (Constraint satisfaction problem (CSP) [3]): A CSP is defined by a set of variables $\mathcal{X} = \{x_1, \ldots, x_n\}$ and a set of constraints $\mathcal{C} = (C_1, \ldots, C_m)$. Each variable x_i has a non-empty domain D_i of possible values. Each constraint C_j involves some subset of variables - the scope of the constraint - and specifies the allowable combinations of values for that subset. An assignment that does not violate an constraint is called consistent (or a solution).

Let $\xi \in \{\sqcup, \neg\}$ denote affirmation or negation. CSP-problems are called SAT-problems if $D_i = \{0,1\}$ for all i, and the constraints are given in normal form $C_j = (\xi_{j1}x_{j1}, \ldots, \xi_{jK}x_{jK})$. An assignment $a : \{x_1, \ldots, x_n\} \to \{0,1\}$ is called consistent with the constraint C_j if at least for one $k \in \{j_1, \ldots, j_K\}$ the condition: $((a(k) = 0$ and $\xi_{jk} = \sqcup)$ or $(a(k) = 1$ and $\xi_{jk} = \neg))$ holds.

(It can be demonstrated, that each constraint on variables x_i with values in $\{0,1\}$ is representable with constraints in normal form.)

A clause C_j can be interpreted as a demand, an agent j has to satisfy. Today it is unknown if SAT-problems are solvable in time intervals whose length depends polynomial on $\sum_{j=1}^{m} \mid C_j \mid$, where $\mid C_j \mid$ denotes the number of variables of constraint C_j. On the other side, a theorem of Schaefer gives us an overview of the methodologies by which SAT-problems are solvable:

Schaefer's Theorem [4]: There are only three polynomial solvable constraint-satisfaction problems on the set $\{0,1\}$, namely,

(1,a) 0-valid problems (problems where all-zeros is always a solution, and similarly 1-valid problems).

(1,b) Horn clauses (problems where every relation in the template can be characterised by a conjunction of clauses with at most one positive literal per clause, and similarly anti-Horn clauses, with at most one negative literal per clause).

(2) 2-SAT (problems where every relation in the template can be characterised by a conjunction of clauses with two literals per clause).

(3) Linear equations modulo 2 (problems where every relation in the template is the solution set of a system of linear equations modulo 2).

All constraint-satisfaction problems on $\{0, 1\}$ that are not in one of these classes are NP-complete.

An interpretation of Schaefer's Theorem in common language provides the following

list of general principles to find consistent assignments to CSP-problems:

I **Irrevocable assignments:** Irrevocable assignments are always optimal.

II **Priority order of assignments:** In a chain of dependent assignments, the first member of the chain (that is unaffected by the others) receives the highest priority. The order priorities then follow the dependencies in the chain.

III **Action decoupling:** Find independent basic actions and construct complex actions by concatenating these basic actions.

B: Heuristics for coordination

From the basic principles of part [A] heuristics are obtained, if we reformulate them in a Fuzzy-language. A fuzzification of a statement is produced replacing:
' all x' by 'most x' and 'no x' by 'only a few x'.
The exact meaning of 'most' and 'a few' can be defined with a Fuzzy logic.

Heuristics for the agents in a distributed system:

(1) Select those actions which are most helpful also for the other agents.
(i.e. for SAT-problems: To satisfy clause C_j select the assignment, which makes C_j true and besides that as many other clauses as possible.)

(2) Define the priority in the set of agents such that agent k has higher priority than agent j if agent k has less freedom to satisfy his demands than agent j.
(i.e. for SAT-problems: Fix the values of small clauses first before the assignment is done for larger clauses.)

(3) Select actions with strong effects and small side effects (Minimise the production of trash!).
(i.e. for SAT-problems: satisfy clause C_j by an assignment which restricts the freedoms of the other agents only minimally.)

C: Rules for mutual disturbance avoidance and coordination

The heuristics do not provide a non-ambiguous command for each agent. E.g. it is not clear what is better in heuristic (1):
(to satisfy a maximal number of other clauses) or (to satisfy as many of the shortest clauses as possible).
To make this heuristics more precise, the ideas in their formulation must be rendered more exactly. For this reason, we examine the possibilities to recognise them in general contexts. Recognition means the detection of the essential. In this way, the ideas in the heuristics must be translated into a language which facilitates its exploration.

Charles Sanders Peirce distinguished between three methods of codification: Codification by means of an icon, an index or a symbol.

- A symbol codifies an object by means of an agreement that creates a relation between object and symbol.
- By an index we mean a pointer to the object.
- The relation between an icon and an object, on the other side, is constituted by properties which are common to both.

In this way, only icons correspond directly to recognitions. Peirce wrote: 'All deductive reasoning... involves an element of observation, namely deduction consists in constructing an icon or diagram the relations of whose parts shall present a complete analogy with those of the parts of the object of reasoning, of experimenting upon this image in the imagination, and of observing the result so as to discover unnoticed and hidden relations among the parts.'

Disturbance avoiding constraints must therefore be recognised from controlling distributed systems in the same way as icons are recognised in a complex world. They correspond to simple ideas which effect disturbance avoidance in many distributed systems. This ideas will be realised by the following

Approach to find coordination rules:

(1) Use the same language in which the systems are given for a characterisation of their properties.
(I.e. for SAT-instances: The number of clauses in an instance, the number of variables and relations between small and large clauses provide terms that characterise a special SAT-instance.)
A list of all those terms whose Kolmogoroff-complexity is less than a fixed value ρ can be obtained and this terms can be used to form sentences with less than ρ symbols that represent properties of SAT-instances.

(2) Select from the set of sentences that had been constructed in (1) those that: characterise easily solvable problems (set A) or hard and insolvable problems (set B).

(3) Using the selected sentences, coordination rules are formed upon the pattern: (Rule:) Select those actions by which the system is changed in a way that sentences in A will hold and sentences in B are negated.

By this approach the heuristics can be adapted to special CSPs or to the examined controlling distributed systems. The approach provides different types of rules. An examination of general constraint satisfaction problems resulted in the following list:

Rule I (Avoidance of trash): Each agent should select those actions that will not produce causes for undesired events (trash).

The efficiency of rule (I) depends on the possibility to specify the meaning of 'trash' or the meaning of 'a cause of an undesired event'. Halpern and Hitchcock wrote with respect to that question [5]: 'The basic structural equations model (a specification of models by equation like equation (1)) does not seem to suffice to completely capture all aspects of causal reasoning.'

If two agents produce an effect by their joint action, it is not clear which of them should be considered to be the causer. Halpern and Hitchcock recommend therefore a definition of causation that had been analysed by Halpern and Pearl (2005):

Definition of causation based on conterfactual dependence: 'A is a cause of B if, had A not happened then B would not have happened.'

To enable more often the trace back to a causal reason, they specify more precisely:

'When showing that if A hadn't happened then B would not have happened, we consider only contingencies that are at least as normal as the actual world.'

Adapting this definition to our discussion, we obtain the

Definition (responsibility of an agent): An agent is responsible of an effect (he is the causer), if that effect is produced by his action under the assumption that all other agents' behaviour is normal and corresponds to their usual actions.

If an agent knows the prevailing behaviour of the others, he will be able to comply with rule (I). The discussion of rule (I) demonstrates us that trash avoidance is very different from coordination, even if the utilisation of trash involves some form of coordination. To enable coordination, it is necessary to select the coordinating agents:

Rule II (Coordination): Coordination implies two steps:

- **Formation of groups of coordinating agents:** Select a significant sentence of the set A or set B that had been found in step (2) of the approach and

establish the rule:
IF agent k's actions influence the selected sentence then k is part of the group.

- **Coordination:** The agents of the group inform each other which actions they plan to execute and adapt their decisions to those of the others.

To identify situations in which coordination will be useful, the agents remember former situations and sentences which provided high advancements. Values that characterise these situations will be stored in a 'harmony memory' [6]. Whenever a similar situation is reached to one stored in the harmony memory, the corresponding sentence will be reused to coordinate the agents again.

The general application of the sentences of set A and set B is formulated in rule (III):

Rule III (Good conduct): Each agent should select those actions that increase the trueness of sentences in set A and decrease the trueness of sentences in set B.

E.g. in surveillance tasks it is often observed that the agents act very inefficiently, if they all stay together. According to rule (III) this observation will be transformed into the devise: 'Move in a direction that will bring you as far as possible from the other agents!'.

Heuristic (2) provides a priority assignment rule for the agents:

Rule IV (Ranking): If agent k has less freedom than agent j then agent k has to choose his action first. Agent j chooses then his action, taking the result of agent k's action into consideration.

Our next rule is obtained by a translation of the basic idea of the well known PLL-algorithm to solve SAT-problems into a general principle:

Rule V (Tabu search): If a first implementation of the principles obtained from the rules (I)-(IV) does not give effective results, change the implementation until an effective one is reached. Remember old implementations to avoid circular digging around.

The implementation of the rules by algorithms must be adapted to the properties of special problems. As in large problem sets, divers properties are encountered, solvers for these problem sets use algorithm portfolios. The design methods that had been proposed by M. Nikolic, F. Maric and P. Janicic [7] will be translated into a portfolio design rule for coordination problems via rule (VI):

Rule VI (Algorithm Portfolio generation): A solver for a class of coordination problems will be designed by the approach:

(1) Select a finite set of problems from the class that are as different as possible.

(2) Construct solution-algorithms for the selected problems using the rules (I)-(V).

(3) For each algorithm and all selected problems, a penalty will be calculated that corresponds to the infeasibilities of the algorithm respective to this special problem.

(4) For a new problem, its nearest neighbours from the set of selected problems (with respect to some distance measure) are found and the algorithm with the minimal penalty for those problems is involved.

3.2 Realisation of coordination rules using an 'ideal behaviour'

The formulation of the coordination rules (I)-(V) depends in part on ideas or terms that are not accessible in the agents logic or cannot be evaluated respective to their knowledge only. Structures are needed that allow the agents a complete evaluation of the state of the overall system. The extension of the agent's knowledge has to be performed in such a way that its effective (tractable) evaluation remains possible. This means that only the coordination methods (II) and (III) can be taken into consideration. Method (II) is realised by rule (II). Method (III) on the other side, offers a world of new possibilities.

The information stored in an 'ideal behaviour' should comprise all data that are needed to satisfy the rules (I-V). But as this information may benefit some agents more than others, it is also necessary to constitute it in a democratic manner. To keep the evaluation tractable, the Kolmogoroff complexity of this information must be independent of global parameters (e.g. K).

The process by which the consensus between the agents will be produced is usually denominated 'preference elicitation' [8]. The overall structure of a controlling distributed system whose coordination is achieved by an ideal behaviour, is sketched in Figure 1.

Specification of a controlling distributed system whose coordination is achieved by an ideal behaviour:

(I) **Behaviour of the agents:** The agents produce the ideal behaviour by means of a preference elicitation procedure. To select their actions, they use the common knowledge and comply (as far as possible) with the additional constraints that are imposed by the coordination rules.

(II) **Specification of the preference elicitation procedure (PEP):** To produce an 'ideal behaviour' the preference elicitation procedure has to achieve the following tasks:

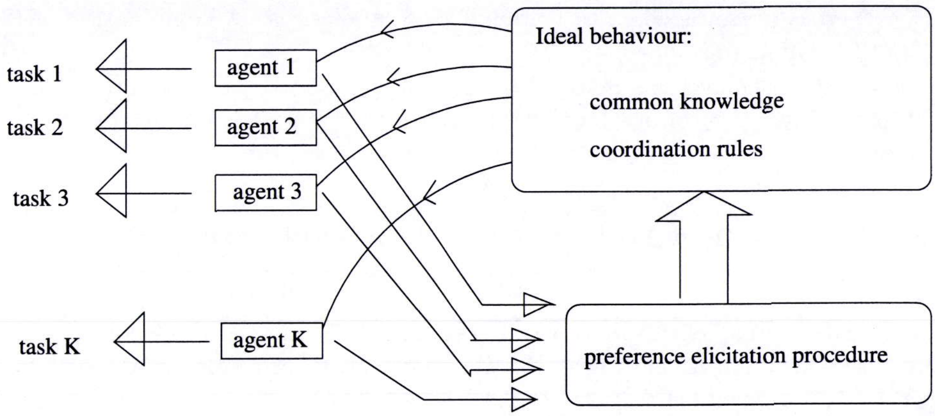

Fig. 1: Coordination of a distributed control system by an 'ideal behaviour'.

- Coordination of the agent's influence,
- creation of rules,
- preparation of a common knowledge.

The optimality of a PEP depends crucially on its adaptation to the real needs of the overall system. E.g. in a plant, no part is allowed to fail, what means that the 'ideal behaviour' has to guarantee that all agents achieve satisfactorily their objectives. On the other side, a group of mountain climbers has the objective to put their banner on the top of a mountain. To reach their objective, it will be enough to bring one agent with the banner to the top and all other agents should act in favour of this single agent.

There exist several guidances to produce an ideal behaviour:

(1) Some features are instructive in general. They are needed by all agents to cope with rules of type I and III. To represent them as exactly as possible in the common knowledge benefits all agents.

(2) Each agent depends on an environment that is created by the collectivity of agents. His desire, to organise this environment for him as favourable as possible, produces behavioural instructions for the others. As each agent wishes to optimise an environment for himself, the PEP should find a balance between the agents. Rule IV provides the directive:
'Agent k's influence on the specification of the behavioural instructions should be inversely proportional to his freedom of action.'

(3) The most successful agents may serve as guides for the others. The behaviour of those agents should therefore be memorised in the common knowledge to guide the others.

Often it is advantageous, to combine these guidances. A combination of (2) and (3) entails:
'If agent k achieves his goals than his influence on the behavioural instructions is inversely proportional to his freedom of action, else agent k will be disregarded.

4 A Discussion of Different Coordination Strategies

In the previous sections we have presented the methods by which distributed systems can be coordinated and we have summarised the principal ideas to realise this coordination formulating the rules (I)-(V). To design the control for a distributed system, we recommend the following approach:

(I) Find problems that complicate the control task and identify situations that make this task hard.

(II) Select those rules from our list by which these problems and critical situations can be avoided and translate them in the same language that was used in the specification of the actual system.

(III) Realise the rules with the coordination methods that were presented in section 1.

As method (I) very often does not provide a tractable algorithm, it is normally only necessary to decide between method (II), method (III) and a combination of both methods. To facilitate this decision, the following questions will be discussed:

Question (1): Do controlling distributed systems exist, for which a tractable control can be designed with method (III) but not with method (II)?

Answer: Yes, the density classification task cannot be generally solved by method (II):
Density classification task (DCT): Given an arbitrary initial configuration of a n-dimensional two-state cellular automata. The cellular automata should converge to a state of all 1's if the initial configuration contains a density of 1's $\geq \rho$ and to all 0's otherwise, for some ρ between 0 and 1.

Mark Lund and Richard Belew demonstrated (1995) that no cellular automata is able to solve the DCT in general [9]. As each cell can be regarded as an agent whose neighbourhood consists of all agents (cells) which are connected to it, this result demonstrates that DCTs cannot be solved with coordination method (II) only. On the other side, calculating the mean value $\bar{s} = \frac{1}{K}\sum_{k=1}^{K} s_k$ where s_k represents the state of cell k, and storing $\bar{s}$ accessible for all agents in a common memory, enables to each agent a simple strategy:

go to state value 0 if $\bar{s} < \rho$ and to state vale 1 if $\bar{s} \geq \rho$. Our answer to question (1) leads directly to question (2):

Question (2): Are all cooperation strategies realisable with method (II) and method (III)?

Answer: No, the variety of controlling distributed systems is too rich, to be coordinated by these methods only.

Assuming $P \neq NP$ our answer would be a direct consequence of the tractability of coordination strategies obtained from methods (II) and (III). But besides this argument, we think it would be interesting, to present also an argumentation which follows the reasoning that had been used to answer question (1).

By method (III) a new knowledge-structure is specified for the agents and respective to this new structure a DCT can be reconsidered for other parameters. The principle that causes the unresolvability of the DCT will be rendered more precisely with the following considerations:
An unresolvable DCT makes the demands on the agents:

To steer one of its parameters to a value that depends on the overall system under the condition that each individual agent cannot obtain enough information from an examination of his own neighbourhood to select an adequate action. More precisely: the information which informs the agents how to select their actions is distributed over the set of all agents in the following way:

- The majority of agents receives a weak signal, to steer into the wrong direction, whereas only a minority of the agents receives a strong signal for the right direction.

- The weighted mean value of all signals would provide the right direction, but as this value is not accessible to the majority of the agents, this majority steers to the wrong side.

As the agents dispose only of a restricted set of actions, it is not possible for them, to adapt their action exactly to the signals they receive from their individual neighbourhood and thus to maintain the mean values of all signals constant. By their actions, this mean value will either increase or decrease and thus, from an adequately selected initial condition, move into the wrong direction into a new state that corresponds to the opposite steering task.

It had been demonstrated by Lund's and Belew's example that such an unresolvable DCT can always be defined for distributed systems with global parameters that are inaccessible for the local agents [9].

Checking SAT-instances, it can easily be recognised that the distribution of the length of clauses or the correlations between the portions of common parameters in different clauses etc. provide an unbounded number of independent global parameters in the set of all SAT-instances. Each independent global parameter can be used to define a new DCT. To facilitate the agents, to solve all DCTs that can be defined with the global parameters, that are available in a general distributed system, a common memory would be needed whose size is unbounded or dependent on the number of all agents K.

Accordingly, the common knowledge used in method (III) has to be adopted to the special demands on the system and cannot be designed in advance for arbitrary exigencies.

The principal problem in distributed systems consists in the restriction of the agent's language. They dispose only of ideas that can be defined by the local information and by the common knowledge. But not all information which the agents need for their coordination can be formulated in this way. E.g. the demand that all agents should be uniformly distributed. Our considerations provide the result:

'Not all ideas that can be specified in the overall system are available for the individual agents, even if they have access to a common knowledge.'

Question (3): Is it possible to characterise some controlling distributed systems, for which control strategies can be realised by effective tractable algorithms?

Answer: Yes, problem decomposition and ranking (which are the basic ideas of heuristics (2) and (3)) provide an effective algorithm for controlling distributed systems that satisfy the **non-overlapping-convexity property (NOCP):** [10]
A system defined by the equations (1) and (2) satisfies the NOCP if there exist subsets of the set of variables $\{x_1, \ldots, x_n\}$: $\bar{\mathcal{X}}_1, \bar{\mathcal{X}}_2, \ldots, \bar{\mathcal{X}}_r$ and functions $f_{k,i}(\bar{\mathcal{X}}_i)$ such that all constraint functions $\vec{g}_k, k = 1, \ldots, K$ can be represented as a sum:

$$\vec{g}_k = \sum_{i=1}^{r} f_{k,i}(\bar{\mathcal{X}}_i) \tag{3}$$

Assuming this property, the value assignment to the variables can be organised by the **algorithm:**

- From equation (3) calculate the parameters

$$f_{k,i} = f_{k,i}(\bar{\mathcal{X}}_i) \tag{4}$$

- DEFINE $\bar{\mathcal{X}} := \{\bar{\mathcal{X}}_1, \ldots, \bar{\mathcal{X}}_r\}$.

- REPEAT UNTIL $\bar{\mathcal{X}} == \emptyset$:
  Find $\bar{\mathcal{X}}_i \in \bar{\mathcal{X}}$ for which no proper subset exists in $\bar{\mathcal{X}}$.
  Assign the values to the variables of the set $\bar{\mathcal{X}}_i$ such that equation (4) will be satisfied.
  $\bar{\mathcal{X}} = \bar{\mathcal{X}} - \{\bar{\mathcal{X}}_i\}$.
  END REPEAT.

Question (4): Is it possible to characterise the principal difficulties that frustrate the controllability of distributed systems?

Answer: A characterisation of hard coordination problems can be found by the following consideration:

Coupling problems had been extensively studied by physicists to understand condensed matter systems. A structural change of a condensed matter system is called 'phase transition'. Leo P. Kadanoff's 'extended singularity theorem characterises the conditions under which phase transitions occur.

'Phase transitions only occur, when the condensed matter system [which corresponds to the plant in our considerations] exhibits the effect of some singularity extended over the entire spatial extend of the system. The infinity arises because of some effect is propagated over the entire condensed system that is, over a potentially unbounded distance.'

Kadanoff's theorem demonstrates that critical couplings that impede a control of the system by local methods (methods (II) and (III)) are produced by the emergence of effects that spread over the whole system and depend on correlations that are extended over large regions of the system. C. Park, R.M. Worth and L.L. Rubchinsky had observed this effect in the dynamics in Parkinson-brain [11]:

'Brain activity in Parkinson's disease is marked by excessive synchrony of neural oscillations in the beta (β) frequency band, which not only accompanies the motor symptoms, but is likely to cause them.... The symptoms of Parkinson's disease ultimately result from the death of dopaminergic neurons in a subcortical brain structure called the basal ganglia. For many basal ganglia synapses this will be translated into an increase of synaptic connections because dopamine tends to suppress them.'
As a result of this consideration, we obtain the following answer to question (4):

Synchronisation effects that spread over large regions of the system impede its control by locally distributed agents. The recognition of these synchronisation effects is essential for the effectiveness of the control of the whole system [12]. Normally it remains only one choice:
'If synchronisation effects are recognised, then the system should be turned off.'

Question (5): Is strong anticipation possible in controlling distributed systems?

Answer: Yes, this effects are produced by processes that will be explained subsequently. Daniel M. Dubois has elucidated the difference between weak and strong anticipation:
Definition (Anticipation): An anticipatory system is a system whose current state is determined by a future state. The cause lies in the future [Rosen]. Weak anticipation depends on an internal model of the environment that permits extrapolation into the future. Strong anticipation does not rely on internal models.

Replacing spatial neighbourhoods by temporal neighbourhoods, the distinction between spatial local - and spatial global constraints will be extended to a distinction between temporally local and temporally global constraints. The result obtained in the answer to question (2) states then with respect to temporal constraints:

'Not all temporal ideas that can be specified in the overall system are available for the individual agents, even if they have access to a common knowledge.'

As a consequence of this result, there exist systems in which an omnipotent supervisor would observe anticipation but where no individual agent disposes of an internal model of the system that permits extrapolation into the future.

Whenever in a controlling distributed system, past, present and future satisfy a global constraint that cannot be represented by local constraints, the agents will notice anticipation effects afterwards if they interchange their observations between each other. But in case the symptoms of these effects were not representable in their own language, they are unable to recognise this effects in advance. In classical physics strongly anticipatory effects are excluded by the assumption of Einstein-locality but in quantum mechanics contextuality implies a mutual dependence between past, present and future and thus a global constraint on the system's time development.

Spatial context dependence causes downward causation: a force which every part receives from the overall configuration. Temporal context dependence produces strong anticipation, an information from the future that cannot be exploited by the agents.

5 Applications

5.1 A supervision task:

A group of K agents is monitoring the region of a plant where emergencies emerge. It is assumed that the emergencies appear arbitrarily distributed in the region. In case of an emergency, one of the agents has

- to detect the emergency, what will be feasible for him, if he comes sufficiently close to the point where the emergency had occurred, and
- to approach the place of the emergency.

If one agent has reached the emergency than this emergency disappears.

Solution strategies:

Strategy (A): Each agent selects a random walk.

Observation: If all agents are close together then strategy (A) works inefficiently.

Rule II yields the additional constraint: 'All agents should be uniformly distributed over the whole region'. To realise this constraint with coordination method (III), the following strategy will be used:
Strategy (B): Calculate averaged agent concentrations near observation points $\vec{x}_i, (i = 1, \ldots, L)$:

$$d(\vec{x}_i) = \sum_{k=1}^{K} \exp(-\|(\text{position of agent } k) - \vec{x}_i\|^2) \tag{5}$$

Each agent has to move in the direction of the point: $\vec{x}_{min} = \arg \underbrace{\min}_{\vec{x}_i \text{ close to agent position}} d(\vec{x}_i)$

To measure the efficiency of groups of agents that use different strategies, we compare for them the decrease/increase of the number of unattended emergencies. Numerical experiments demonstrate the superiority of strategy (B).

5.2 A route planning task:

In a town with many streets, K agents search for those routes that bring them from point P to point Q in a minimal effort of time. Each agent takes the decisions of the others into account but a consultation between all agents is impossible. The time T_s needed by an agent to pass through street s depends on the length of the street, on the street condition and on the number N_s of agents that use this street: $T_s = a_s + b_s \cdot N_s$, with appropriately chosen parameters $a_s, b_s \in \mathcal{R}_+$.

Solution strategies:

Strategy (A): Each agent selects the route that minimises his own time effort.

Observation: Often strategy (A) stops in Nash equilibria, where no agent is able to ameliorate its own selection even though a collective optima is missed. An astonishing phenomenon that occurs in these tasks, is denominated:
Braess' Paradox [13]: After an addition of further new streets, by strategy (A) the solution may drift into a worse Nash equilibrium.

Taking these observations into consideration, Rule I yields the additional constraint: 'Avoid contamination of the streets'.

To realise this constraint with coordination method (III), the following strategy will be used:
Strategy (B): The trash produced by agent k will be evaluated by the value:

$$Tr(k) = \sum_{\text{streets } s \text{ utilised by agent } k} b_s \tag{6}$$

Each agent selects the route that minimises its time effort $+\lambda \cdot Tr(k)$, where $\lambda \in \mathcal{R}_+$ is adequately chosen.

It was observed that an adequate realisation of Strategy (B) avoids Braess' Paradox.

5.3 The capacitated facility location task

K consumers, each with the demand of b_k units, will be supplied by N providers with the capacities a_n and the one-time costs f_n.
It is assumed that the overall cost z depends on the parameters:
c_{nk} : costs for the transport of one unit from provider n to consumer k,
x_{nk} quantity requested by consumer k from provider n,
$y_n = 1$ if provider n is in readiness and else $y_n = 0$.
z is specified by the equation:

$$z = \sum_{n=1}^{N} \sum_{k=1}^{K} c_{nk} x_{nk} + \sum_{n=1}^{N} f_n y_n \tag{7}$$

A **capacitated facility location task (CFLT)** [14] means to find the readiness parameters y_n and the request quantities x_{nk} for which each consumer's demand is satisfied and z will be minimised.

Solution strategy using rule (IV): Consumer k will be represented by an agent k. Each agents k calculates its supposed requests x_{nk} such that the value $z_k = \sum_{n=1}^{N} c_{nk} x_{nk} + \sum_{n=1}^{N} f_n y_n$ is minimised under the constraint $\sum_{n=1}^{N} x_{nk} = b_k$. In addition, agent k calculates the maximal cost ω_k that would be necessary if one of its demands could not be fulfilled by the optimal provider.
The priority assignment to the agents is defined inversely proportional to the values ω_k. The agents arrange their demands definitively in the order of their priorities.

6 Conclusion

In analogy to the characterisation of the algorithmic procedures by Church's thesis, we searched for a general methodology to design heuristics for coordination problems. Our analysis provides the following results:

- A general heuristic methodology to specify coordination algorithms had been presented.
- We noted, that not all approaches that can be formulated are algorithmically realisable. Ideas that are representable in the language of the overall system may not be available in local parts of the system and thus for the individual agents.
- Different realisation methods ((I),(II) and (III)) provide classes of differently powerful algorithms.

- Downward causation and strong anticipation are real phenomena in controlling distributed systems.

References

[1] W. V. Quine. On the reasons for indeterminacy. *J. of Philosophy*, 67(6):178–183, 1970.

[2] D. M. Dubois. New trends in computing anticipatory systems: Emergence of artificial conscious intelligence with machine learning natural language. In D.M. Dubois, editor, *CASYS'07*, volume 1051 of *AIP Conference Proceedings*, pages 25–32. American Institute of Physics, 2008.

[3] D. Tolpin and S.E. Shtmony. Rational deployment of csp heuristics, 2011. arXiv:1104,1924v1[cs.AI], 11 Apr 2011.

[4] T. Feder and M.Y. Vardi. The computational structure of monotone monadic snp and constraint satisfaction. *SIAM J. Comput*, 28(1):57–104, 1998.

[5] J.H. Halpern and C. Hitchcock. Actual causation and the art of modelling, 2011. arXiv:1106.2652v1[cs.AI], 14 Jun 20011.

[6] M. Mahdavi, M. Fesanghary, and E. Damangir. An improved harmony search algorithm for solving optimization problems. *Applied Mathematics and Computation*, 188(2):1567–1579, 2007.

[7] M. Nikolic, F. Maric, and P. Janicic. Simple algorithm portfolio for sat, 2011. arXiv:1107.0268v1[cs.AI], 1 Jul 2011.

[8] T. Walsh. Complexity of terminating preference elicitation, 2009. arxiv:0903.1137v18[cs.AI], 6 Mar 2009.

[9] M. Lund and R.K. Belew. No perfect two-state cellular automata for density classification exists. *Physical Review Letters*, 74(25):5148–5150, 1995.

[10] M.C. Coper and S. Zivny. Hybrid tractability of valued constraint problems. *Artificial Intelligence*, 175:1555–1569, 2011.

[11] C. Park, R.M. Worth, and L.L. Pubchinsky. Neural dynamics in parkinsonian brain: The boundary between synchronized and nonsynchronized dynamics, 2011. arXiv:1103.0182[physics], 1 Mar 2011.

[12] A. Geiger, H. Sommer, and A. Kroll. Koevolutionaerer algorithmus zur analyse struktureller steuerbarkeit - eine methode der 'komplexen netzwerke' zur systemanalyse. In E. Huellermeier F. Hoffmann, editor, *20. Workshop Computational Intelligence*, volume 33, pages 235– 249, Karlsruhe, 2010. Institut fuer Angewandte Informatik/Automatisierungstechnik, Institut fuer Technology.

[13] D.H. Wolpert and K. Tumer. Collective intelligence, data routing and braess'paradox. *J. Artificial Intelligence Research*, 16:359–387, 2002.

[14] Minghe Sun. A tabu search heuristic procedure for the capacitated facility location problem, 2011. DOI:10.1007/s10732-011-9157-3.