Polytechnica (2022) 5:21–50
https://doi.org/10.1007/s41050-022-00038-0

ORIGINAL ARTICLE

Oil Supply Chain Integrated Planning based on Holonic Agents and Constraint Programming

F. J. M. Marcellino[1] · **J. S. Sichman**[1]

Received: 30 March 2022 / Revised: 28 August 2022 / Accepted: 28 September 2022 / Published online: 1 November 2022

Abstract
The oil area is one of those that may most benefit from the improved efficiency of supply chain management. However, the dynamic behavior of such chains is too complex to be tackled by traditional approaches. Moreover, these chains show several intrinsic characteristics in common with multi-agent systems, which offer the required flexibility to model the complexities and dynamics of real supply chains without rather simplifying assumptions. Since the problem of managing the supply chain has a recursive structure, it becomes more convenient to use a holonic agent-based model, which show a fractal-type structure. Furthermore, the type of relationship between entities in the chain and the need for global optimization suggest to model their interactions in the form of a constraint network. For this reason, this work defines a new optimization problem called Holonic Constraint Optimization Problem (HCOP), which is based on concepts from Distributed Constraint Satisfaction Optimization Problem (DCOP) and holonic agents. In addition, we developed a meta-algorithm based on DPOP algorithm for solving this type of problem, using the FRODO framework in an environment where available centralized optimization algorithms are integrated so as to obtain the optimization. Finally, experiments were performed on a case study of the PETROBRAS company, where a typical supply chain of the petroleum industry was modeled as HCOP. Those experiments integrated the optimization systems for production and logistics, which are representative in relation to actual situations, and allowed the verification of the feasibility of this model and its comparison with conventional approaches.

Keywords Supply chains · Agents · Holons · Constraint programming · Optimization · Oil and gas

1 Introduction

Supply chains may be defined as an integrated network of facilities and transportation options for the supply, manufacture, storage, and distribution of materials and products. They vary considerably in size, complexity, and scale from industry to industry (Chopra and Meindl 2012). Very few industries can benefit more from maximizing supply chain efficiencies than the oil and gas companies (Chima and Hills 2007). Moreover, the major oil companies operate in a complex way and have producer-consumer relationships between their own units, which can cross national borders. This requires that the operations of logistic and production planning associated with the different production units are properly synchronized and closely coupled. It has also been advocated that the supply chain be managed as an integrated and coordinated system (Forrester 1958).

Constraint programming with optimization is a powerful paradigm that can model a large range of problems like scheduling, planning, optimal process control, etc. Traditionally, such problems are gathered into a single place, and a centralized algorithm is applied in order to find a solution. However, the complexity of the whole chain integration makes the development of a single centralized system an unfeasible task. And even if it were possible, the frequent and unforeseeable changes in the business environment would make the results of such a system obsolete and useless very fast. The dynamic behavior of a complex supply chain is difficult to be taken into account by the traditional models. Usually each entity in the chain is likely to act in its best interests to optimize its own profit.

✉ F. J. M. Marcellino
fjm.marcellino@gmail.com

J. S. Sichman
jaime.sichman@usp.br

[1] Laboratorio de Tecnicas Inteligentes (LTI), Escola Politecnica (EP), Universidade de Sao Paulo (USP), Av. Prof. Luciano Gualberto, 158, travessa 3, Sao Paulo 05508-010, SP, Brazil

Therefore, in general, these models do not meet the goal of the optimization of the entire supply chain.

On the other hand, supply chains and multi-agent systems show numerous intrinsic characteristics in common (Yuan et al. 2003). A problem like this is naturally distributed and since the Constraint Programming approach may provide a tight integration of the involved entities, it allows a global optimization. Thus, Distributed Constraint Optimization Problem (DCOP) was defined as an extension from Distributed Constraint Satisfaction Problem (DisCSP) (Modi et al. 2003), which had been formalized by Yokoo et al. (1992) previously. In general, an optimization problem is much harder to solve than a DisCSP, as the goal is not just to find any solution, but the best one. In both paradigms the problem is divided among a set of agents, which have to communicate with each other to solve it.

After further analysis of the real constraint optimization problems, it is possible to realize that some of them own a recursive nature, which is not currently exploited by the available distributed optimization frameworks and their associated algorithms. Since the supply chain integrated planning is an example of that kind of problem, a holonic agent approach proved to be quite appropriate for it and that is the specific paradigm used in this work. Thus, it is proposed a distributed approach based on holonic agents using constraint optimization to model a planning problem which owns the main components of the typical oil supply chain. For its solution this work defines a Holonic Constraint Optimization Problem (HCOP) as a new paradigm to model distributed optimization problems with recursive nature using the integration of solvable sub-problems into which they may be naturally partitioned. Thus, it specifies an architecture for the integration of local algorithms and optimization softwares associated with those sub-problems, which allows the optimization of the whole problem. In addition, to achieve the referred solution and integration, a meta-algorithm was developed to deal with HCOP, which is called HCOMA (Holonic Constraint Optimization Meta-Algorithm).

Section 2 makes a review about the supply chain for the oil industry, discussing the available models to solve its integrated planning problem, their motivations and the challenges faced by them. Section 3 synthesizes the basic concepts involved in this work model, whereas Sect. 4 describes and formalizes HCOP. In addition, that section presents a meta-algorithm (HCOMA) for its solution. Section 5 models the integrated supply chain planning problem addressed in this work as HCOP, whereas Sect. 6 shows the performed experiments and its results, showing the viability and the advantages of the proposed approach. Finally, the conclusions and an outlook on future research activities are presented.

2 The Oil Supply Chain Planning Review

The petroleum industry has a typical supply chain. According to Eichman (2000), managing such supply chain presents some of the most difficult challenges found in Supply Chain Management. This supply chain covers from the stage of oil extraction up to the distribution of derivative products, including a complex logistic network and various transformation processes occurring in refineries. The activity of oil extracting represents the raw material production, while the refining is the manufacturing stage of the supply chain and its output are the final products like gasoline and diesel. In order that these products reach their destination, which is represented by the customer companies, they must be carried through different transport modes such as pipelines and vessels.

Grossmann (2014) presents Enterprise-Wide Optimization (EWO) as a major goal in the supply chain of *Process Industry (PI)*, which ranges from the petroleum industry to the pharmaceutical one. EWO involves optimizing the operations of supply, manufacturing, and distribution activities of a company belonging to such industries to reduce costs and inventories It includes manufacturing activites, which often requires the use of nonlinear process models, as well as planning, scheduling and inventory control. In order to achieve this goal one of the key features is the integrated and coordinated decision-making across the various functions in a company (purchasing, manufacturing, distribution,sales), across various geographically distributed organizations (vendors, facilities and markets), and across various levels of decision-making (Shapiro 2006). In general, the supply chain planning is classified into three levels: strategic (long-term), tactical (medium term), and operational (short-term). The long-term planning covers the time horizon from one to several years, the medium-term a few months to a year, and short-term covers a week to three months (Grossman et al. 2001). Strategic planning determines the supply chain structure (eg, production capacity expansion). Tactical planning affects decisions such as the allocation of production targets and transportation up to the customers. On the other hand, operational planning determines the tasks of production units and the products transportation mode, considering resource constraints and time. Furthermore, the scheduling concerns the detailed information on decisions such as sequencing and allocation of tasks to resources in order to meet the targets set by the planning (Magalhaes et al. 1998).

The information is shared along the various chain entities by modern IT tools. However, these do not provide comprehensive decision making capabilities for optimization that account for complex tradeoffs and interactions

across the various functions, subsystems and levels of decision making. Lasschuit and Thijssen (2004) emphasize the importance of achieving full integration in these process industry supply chains and describe the desire of a tool for this purpose. It is explained the need for decision support tools that satisfy a fundamental need that the decisions of the strategic and tactical level could take into account operational information, such as, utilization of production capacity, utilization of transportation modes and allocation of demand.

The key issues in the PI supply chain management can broadly be divided into three main categories: (i) supply chain design (infrastructure), (ii) supply chain planning and scheduling and (iii) supply chain control (real-time management) (Garcia and You 2015). This paper focuses on the second category, but integrated supply chain planning at tactical and operational level for the oil industry. However, the proposed model may be applied to a generic PI supply chain eventually. That model aims to maximize expected profit across the chain with customer satisfaction guarantee. The next subsections describe the available models to treat this problem, the solution strategies, the comparison between the distributed and centralized approach and the major challenges to be faced. Finally, the proposed model is introduced.

2.1 Supply Chain Planning Modeling

Generally, models for design and analysis of the supply chain can be divided into four categories: (1) deterministic analytical models, in which the variables are known and specified, (2) stochastic analytical models, where it is assumed that at least one of the variables follows a particular probability distribution, (3) economic models, which aim at providing a qualitative criterion with respect to economics and (4) simulation models, which are used to understand and predict the behavior of systems by experimentation (Beamon 1998). Another way to classify the models is between *descriptive* and *normative* models (Shapiro 2006). The former are developed for a better understanding of the functional relationships inside the chain and between it and its environment (economic and simulation models). The latter are the normative models, which are developed in order to assist in making the best decisions. The term normative refers to processes to identify the norms that the company should strive to obtain. According to Shapiro (2006) this type of model is to be confused with *optimization* models, and this is the point of view of this work. Within this group are analytical models, both deterministic and stochastic ones, and both centralized as distributed ones. Incidentally, the simulation is not suitable for the purpose of achieving optimization, particularly in complex problems like the supply chain integrated planning. These models are associated with *optimization problems*, which have to minimize or maximize an objective function f subject to the satisfaction of a given set of constraints (Papadimitriou and Steiglitz 1982). These problems can be classified in several ways, for example, according to the nature of their objective function (*linear* or *nonlinear*) or to the type of their variables (*continuous* or *discrete*), where in the first case the variables assume real values, while in the latter the values are integer and the problem is also called *combinatorial*. The most common frameworks employed to build the analytical optimization model are briefly described next. They are chosen according to the nature and the characteristics of the problem. They are strongly related to *Operations Research* (OR) and *Artificial intelligence* (AI) areas.

Mixed-Integer Linear Programming (MILP) is used to model a large number of PI supply chain problems like planning, scheduling, logistics, distribution and manufacturing. The associated real world problems usually lead to large scale models, due to the size of the system under study. In recent years great progress has been made in algorithms and hardware, which has resulted in an impressive improvement in their ability to solve this kind of model. However, integrated models with detailed formulations often result in large MIP models that can not be solved with optimization (Grossmann 2014). One way to overcome this limitation is through the use of advanced solution strategies, a topic that will be reviewed in the Sect. 2.2.

Mixed-Integer Nonlinear Programming (MINLP) is less common than MILP in PI supply chains, since solving MINLP problems is a non-trivial task. However, there is an increasing interest in sub-problems that require handling nonlinearities. Methods used to solve MINLP models are generally a direct extension of those employed for MILP models, but their application may be computationally very expensive for large scale problems. Therefore, a frequent approach is to reformulate the MINLP problem as MILP by using exact linearizations or using piecewise linear approximations (Grossmann 2014).

Dynamic Programming breaks the optimization problem into smaller sub-problems, so that it can be seen as a synthesis of optimal solutions for those sub-problems. In this case the principle of optimality applies, where solutions are constructed for the most trivial sub-problems first and those solutions are extended with branch and bound to larger problems. An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision (Bellman 1957). This method is also called *recursive optimization*.

Constraint Programming (CP) has been recognized as a suitable modeling and solving tool to face combinatorial optimization problems, which appear in many real life application, such as production scheduling, DNA sequencing, hardware design, protocol simulation, etc. This class of problems is, in general, extremely difficult to solve. It does so via the search, propagation and optimization processes (Ajili and Wallace 2003). The constraint-based scheduling is one of the most successful application areas of CP. One of the key factors of this success lies in the fact that a combination was found of the best of two fields of research that pay attention to scheduling–namely, OR and AI. The use of CP in planning is less mature than its use in scheduling, because of its problem complexity. However, constraint-based planning follows the same pattern as constraint-based scheduling where CP is used as a framework for integrating efficient special purpose algorithms into a flexible and expressive paradigm (Baptiste et al. 2006). The CP modeling and solving activity is highly influenced by the AI area and apply to a problem category called Constraint Satisfaction Problems (CSP) (Tsang 1993). It is described in more detail in Sect. 3.2, since it is one of the basics of the model proposed in this work.

2.2 Solution Strategies

Techniques based on a decomposition process into two levels are used for the integration of planning and scheduling. These usually involve Lagrangean decomposition, Benders decomposition, rolling horizon algorithms, etc. Thus, the higher level problem (planning problem) is an aggregation of the lower level problem (scheduling). Another solution approach is based on using a moving horizon technique, where the planning problem is solved by means of the treatment of the first periods in detail, while the later periods are aggregated recursively (Dimitriadis et al. 1997). Another option is to develop an approximation of the original model, which provides some information in the short term, which makes easier its solution. An example of approximate model is obtained by removing some of the constraints or by aggregating some of the original scheduling formulation decisions (Maravelias and Sung 2009). In sum, a compromise between accuracy and computational load of the modeling is to use detailed scheduling models for some initial periods and a relaxed, aggregated or surrogate formulation for subsequent periods. In the case of spatial decomposition the idea is to separate the links between subsystems through the dualization of the corresponding interconnection constraints. On the other hand, the temporal decomposition requires the multi-period approach. Since the PI supply chain planning problem is multi-period in nature, Van Den Heever and Grossmann (1999) analyzed the multi-period planning models for that kind the industry.

In order to be computationally tractable an alternative method is generating an accurate description of parts of the model by performing calculations offline. The goal of this approach is to generate constraints which can be resource-intensive, but once they are done offline, they may be incorporated in the formulation integrated without additional computing. They are called *offline surrogate models* (Maravelias and Sung 2009).

In a more general way, through decomposition methods the problem may be decomposed in a master sub-problem (high level) used to determine production targets and a slave sub-problem (low level) with detailed scheduling (operations). Production targets or other high-level decisions are used as inputs to the slave sub-problem. If the information flow is only from the master sub-problem to the slave sub-problems, then the methods are *hierarchical*. If there is a feedback loop of sub-models back to the master sub-problem, then the methods are *iterative*. If the integrated formulation contains submodels of detailed scheduling for each planning period, then its solution provides all the necessary information. However, these models are difficult to solve and require advanced solution methods. They are called *full-space* (Maravelias and Sung 2009).

Lima et al. (2016) performed a review which considers the relevant works around the usage of optimization-based decision-support tools applied to the downstream oil supply chain. Its main objectives are to point out main contributions, besides identifying the major voids and new trends in order to establish an agenda for future research directions. The selected papers are classified into two main groups according to the decision-making levels, namely: strategic and tactical planning; and tactical and operational planning. As this work belongs to the second group, just the results of the latter are shown here. The developed models aim to the integration between tactical and operational decision levels, as well as the integration between the different segments in the supply chain, in order to coordinate the entire system to fulfill the demand through improving the overall results, within a system perspective. Based on these reviewed articles, the integration between tactical and operational decision levels is identified as required in order to avoid infeasibilities, because when operational constraints are not taken into account on the tactical planning, the model can lead to either infeasible solutions or suboptimal solutions, due to the conditions imposed by the upper planning level to the lower planning level. However, in the operational level, given the established constraints at the tactical level, the goal is to optimize the network performance in the short term, where uncertainties are fewer. Within the ten papers addressing the tactical and/or operational planning, seven models are deterministic and three models are stochastic. The limitation is in the period of time considered and in the size of the problem chosen for the solved case studies, which must be larger as the actual oil supply chain networks are very complex. According to Lima et al. (2016), it is necessary more

efficient methodologies and solution techniques to aid the decision-making process by developing more robust optimization decision-support tools to cope with a larger range of complex problems. Also, these techniques should pursue the optimization of the entire network, where the major objective is the integration between planning and scheduling of the downstream oil supply chains.

2.3 Distributed × Centralized Approach

A summary analysis of the supply chain planning problem leads to the conclusion that only its sub-problems individually have been studied in reasonable detail. Most tools focus on specific parts of the oil supply chain, which often lead to a lack of integration. The reason of that is the complexity that arises when all these parts are put together within the same model. Nevertheless, there is strong motivation to increase the scope of the models of the oil supply chain without reducing the complexity of the real problem (Vecchietti and Grossmann 2000). On the other hand, simultaneous optimization approaches for the integration of entire supply chains lead to the definition of centralized systems.

However, in general, the centralized approaches are based on mathematical models which need simplifying assumptions, usually restrictive, which lead to analytical solutions, but far from the desired optimal situation. In practice, however, the actual operation tends to happen as if the supply chain were a decentralized system. It is needed that coordination procedures can maintain a certain degree of subsystem independence, while, at the same time, aiming at the objectives intended by the global integrated optimization of the system (Perea et al. 2001). The supply chain is composed of autonomous entities which possess roles and responsibilities defined from their expertise and activities that they are able to accomplish. Thus, it is necessary a model that takes into account all dimensions of the network and the aspects of the organization, as well as the distributed nature of these entities within the supply chain, with its dynamics and decision-making autonomy (Labarthe et al. 2007). Therefore, that agents approach, which was adapted to the representation of complex systems and organizations, is used in this work and is more detailed in the Sect. 3.1.

2.4 Major Challenges

The following are the major challenges that arise in the application of the previous modeling techniques to PI supply chain problems (Grossmann 2014). In the next subsection they are analyzed in relation to the model proposed in this work.

1. **Linear versus Nonlinear Models**. Linear models are traditionally used in operational research approachs, but in the supply chain integrated optimization the emphasis is on production facilities with greater focus on logistics, containing sub-problems which require a realistic optimization and nonlinear models. In a number of cases it is possible to use approximate MILP models to solve MINLP models, since the latter may be prohibitive for large scale problems. So the dilemma is whether to rigorously solve the approximate MILP model, or whether to obtain an approximate solution to the rigorous MINLP models.

2. **Multiscale Optimization**. Addressing PI suppply chain problems for functional, spatial and temporal integration is critical to optimize decision making throughout the chain. In order to face it two main approaches are: considering a simultaneous optimization model in large scale, or the use of decomposition in both spatial and temporal forms (Graves 1982). The spatial integration of geographically distributed manufacturing and inventory facilities in supply chains leads to large scale problems that often require the application of specialized decomposition techniques, as already mentioned. The temporal integration, however, requires effective representations and strategies in the first place so as to integrate long term design decisions, with intermediate term production planning and short term scheduling decisions. Then these require in turn decomposition schemes for the optimization across different time scales (Maravelias and Grossmann 2004).

3. **Optimization under Uncertainty**. Uncertainties are common in PI supply chain problems. For long term strategic problems, *stochastic programming* is better suited because of its capability to account for recourse actions for the different scenarios (Sahinidis 2004). However, it is advisable to first develop computationally effective deterministic models that can be used as a basis for developing corresponding robust stochastic programming models in the future.

4. **Optimizing Entire Supply Chain**. Although important progress has been made in modeling and optimizing major components of PI supply chains, the optimization of entire supply chains still remains an elusive problem. The difficulty is partly due to the very large size of the resulting models, but also on account of the somewhat distinct nature of these major components. Particularly in the oil supply chain, it is not obvious how to integrate the models for upstream exploration, marine transportation, crude oil delivery unloading and refinery optimization, for instance, since they often rely on both different modeling paradigms and space-time representations. To achieve the goal of planning optimization across the entire chain will require not only advances in new computing architectures, algorithms and individual models, but mainly a new macro model which integrates all of the latter ones.

2.5 Our Approach

The academic literature tends to emphasize computation speed when evaluating a new approach to problem solving, perhaps because it is easily measured. Practitioners know, however, that model development time is often at least as important as

solution time. This argues for the convenience of having all the modeling and algorithmic resources available in a single integrated system. One can try several approaches to a problem without having to learn several systems and port data between them (Hooker 2007). That is the main idea proposed in this work. In fact, it is an distributed architecture which may contain several submodels, whether linear or nonlinear, according to the needs of the respective sub-problems in which the complete problem has been divided. Moreover, that distributed architecture allows the interconnection of the available optimization software or algorithm, aiming at the global best possible optimization. It is employed the agent technology, but the holonic agent, which is suitable for dealing with the different problem scales, providing different granularities.

As already mentioned, the planning problem addressed in this work considers only the tactical and operational levels, so it was preferable to focus upon a deterministic model. However, its distributed nature allows a quick reaction in order to deal with unforeseen changes in the environment (*Random Uncertainty*). At the same time, the model deals with the *Epistemic Uncertainty* associated with the great complexity of the problem in question, as it will be seen later.

In CP the problem is easily modeled as a set of sub-problems each represented by a constraint, as in the proposed model. Thus, integration is accomplished by using CP as an infrastructure connecting submodels, which are solved by any available optimization software or algorithm, as long as using finite domains, as required by CP. Instead of domain reduction, as in the CP inference process, in this work there is a domain granulation, making it meaningless to consider continuous domains, as it will be seen later.

In sum, the proposed model addresses the complete integration by modeling the chain components, which must be connected via a holonic network and solved using the respective available optimization software or algorithm. The global best possible optimal solution across the entire chain is possible due to the constraint optimization approach.

3 Building Blocks and Adopted Technologies

This section briefly presents the basics of the model proposed in this work, which are Holonic Agents and Constraint Programming. The distributed behavior of the model is present in the choice of both agent technology and the distributed version of the optimization constraint framework.

3.1 Holonic Agents

A Multi-Agent System (MAS) is a collection of active entities, called agents, which are able to act on itself and on the environment in which it evolves, and communicate with other agents (Ferber 1995). Complex systems are characterized by multiple interactions among many different components. They are called complex because design, or function, or both, is difficult to understand and verify. The behavior of the system is the result of the nonlinear aggregation of local behaviors of its components (Hilaire et al. 2008). According to Jennings (2000), the agent modeling presents three advantages associated with tools to handle the software complexity: *Decomposition*: efficient partitioning of the problem into easier sub-problems; *Abstraction:* definition of a simpler model that emphasizes the details and important properties, while suppressing others more superfluous and *Organization:* focus on relationships between the various components relevant to the problem solution.

Therefore, MAS has become a natural tool for modeling, simulating and programming complex systems. However, in those systems there usually are a great number of entities interacting among themselves, and acting at different *abstraction levels*. In this context, it seems unlikely that MAS will be able to faithfully represent complex systems without multiple granularities. This is the reason holonic agents have attracted the attention of researchers (Hilaire et al. 2008). That concept will be introduced in the next subsection.

3.1.1 Holonic Paradigm

The term *holonic* is derived from the word *holon*, which was introduced by the philosopher Arthur Koestler (Koestler 1967), as a combination of the Greek *holos* (whole) and the suffix *on* (part). He analysed recursive and self-similar structures which behave as stable intermediate forms in both living organisms and social organisations. According to this point of view, no natural structure is whole or part in an absolute sense, but in fact every holon is a composition of subordinate parts, as well as a part of a greater whole. Thus a holon can be seen, depending on the level of observation, as an atomic and autonomous entity, such as an agent, or as an organization of other holons. A holon (superholon) is composed of other holons (subholons) and should meet three conditions: (i) to be stable, (ii) to be autonomous and (iii) to be able to cooperate.

A *holarchy* is a hierarchy of self-regulating holons that function first as autonomous wholes in supra-ordination to their parts, secondly as dependent parts in sub-ordination to controls on higher levels, called *echelon*s, and thirdly in coordination with their local environment (Koestler 1967). In contrast to hierarchies, this type of structure also considers the decision power of the lower organizational levels. Another advantage is the ability to map an application domain directly in a multi-agent system through *agentification* of entities at any level of granularity and without losing higher level abstractions (Gerber et al. 1999). The

strength of the holonic paradigm is its recursive definition of holons. Thus it is well adapted for large complex systems where different granularities are required. Holonic systems offer the possibility to model a system from a high-level coarse-grained perspective to a low-level fine-grained one. Another type of systems are those that can be decomposed recursively into smaller sub-components (Rodriguez et al. 2005). Thus, the holonic system can be actually seen as a multi-agent systems specialization, which promises a better modeling of the integrated planning problem, which, in fact, has great complexity, multiple granularity levels and some entities with recursive structure.

3.1.2 Holonic Multi-Agent System

Holonic Multi-Agent System (HMAS) has a structure formed by a set of hierarchical levels, where the agents can interact only with other agents at the same level or at the level immediately below or above. It was proposed three types of holon organization, which vary with respect to the degree of autonomy of its members (Gerber et al. 1999). The *moderate group* is the intermediary structure, which was chosen for this work due to its greater flexibility and where agents give up only part of their autonomy. According to Hilaire et al. (2008), this kind of holonic structure owns three main roles: *head* players are moderators of the holon, whereas represented members have two possible roles: *part*, whose players belong to only one superholon, and *multipart*, where subholons belong to more than one superholon at the same time. The *head* represents the shared intentions of the holon and negotiates them with other holons outside. The remainder of the holon, i.e. the set of parts or multiparts, is called *body*. This organizational model will be named *holonic organization* hereafer. Besides it there is another organization model called *internal organization* (Hilaire et al. 2008), which is dependent on the problem domain and models this second aspect of the holons related to their interactions, aiming at their goals. Thus, one holon is a special type of agent which exhibits holonic roles associated with the holonic organization and, at the same time, playing also the roles defined by its specific problem internal organization.

Every HMAS needs a *holonification* process for creation of the holons structure, which specifies the subholons within each holon and the different levels in the holarchy. Holonification reduces the complexity of a network by dividing it into much simpler units such that a large scale problem may be converted to multiple smaller sub-problems that are easier to solve. It is considered very critical in design of holonic multi-agent system and an improper method may increase the complexity of the system and decrease the efficiency (Abdoos et al. 2012). It is very similar to the concept of partitioning in graph theory, where an agents' network is partitioned into communities or superholons in such a way that the most related group of subholons belong to the same holon. That problem, such as the graph clustering process, is NP-hard (Brandes et al. 2007). However, given the model proposed by this work, it is not necessary to worry about the algorithm aiming at that goal, since holonification is taken for granted by the natural modeling process, as it will be seen in Sect. 4.1.

3.1.3 Granularity Theory

We look at the world under various grain sizes and abstract from it only those things that serve our present interests. The ability to conceptualize the world at different granularities and to switch among them is fundamental to our intelligence and flexibility. It enables us to map the complexities of the world into simple theories that are computationally tractable to reason in. If there is a machine of even moderate intelligence, it must have a theory of granularity woven into the very foundation of its reasoning processes. For this reason, Hobbs (1990) has suggested a theory of granularity in which a complex theory is abstracted onto a simpler, more *coarse-grained* theory with a smaller domain.

Assume that a real problem, whatever its complexity, is represented in a global theory, which may dealt with through a first-order logical theory T. The proposed approach to granularity is to extract from that global theory smaller, more computationally tractable, local theories. Let P be the set of predicates of T, and S its domain of interpretation. Suppose a subset R of P has been determined to be the predicates relevant to the situation at hand. Then it can be defined an *indistinguishability relation* $\sim$ on S by means of the following second-order axiom:

$$(\forall x,\ y)\, x \sim y \equiv (\forall p \in R)\, (p(x) \equiv p(y)) \tag{1}$$

That is, x and y are indistinguishable if no relevant predicate distinguishes between them. According to Hobbs, it is expected that in the course of modeling a problem in general, the set of relevant predicates becomes more constrained, and as it does, more entities become indistinguishable for all practical purposes.

In addition, various local theories must be linked with each other by means of articulation axioms, to allow shifts of perspective. Thus, a theory of granularity must say something about how various local theories articulate with each other. There has been a certain amount of work in AI on this problem - research on hierarchical problem-solving in expert systems and on hierarchical planning. When we move from one level of a hierarchy to the level below, we are moving from a coarse-grained local theory to a more fine-grained local theory, and the axioms that specify the decomposition of coarse-grained predicates into fine-grained

ones constitute the articulation between the two theories. When shifts in perspective are required, we must translate the problem from one local theory to another. In this situation articulation axioms are used.

In a certain sense the theory of granularity is applied when a problem is modeled using the constraint paradigm, since a complex theory of the real world of continuous quantities is mapped onto a simpler (micro) world of variables with discrete domains. The model proposed in this paper goes further by applying this theory in the articulation between local theories associated with consecutive echelons in the holonic organization as it will be seen later.

3.2 Constraint Programming Paradigm

CP is a paradigm that combines declarative description of problems with efficient algorithms and solving techniques (Tsang 1993). As already mentioned in the Sect. 2.1, the CP is a suitable framework to model various AI problems such as scheduling and planning, which constitute a category called *Constraint Satisfaction Problem* (CSP). It consists of a set of variables $V = \{x_1, \ldots, x_n\}$, each having a corresponding set of finite and discrete domains $D = \{D_1, \ldots, D_n\}$, and a set of constraints $C = \{c_1, \ldots, c_m\}$ specifying which values of the variables are compatible with each other. A solution to a CSP is an assignment of values (an instantiation) to all variables, such that all constraints are satisfied. Thus, a *Constraint Network* is defined by the triple (V, D, C). The arity of a constraint refers to the cardinality, or size, of its scope. A unary constraint is defined on a single variable; a binary constraint, on two variables. A binary constraint network has only unary and binary constraints (Dechter and Cohen 2003). Later work in Distributed Artificial Intelligence (DAI) has considered the distributed CSPs (DisCSPs) in which CSP variables are distributed among agents (Faltings and Yokoo 2005) (Yokoo et al. 1992).

CSP may be extended to become an optimization problem as defined in the Sect. 2.1. In this case it is called *Constraint Optimization Problem* (COP), which must find a complete assignment of values to all its variables, satisfying all the constraints, and optimize the objective function as well. In a classical CSP all the constraints must be satisfied, but real-life problems frequently involve both hard and soft constraints, where the first must be satisfied and the latter represent preferences rather than strict requirements. Thus, it has been defined a valued CSP (VCSP), which is obtained by annotating each constraint with a valuation (usually a number), which expresses the impact of its violation. These valuations are combined using an operator that gives specific semantics, so that the solution represents an assignment with a minimum valuation. In other words, VCSP works like a COP, where such valuation corresponds to the optimal solution (Schiex et al. 1995). Just as DisCSP is the distributed CSP, the distributed problem associated with VCSP is called DCOP. It is presented next.

3.2.1 DCOP

Distributed Constraint Optimization Problem (DCOP) is a formalism that can model optimization problems distributed due to their nature. These are problems where agents try to find assignments to a set of variables that are subject to constraints. It is assumed that agents optimize their cumulated satisfaction by the chosen solution. This is different from other related formalisms involving self-interested agents, which try to maximize their own utility individually. Thus, the agents can optimize a global function in a distributed fashion communicating only with neighboring agents, and even in a asynchronous way. A DCOP consists of n variables $V = \{x_1, \ldots, x_n\}$ each assigned to an agent, where the values of the variables are taken from finite and discrete domains $D = \{D_1, \ldots, D_n\}$, respectively. Only the agent who is assigned a variable has control of its value and knowledge of its domain. The goal for the agents is to choose values for variables such that a given global objective function is *minimized* (or *maximized*). The objective function is described as the summation over a set of cost functions, each one for a pair of variables x_i, x_j and defined as $f_{ij} : D_i \times D_j \rightarrow \mathbb{N}$. The cost functions in DCOP are the analogue of constraints from DisCSP, but they are referred to as *valued* or *soft* constraints (Modi et al. 2003).

3.2.2 Generalization of DCOP for Complex Local Problems

According to the DCOP definition in the previous section, each agent controls only a single variable. This limits the applicability of its algorithms to distributed practical applications and leaves some important open questions (Burke 2008): how to solve complex local problems (multiple variables per agent) as part of a larger global problem? How to integrate the local solving process with the distributed search? It was shown that any DCOP with complex local problems (multiple variables per agent) can be transformed to an original DCOP with exactly one variable per agent by problem reformulations. Therefore, a *DCOP with a Complex Local Problems* is defined by a tuple (A, V, D, F), where (i) $A = \{a_1, \ldots, a_n\}$ is a set of n agents; (ii) $V_i = \{v_{i1}, \ldots, v_{im_i}\}$ is a set of variables which the agent a_i controls, such that $\forall i \neq j \; V_i \cap V_j = \emptyset$; $V = \cup V_i$ is the set of all the variables of the problem. The variables can be classified into 2 categories: *private* or *local* variables, which participate only in internal constraints of the corresponding agent (intra-agent constraints) and *public* variables, which also participate in external constraints with other agents (inter-agent constraints); (iii) $D = \{\ldots, D_{ij}, \ldots\}$ is a set of finite and discrete domains, where D_{ij} is associated with the corresponding

variable v_{ij}; and finally (iv) $F = \{f_1, \ldots, f_i, \ldots, f_k\}$ is a set of constraint functions, with $f_i : \prod_{ij} D_{ij} \rightarrow \mathbb{N}$. The goal is to find a complete instantiation V^* for all variables v_{ij} that minimizes (or maximizes) the global objective function $OF = \sum f_i$ (Burke 2008) (Fioretto et al. 2018).

3.2.3 DCOP Algorithms

Several search-based distributed algorithms have been proposed for DCOP, and many of them share a number of common features. In most algorithms, during search, each agent executes as an autonomous entity and repeatedly performs three core tasks as part of a single computation cycle like DisCSP (Yokoo and Hirayama 1998). It is shown in Algorithm 1.

Algorithm 1 DisCSP Algorithm Cycle

```
1: for all agents do
2:     read incoming messages from other agents;
3:     perform some computation;
4:     send messages to other agents;
5: end for
6: return
```

Using a cycle as a basic building block, Lynch (1996) broadly categorizes distributed constraint reasoning algorithms according to three different timing models: *synchronous,* where all agents' cycles are executed simultaneously, *asynchronous*, where all agents' cycles are executed in an arbitrary order in parallel, and *partially synchronous*, which is a combination of both. These algorithms may be also classified as complete or incomplete algorithms. Since a DCOP solution is always optimal by definition, its algorithm is complete when it is guaranteed to find an optimal solution. In addition, they may be also divided into search-based or inference algorithms as in CSP. The search-based algorithms generally employ the backtracking method and send one or more message for each new variable assignment. Given that the number of possible assignments in the search space is exponential in the number of variables, this means that the number of messages grows exponentially. ADOPT (*Asynchronous Distributed Optimization*) is a representative algorithm of this category (Modi et al. 2003). It is a search-based algorithm which operates asynchronously. It is proven complete. On the other hand, the inference algorithms share constraints instead of variable assignments. In this work they are represented by DPOP (*Distributed Pseudotree Optimization Procedure*) (Petcu and Faltings 2005), which is based on Dynamic Programming (see Sect. 2.1). It provides a linear number of messages, but the message size is exponential. It is an evolution of DTREE algorithm for arbitrary topology (Petcu and Faltings 2004). It is proven complete too.

Regardless of the category, DCOP algorithms must prioritize the agents before executing. In some algorithms, agents are prioritized into a chain, while others prioritize agents into a Depth-First Search (DFS) tree. This prioritization has the property that any two neighbouring agents appear on the same branch of the tree, i.e. for any agent in a DFS tree ordering, constraints are only allowed between it and its ancestors or descendants (Petcu 2009). Both ADOPT and DPOP prioritize agents in a DFS tree. Due to its importance and representativity in the research community area these two DCOP algorithms are used in this work.

The prioritization structure and the message protocol defined by the algorithm determine the messages that are sent between the agents in a DCOP algorithm. While the number of messages used by the different algorithms varies, most of them have at least 2 types of messages :

- VALUE: represents the current assignment value of a variable managed by the concerned agent. These messages are sent from the highest to the lowest priority agents. They specify the assignment of the sending agent, and sometimes the assignments of other higher priority agents.
- COST or UTIL: represents the current optimized value in relation to all variables belonging to concerned agent subtree. These messages are sent from lower to the higher priority agents. They typically specify a cost incurred by the sending agent and all its lower priority agents.

In the framework defined by the message protocol, agents systematically propose value assignment and record the costs (or utilities), exploring the search space of possible assignments. This continues until the algorithm finds an optimal global assignment.

3.3 Related Works

According to Giret and Botti (2004) the agent research is mostly involved in the investigation of behavioral models, cooperation and coordination strategies, while the holons have been used for the distributed intelligent control. It has been proved that HMAS is an effective solution for several problems associated with hierarchical and self-organizing structures (Rodriguez et al. 2005). It has been successfully applied in a wide range of complex systems, mainly in manufacturing systems, where it was done for the first time by Suda (1989), and it led to the proposal of Holonic Manufacturing Systems (HMS) later (Hms 1994). In addition, for instance, it was employed in areas such as traffic signals network control (Abdoos et al. 2013), health organizations (Ulieru and Geras 2002), and complex software systems (Moise 2008). Although there is any holonic application dedicated to the Supply Chain Management Problem as a

whole as yet, it is worth presenting two works which focus on parts of the chain. Both involve transport activities and one of them employs CP paradigm too.

The first work is called TELETRUCK (Burckert et al. 1998). Its purpose is to model the allocation of resources like drivers, trucks or trailers to transport requisitions arriving at a transportation company. Holonic agents were used in a multi-agent system of a trucks fleet scheduling. The holonic structures for the TELETRUCK project were implemented through the moderate group organization (see Sect. 3.1.2). This decentralized approach is appropriate for that complex scenario, since local information is sufficient for a globally efficient distribution of tasks and resources.

The second application is named Global Automated Transportation System (GATS) (Versteegh et al. 2010). It is an integrated transport system based on an idea of Zelinkovsky (1999), which allows locomotion without the need of a driver. Its future objective would be that millions of vehicles might be driven simultaneously and automatically across a virtually limitless geographic region. Holonic architecture proved appropriate with GATS decentralized and modular nature and with its ability to coordinate simultaneously the macro and micro needs of the road transport networks. In fact, traditional centralized techniques are unable to model and implement such problems due to their size and complexity. In that architecture each holon is responsible for solving a scheduling sub-problem, which may represent a *continent*, a *country* or a *region*, obeying a holonic organization. Each sub-problem is modeled as a CSP, which is best suited for scheduling problems (see Sect. 3.2), but in a distributed way. Thus, the complete problem is divided into a set of sub-problems, which are solved simultaneously to try to achieve a global solution in an acceptable time. The distributed holonic model has also the advantage of using a same algorithm at all levels in a recursive way. However, that model does not address the main problem as an optimization problem, since it depends on a negotiation strategy between the holons to achieve a global solution, which is generally sub-optimal.

The decision to use the constraint optimization approach, as well as the distributed holonic agents paradigm, forms the basis of the model used in this work. However, the straightforward use of these concepts could make the task very difficult and less elegant. Therefore, it was necessary to create a sounder theoretical base, involving all these elements in a consistent way, before starting the problem modeling task itself. This is exactly what is done in the next section.

4 Holonic Constraint Optimization Problem (HCOP)

The possibility of a holonic agent approach to address the supply chain planning integration for a global optimization first occurred in Marcellino and Sichman (2010a). Later Marcellino and Sichman (2010b) proposed the DCOP framework for a distributed optimization. However, that model was still incipient and insufficient for the proposed objective. Hence, it has evolved into the Holonic Constraint Optimization Problem (HCOP) as a subclass of DCOP, which was first introduced in Marcellino and Sichman (2011). HCOP model has been further improved up to the more mature version, which is shown in this work.

4.1 Description

HCOP is a distributed constraint optimization problem with a holonic organization, so that it has a holarchical and recursive structure. It consists of a set of holon agents, which are distributed into different abstract levels, which are named *echelons.* Like any *holarchy* each holon may contain other subholons and it is part of a superholon. However, the most fundamental echelon ($\eta = 0$) comprises only atomic holons, each one formed by a single conventional agent. The highest echelon consists of an unique holon called *global holon*, which contains the whole system. There may be atomic holons at the intermediate echelons between these two extremes. However, they can not exist in isolation, being parts of a superholon located at the next higher level.

Each holon is responsible for a variable called *holonic variable*. In the case of an atomic holon, it is a *decision variable*, which is an independent variable in the same way as a DCOP variable (see Sect. 3.2.1). On the other hand, each non-atomic holon belonging to a higher echelon ($\eta > 0$) is called superholon. In this case, its *holonic variable* is an *emergent variable*, which is dependent on the *holonic variables* associated with its subholons. Such dependency is imposed by a *holonic constraint,* which is called *emergence constraint* and defined in turn by an *emergence function.*

The model adopts the *moderate group* as holonic organization*,* with the role *head*, which is unique for each superholon, and the role *part*, which may be unique or multiple (see Sect. 3.1.2). It is also assumed that the *head* holon is a special agent, which is responsible for the internal coordination among the other *part* holons and the communication with the outside world. Due to the distinctive behavior of the *head* holon, it is natural to treat it differently from the remainder of the holon, which comprises all the *part* holons and is called *body*. For the sake of model elegance, it is a conventional agent such as an atomic holon.

The proposed structure is characterized by entities with great cohesion with respect to their fellows, but only a coupling relationship with their parents and children along the holarchy. The coupling between each holon and its superholon or its subholons is generated by the *holonic constraints*, being less intense than the internal strong cohesion among the subholons inside a same superholon. This latter type of cohesion is represented by another organization named

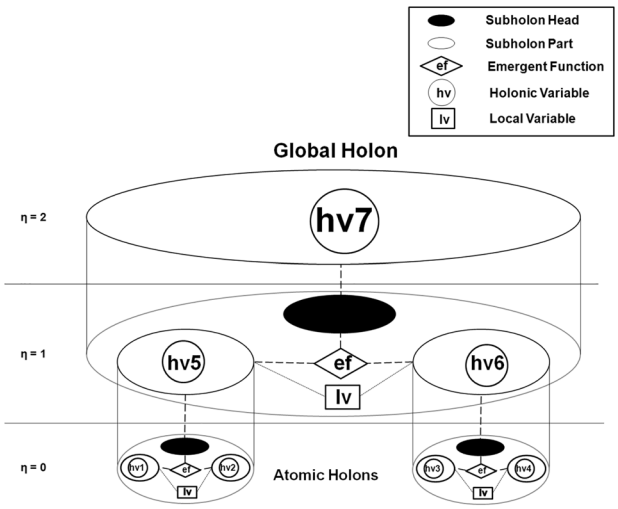


Fig. 1 HCOP Basic Diagram

internal, which consists of a set of variables associated with the specific problem domain. They are called *local variables,* which are linked to each other and may also be connected to the holonic variables of subholons members by intraholon relationships called *local constraints*.

HCOP may be viewed as a partition of coupled smaller *Optimization Problems* (OPs), one for each superholon. Although not independent from each other, they present such a low coupling level that enables some parallelism in their solution process. In addition, that partition makes it easier to tackle the complexity of the whole problem, which is modeled by simpler submodels. Each one of them may repeat itself recursively throughout the whole model. The OP is defined by its local variables along with the holonic variables associated with its subholons. Each OP is solved by a corresponding *Optimization Algorithm* (OA). Figure 1 illustrates an example of HCOP.

The proposed model doesn't consider the role *multipart*, which would be played by subholons shared by more than one superholon at the same time. HCOP model assumes the strong cohesion within each superholon. In this sense, the concept of multipart holon becomes inconsistent, for it would be impossible to partition the problem to solve it through integration of optimization algorithms already available. In fact, Fig. 2a shows an example of holonic modeling using holons multipart, with 4 superholons (holons 1, 2, 3 and 4) and 6 atomic holons (holons 5, 6, 7, 8, 9 and 10), where holon 7 is multipart. The existence of this type of holon indicates that there is a strong cohesion between superholons 2 and 3, which deviates from the previously expressed directive for the holonic modeling proposed in this work. Thus, the HCOP model is that shown in Fig. 2b, where superholons 2 and 3 are fused to a single superholon 2 ', showing the affinity between subholons 5 and 6, on the one hand, and subholon 8, on the other, with respect to subholon 7, which has now become only a holon part. This change makes it possible the existence of an OA associated with the superholon 2 '. Otherwise, there would be the development of OA 2 and OA 3, which would be completely dependent on each other within the multipart model. In other words, the model considers only disjoint holons as it will be seen in the next subsection.

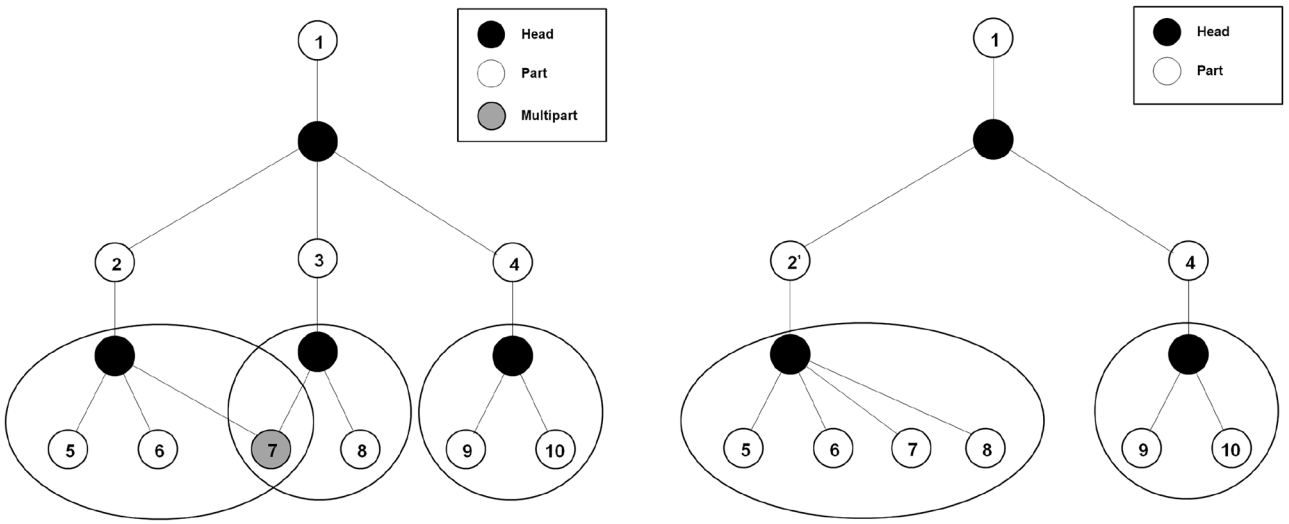


Fig. 2 Holonic Modeling Example

As already said, the key to a successful HCOP modeling is the holonification process, i.e. the creation of the holons, which is a NP problem by itself. However, this task is accomplished by model designers, which take into account the particular features of each problem, mainly its constraints configuration. Thus, it is possible to locate where they are harder (intraholon coesion constraints) or where they are softer (interholons coupling constraints). Another guide to the holonification is the identification of available optimization algorithms, which are associated with each superholon.

4.2 Formalization

HCOP is formalized as a tuple $(H, V, Dv, X, Dx, E, U, F, O)$, where:

- $H = \{H_0, \dots, H_\eta, \dots H_{\eta_{max}}\}$ is a partition of the set $\mathbb{H} = \{h, \dots, h_i, \dots, h_n\}$ of all holons of the problem, formed by the equivalence classes in relation to belonging to the same echelon η. Therefore, by definition H satisfies the following conditions:
  - $H_\eta \neq \oslash, \forall \eta \in \mathbb{N}, \eta \leq \eta_{max}$
  - $\bigcup_{0 \leq \eta \leq \eta_{max}} H_\eta = \mathbb{H}$
  - $H_i \cap H_j = \oslash, i,j \in \mathbb{N}, i \neq j$

  where η_{max} is the highest echelon. The proposed model contains a set of agents $\mathbb{A} = \{a_1, \dots, a_i, \dots, a_n\}$, which consists of two kinds of agents:

  - *atom* agent, which is a conventional agent;
  - *head* agent, which is responsible for the internal coordination and external communication of each superholon.

  Each holon h_i maps to each of the agents a_i of $\mathbb{A}$, such that $|\mathbb{H}| = |\mathbb{A}|$. Thus, a holon may be *atomic* (contains only an *atom* agent) or a *superholon* (contains a *head* agent only and others subholons). On the other hand, $H_\eta = \{h_{\eta 1}, \dots, h_{\eta i}, ..., h_{\eta N_\eta}\}$ is the set of holons of the echelon η, where each holon $h_{\eta i}$ is a subholon of a superholon of $H_{\eta+1}$ for $\eta < \eta_{max}$ and N_η represents the number of holons in the echelon η. Thus, H_0 is the set of atomic holons h_{0i} of the fundamental echelon ($\eta = 0$) and $H_{\eta_{max}} = \{h_{\eta_{max}1}\}$ contains a single holon called *global*, which contains the whole holarchy. More formally each holon may be classified into two cases:

  - $h_{\eta i}$ is an *atomic* holon, which is the singleton formed by the *atom* agent;
  - $h_{\eta i} = \{K_{\eta i}, B_{\eta i}\}$ is a superholon, where:

    $K_{\eta i} = h_{\eta-1 j}$ is the *head* holon of holon $h_{\eta i}$, which is the singleton formed by the *head* agent;
    $B_{\eta i} = \{h_{\eta-1 \alpha_1}, h_{\eta-1 \alpha_2}, \dots\} \neq \oslash$ is called the *body* of the holon $h_{\eta i}$. Each holon $h_{\eta-1 \alpha}$ is called *part* or subholon of the holon $h_{\eta i}$.

  Each holon also satisfies the following conditions:

$$h_{\eta i} \neq \oslash, \forall i \in \mathbb{N},\ 1 \leq i \leq N_\eta \tag{2}$$

$$h_{\eta i} \cap h_{\eta j} = \oslash,\ i,j \in \mathbb{N},\ i \neq j \tag{3}$$

  The second condition imposes that holons of the same echelon η are disjoint, and hence do not contain multipart subholons, according to the assumption of this work (see Sect. 4.1). The element H of the tuple defines the holonic structure of the problem. The holonic relations 4 and 5 may be used to make it easier to navigate between the holons within this structure.

$$headOf_\eta = \{(h_{\eta i}, h_{\eta-1 j}) \in H'_\eta \times H_{\eta-1} \ :\ h_{\eta-1 j} = K_{\eta i}\} \tag{4}$$

$$partOf_\eta = \{(h_{\eta i}, h_{\eta-1 j}) \in H'_\eta \times H_{\eta-1} \ :\ h_{\eta-1 j} \in B_{\eta i}\} \tag{5}$$

  where $\eta \in \mathbb{N}, \eta > 0, H'_\eta \subset H_\eta \mid H'_\eta = \bigcup_i B_{\eta+1 i}\ para\ \eta < \eta_{max}\ ou\ H'_\eta = H_\eta\ para\ \eta = \eta_{max}$.

- $V = \{v_{0\,1}, ..., v_{\eta i}, ..., v_{\eta_{max}1}\}$ is the set of *holonic variables*, where each variable $v_{\eta i}$ is associated with a holon $h_{\eta i}$ of the echelon η (a holonic variable per holon). Each holonic variable may be classified into the following two cases:
  - a *decision variable*, if the associated holon is *atomic*. It is an independent variable of the whole problem;
  - a *emergent variable*, if the associated holon is a superholon. It is dependent on the holonic variables of the constituent subholons.

  These variables take part in the holonic constraints, but they may also participate in local constraints within their corresponding superholon. These constraints are presented later.
- $Dv = \{Dv_{0\,1}, ..., Dv_{\eta i}, ..., Dv_{\eta_{max}1}\}$ is the set of domains, where each $Dv_{\eta i}$ is a discrete and finite set of elements which may be assigned to the respective holonic variable $v_{\eta i}$;
- $X = \{x_{0\,1\,1}, ..., x_{\eta i j}, ..., x_{\eta_{max}-1\, i_{max} j_{max}}\}$ is the set of local variables, where each variable $x_{\eta i j_i}$ is one of the internal variables for the superholon $h_{\eta+1 i}$. They are optional, but there may be several ones per holon, which must be a superholon. These variables take part only on local constraints, along with the holonic variables associated with the subholons of the same superholon.

- $Dx = \{Dx_{0\,1\,1}, ..., Dx_{\eta\, i\, j_i}, ..., Dx_{\eta_{max-1} i_{max} j_{imax}}\}$ is the set of domains, where each $Dx_{\eta\, i\, j_i}$ is a discrete and finite set of elements which may be assigned to the respective local variable $x_{\eta\, i\, j_i}$;
- $E = \{E_{1\,1}, ..., E_{\eta\, i}, ..., E_{\eta_{max}\,1}\}$ is the set of emergence functions, where each function $E_{\eta\, i}$ is associated with the respective holon $h_{\eta\, i}$, where $\eta > 0$ (one per superholon), so that:

$$E_{\eta\, i} : Dv_{\eta-1\alpha_1} \times \ldots \times Dv_{\eta-1\alpha_{B_{\eta i}}} \rightarrow Dv_{\eta\, i} \tag{6}$$

where the function domain is the cartesian product of the holonic variables domains associated with the subholons of $h_{\eta\, i}$ and the function image is the domain of the respective emergent variable $v_{\eta\, i}$. The emergence function $E_{\eta\, i}$ allows the definition of an n-ary constraint $c_{\eta\, i}$ which connects the holonic variables corresponding to each of its subholons with the holonic variable of the superholon. As it can be seen in the Eq. 7, the constraint $c_{\eta\, i}$ is a hard one, i.e., it is satisfied or violated (see Sect. 3.2). It represents the aggregation force between the body of the superholon and its head, which in turn connects with the emergent variable of the same superholon through a granularity filter. For this reason, it is called *Emergence Constraint*

$$c_{\eta\, i} = constraint\left(v_{\eta\, i} == E_{\eta\, i}\left(v_{\eta-1\alpha_1}, \ldots, v_{\eta-1\alpha_{B_{\eta i}}}\right)\right) \tag{7}$$

The emergence function can be seen as a composite function, which is broken up into the following ones:

– *Aggregation Function*:

$$Ag_{\eta\, i} : Dv_{\eta\text{-}1\alpha_1} \times \ldots \times Dv_{\eta-1\alpha_{B_{\eta i}}} \rightarrow Dv_{\eta-1\,0} \tag{8}$$

– *Granularity Function*:

$$Gr_{\eta\, i} : Dv_{\eta-1\,0} \rightarrow Dv_{\eta\, i} \tag{9}$$

Due to the head role in the superholon, its domain $Dv_{\eta-1\,0}$ must reflect the behavior of the subholons which it coordinates. Thus:

$$Dv_{\eta-1\,0} = Dv_{\eta\text{-}1\alpha_1} \times \ldots \times Dv_{\eta-1\alpha_{B_{\eta i}}}$$

However, the superholon emergent variable $v_{\eta\, i}$ has another level of granularity associated with the context of echelon η, which is different from that of the lower adjacent echelon $\eta - 1$. Therefore, its domain $Dv_{\eta\, i}$ must be smaller than the special domain $Dv_{\eta-1\,0}$ of the head holon. This attenuation is performed by the granularity function, which behaves as in the *Granularity Theory* (see Sect. 3.1.3). In fact, it is an articulation mapping between 2 consecutive echelons. Thus, the *Emergence Constraint* may be viewed as the concatenation of the *Aggregation Constraint* and the *Granularity Constraint* (see Fig. 3).

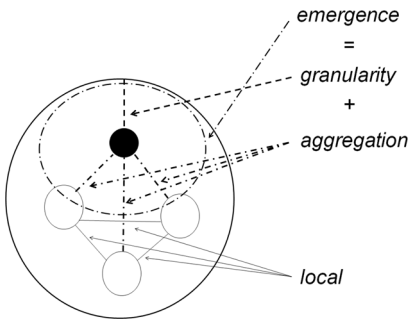


Fig. 3 Superholon Constraints

- $U = \{u_{0\,1}, \ldots, u_{\eta\, i}, ..., u_{\eta_{max}\,1}\}$ is the set of valued unary functions, one for each holon, where $u_{\eta\, i}$ stands for the *utility* of the holon $h_{\eta\, i}$ and depends on its holonic variable $v_{\eta\, i}$. It requires an aggregation operator *sum* which must be associative, commutative and monotonic and it will be important for the definition of the *objective function*, which determines the goal to be pursued in the the optimization process, as it will be seen next.

$$u_{\eta\, i} : Dv_{\eta\, i} \rightarrow \mathbb{N} \tag{10}$$

- $F = \{F_1, \ldots, F_\eta, ..., F_{\eta_{max}}\}$ is the set of $F_\eta = \{f_{\eta\,1}, \ldots, f_{\eta\, i}, ..., f_{\eta\, N_\eta}\}$, one for each echelon $\eta > 0$, where $f_{\eta\, i}$ is an n-ary valued constraint among the holonic variables $v_{\eta-1\,\alpha_{ij}}$ and/or of the local variables $x_{\eta-1\, i\, \beta_{ij}}$ of the superholon $h_{\eta\, i}$, where N_η is the number of holons in the echelon η. These valued constraints are functions which are represented in Eq. 11, whose domain is the cartesian product of the holonic variables$\{v_{\eta-1\alpha_1}, \ldots, v_{\eta-1\alpha_{N_{\eta ij}}}\}$, which is a subset of the subholons of the superholon $h_{\eta\, i}$ and its local variables $\{x_{\eta-1\, i\, \beta_1}, \ldots, x_{\eta-1\, i\, \beta_{O_{\eta ij}}}\}$. They define local constraints of the superholon, unlike the holonic emergence constraint, which is an unvalued n-ary one (see Eq. 7).

$$f_{\eta\, i} : Dv_{\eta-1\alpha_1} \times \ldots \times Dv_{\eta-1\alpha_{N_{\eta ij}}} \times Dx_{\eta-1\, i\, \beta_1} \times \ldots \times Dx_{\eta-1\, i\, \beta_{O_{\eta ij}}} \rightarrow \mathbb{N} \tag{11}$$

The local constraint function $f_{\eta\, i}$ represents an utility generated by the local constraint integration inside the superholon $h_{\eta\, i}$ among its subholons and its local variables. Thus, the utility $u_{\eta\, i}$ of a superholon $h_{\eta\, i}$ may be defined recursively, as it is the feature of the holonic model, using the utilities of its subholons and its local constraint function (Eq. 12). Since an atomic holon has no local constraint function unlike a superholon, it has only a predefined utility $u_{\eta\, i}$.

$$u_{\eta\, i} = \sum_{j=1}^{M_{\eta i}} u_{\eta-1\, j} + f_{\eta\, i} \tag{12}$$

where $M_{\eta i}$ is the number of subholons of the superholon $h_{\eta i}$.

- $O = \{OA_{1\,1}, ..., OA_{\eta i}, ..., OA_{\eta_{max}1}\}$ is the set of Optimization Algorithms, where each $OA_{\eta i}$ solves and therefore optimizes the problem defined by the $f_{\eta i}$ associated with the respective superholon $h_{\eta i}$, where $\eta > 0$. In addition to being correct and complete, these algorithms must respect the input and output protocols defined by Eqs. 13 and 14, respectively, as shown next. Inputs:
  - $c_{\eta i}$: emergence constraint definided by the emergence function $E_{\eta i}$ associated with the superholon $h_{\eta i}$ (Eq. 7). It represents the holonic constraint for this superholon;
  - $u_{\eta-1\,j\,l}$: utility of the subholon $h_{\eta-1\,j}$ associated with the value $d_{\eta-1\,j\,l} \in Dv_{\eta-1\,j}$ of the holonic variable $v_{\eta-1\,j}$. If $h_{\eta-1\,j}$ is:

    a superholon also, it is obtained from $OA_{\eta-1\,j}$ recursively;
    a atomic holon, it is equal to its utility known a priori;
  - $d_{\eta\,i\,ind}$: is the element ind of the set $Dv_{\eta i}$, i.e., $d_{\eta\,i\,ind} \in Dv_{\eta i}$;

  Outputs:

  - $u_{\eta\,i\,ind}$: optimal utility of the superholon $h_{\eta i}$ associated with the value $d_{\eta\,i\,ind} \in Dv_{\eta\,i}$ of the emergent variable $v_{\eta i}$ of the same superholon;
  - $v^*_{\eta-1\,j}$: represents the partial solution of the holonic variable $v_{\eta-1\,j}$ associated with the value $d_{\eta\,i\,ind} \in Dv_{\eta\,i}$ of the emergent variable $v_{\eta i}$ of the superholon $h_{\eta i}$;
  - $x^*_{\eta-1\,i\,k}$: represents the partial solution of the local variable $x_{\eta-1\,i\,k}$ of the superholon $h_{\eta i}$ associated with the value $d_{\eta\,i\,ind} \in Dv_{\eta i}$ of the emergent variable $v_{\eta\,i}$ of the same superholon;

$$OA_{\eta i} \leftarrow (c_{\eta i},\, u_{\eta-1\,j\,l},\, \ldots,\, d_{\eta\,i\,ind}) \tag{13}$$

$$\left(u_{\eta\,i\,ind},\, (v^*_{\eta-1\,j}, \ldots, v^*_{\eta-1\,\alpha_{max}}),\, (x^*_{\eta-1\,i\,k}, \ldots, x^*_{\eta-1\,i\,\beta_{max}})\right) \leftarrow OA_{\eta i} \tag{14}$$

The goal of HCOP is to meet a complete instantiation V^* for all holonic variables $v_{\eta i}$ and also a complete instantiation X^* for all local variables $x_{\eta k}$ in order to maximize the *objective function* $\mathcal{OF}$, which corresponds to the utility $u_{\eta_{max}1}$ of the global holon $h_{\eta_{max}1}$, as defined in the Eq. 15.

$$\mathcal{OF} = u_{\eta_{max}\,1} \tag{15}$$

Taking into account the recursive definition of utility for each superholon in Eq. 12, it is possible to conclude that:

$$\mathcal{OF} = \sum_{\eta=0}^{\eta_{max}-1\ atomic\ holons} \sum_{i} u_{\eta i} + \sum_{\eta=1}^{\eta_{max}\ superholons} \sum_{i} f_{\eta i} \tag{16}$$

In the first term the sum over the echelons starts at the fundamental one (all holons are atomic) and goes through all echelons, since they may contain other atomic holons, but the highest one (global superholon). The sum of the second term ignores only the fundamental echelon. On the other hand, it is important to point out that the condition applied to aggregation [1] operator "sum" for local constraint functions, also holds for emergence functions and, consequently, for aggregation and granularity functions as well. Except for the condition presented in the utilities item U of the current section, there is no limitation with respect to the type of constraint function, which may be even nonlinear.

Figure 3 illustrates the constraints of the model.

4.3 Domain Size Magnitude Order Invariance

Taking into account the aggregation constraint, the head holonic variable domain is built by the cartesian product of the domains of the holonic variables associated with the subholons of the same superholon (see Eq. 8). Thus, if the granularity constraint did not exist, the higher the echelon the larger the domains of its holonic variables would become. In same way, the higher the echelon the more difficult the associated OP would be, since its complexity depends on the number of variables, the number of constraints and also the size of the variable domains. On the other hand, the emergent variable of a superholon should have the same order of magnitude as the holonic variables of its subholons, since it is viewed by others holons in its echelon in the same way as its subholons see each other. Thus, it is reasonable to assume that each superholon, regardless of its echelon, should, in principle, have the same difficulty level relative to the solution of its Optimization Problem (*OP*). For this reason, the granularity constraint adjusts the size of the head holonic variable domain in order to obtain an acceptable domain for the emergent variable domain.

Therefore, in this work it is adopted a common sense rule, which considers that the order of magnitude of the domain is invariant for the all holonic variables, except for the decision variables, which are associated with atomic holons and have domains known a priori. In fact, that invariant is represented by k, which is the nearest integer associated with the domain magnitude of each holonic variable (Eq. 17).

$$k = \lfloor log(|\, Dv_{\eta i}\,|) \rceil \tag{17}$$

[1] Do not confuse with the aggregation function, which has another meaning in this model.

where $k \in \mathbb{N}$ and $v_{\eta i}$ is any emergent variable belonging to a superholon.

4.4 Relation Between DCOP and HCOP

The Holonic Constraint Optimization Problem (HCOP) possesses specific characteristics that make it different from a DCOP, among other things, due to the existence of holons instead of simple agents. However, taking into account the properties of HCOP, it is easy to realize that they contain all the properties of DCOP for Complex Local Problems (see Sect. 3.2.2), as well as specific properties as Emergence Functions and Optimization Algorithms. Thus, it is possible to conclude that the class of problems HCOP is a subclass of those comprising all DCOPs. Therefore, any HCOP is also a DCOP with Complex Local Problems.

Most algorithms for DCOP admit that each agent controls only one variable. This assumption has led to the two reformulations which were proposed by (Yokoo and Hirayama 1998), by which any DCOP with complex local problems (i.e., multiple variables in each agent) can be transformed to one with just one variable per agent (Burke 2008): *Compilation*, which defines a variable whose domain is the set of solutions to the local original problem for each agent, or *Decomposition*, which creates an unique agent to manage it for each variable in each local problem. In the case of HCOP, the compilation method fits with the holonic organization naturally, for the head holon has a special role in this transformation. In fact, the domain of its holonic variable is the cartesian product of the domains of the holonic variables of the subholons associated with the respective superholon. Thus, its domain may represent the set of solutions to the Optimization Problem (*OP*) associated with its superholon. Similarly the compilation method was used in the transformation from DCOP with Complex Local Problems into basic DCOP (Burke 2008). It is presented formally for HCOP next:

- for each superholon $h_{\eta i}$ the domain $Dv_{\eta-1\,w}$ of the holonic variable $v_{\eta-1\,w}$ of its holon head, i.e., $h_{\eta-1\,w} = headOf_{\eta}(h_{\eta i})$, represents the set of all partial solutions of the internal problem of this superholon ($OP_{\eta i}$), regarding the holonic variables $v_{\eta-1\,\alpha}$ of its subholons and the local variables $x_{\eta-1\,i\,\beta}$ of the same superholon. Each partial solution is associated with a respective value $d_{\eta i\,ind} \in Dv_{\eta i}$ of the emergent variable $v_{\eta i}$ of the superholon $h_{\eta i}$. This solution is obtained by calling the optimization algorithm $OA_{\eta i}$ associated with this superholon, which respects the input and output protocols defined by Eqs. 13 and 14 of Sect. 4.2. As already mentioned in the same section, the holonic variable $v_{\eta-1\,w}$ of the holon head is directly linked to the emergent variable of its superholon via the granularity constraint. Thus, both have the same meaning, but different degrees of granularity, what is defined by the granularity function, which maps the domain $Dv_{\eta-1\,w}$ of the holon head to the domain $Dv_{\eta i}$ of its superholon. Therefore, each partial solution in the domain $Dv_{\eta-1\,w}$ relates to a value $d_{\eta i\,ind} \in Dv_{\eta i}$ of the superholon via that function, which is not bijective;
- for each holon $h_{\eta i}$ there is a valued unary function $u_{\eta i}$ called *utility* which is dependent on the assignment $d_{\eta i\,ind} \in Dv_{\eta i}$ of its holonic variable $v_{\eta i}$ (see Sect. 4.2);
- for each superholon $h_{\eta i}$ there is a $f_{\eta i}$ which represents the local constraints associated with the local problem of this superholon and depends on the holonic variables $v_{\eta-1\,\alpha_{ij}}$ of its subholons and its local variables $x_{\eta-1\,i\,\beta_{ij}}$ (see Sect. 4.2). Both the utility of this superholon $h_{\eta i}$, which is defined by Eq. 12, and the partial solutions of the holonic variables of its subholons as well as the local variables of the same superholon, are obtained by invoking the respective $OA_{\eta i}$ associated with $f_{\eta i}$.

Figure 4 illustrates the transformation process from HCOP into DCOP by the compilation method. In this example HCOP has three echelons, 3 superholons, two of them with a local variable and the global holon with 2 local variables (each one with its respective head holon), 4 atomic holons in the fundamental echelon and an atomic holon in the intermediate echelon. In sequence Fig. 4b presents the resulting DCOP with Complex Local Problems, where the 8 holons are replaced with conventional agents (one for each head holon and atomic holon, as already mentioned in Sect. 4.2) and the holonic variables of the 3 superholons and the 5 atomic holons become public variables. The DCOP local variables came from the HCOP local variables, but also from the head holons holonic variables, since the latter function as an internal link between the public variables, which have constraints with other agents. Finally, Fig. 4c shows the resulting basic DCOP, which has 8 agents, each one with a variable. They are associated with the 8 holonic variables, which are 3 emergent variables, one for each superholon, and 5 decision variables, one for each atomic holon.

There are several algorithms developed to solve DCOP with the limitation that they assume that each agent has only a local variable. By transforming HCOP into DCOP, the first can be solved by any of the complete algorithms for DCOP, since an interface is provided, allowing the integration with available OAs. This is exactly what is done in the next section with the proposal of a meta-algorithm to solve HCOP, using a DCOP algorithm. They allow to solve generic HCOP, even for complex local problems like the supply chain integrated planning. Thus, the two algorithms

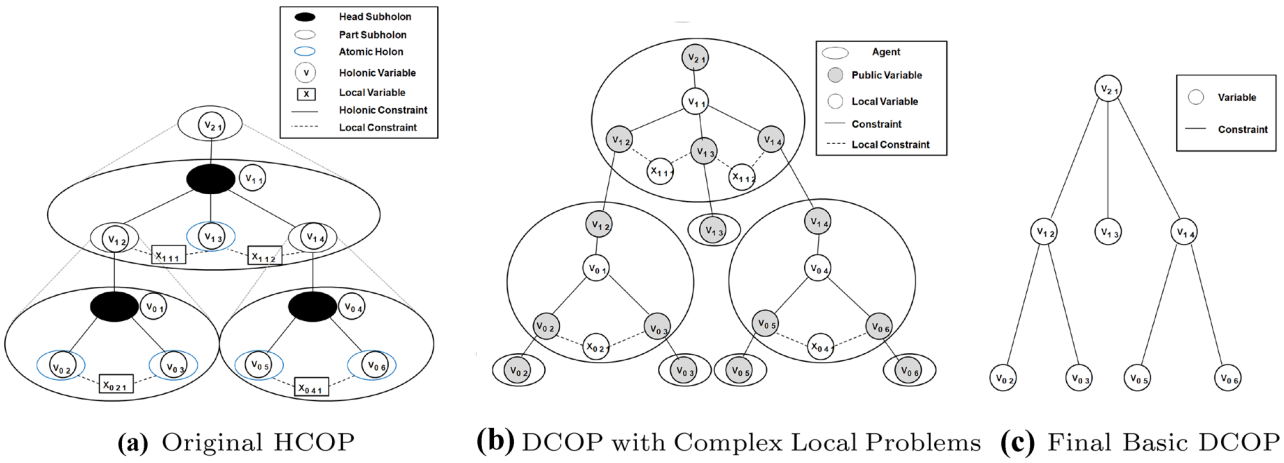


Fig. 4 Transformation from HCOP into DCOP

for DCOP presented in Sect. 3.2.3 are used to develop the corresponding meta-algoritms for HCOP.

4.5 Holonic Constraint Optimization Meta-Algorithms (HCOMAs)

HCOP can be seen as a large distributed optimization problem, which is modeled as a holarchy where each superholon uses an Optimization Algorithm OP associated with its corresponding Optimization Problem OP. Therefore, to take advantage of that feature, it is more appropriate a meta-algorithm for HCOP, rather than a single DCOP algorithm. It allows to embed a DCOP algorithm into the more abstract framework, which integrates various *OA*s, taking into account also the holonic properties of the model. This meta-algorithm is called Holonic Constraint Optimization Meta-Algorithm (HCOMA). Section 3.2.3 described the DCOP algorithms which are used in this work: ADOPT and DPOP. They represent important categories within this area of academic research. Both algorithms have its agents prioritized into a DFS tree, whose structure forms a connected graph, i.e., a graph without cycles. This feature fits well with HCOP, since its holonic organization does not include multipart holons (see Sect. 4.1).

4.5.1 Adaptation of a DCOP Algorithm to HCOMA

In the meta-algorithm HCOMA the specific DCOP algorithm considers holons as conventional agents just as it is shown in Sect. 4.2, i.e., a set comprising two kinds of agents: *atom* agent and *head* agent. DCOP algorithm exchanges the messages UTIL (or COST) and VALUE with them. The atomic holon acts as a conventional agent, while head holon takes care of the holonic constraints and local constraints via the *OA*s. Then, when a head agent receives messages from the neighbor agents, HCOMA takes control, invoking the respective *OA*. It forwards the UTIL (or COST) messages received from lower priority agents to the *OA* (see Sect. 3.2.3). Later when HCOMA receives the optimal utility from it, as well as the partial solution composed of the values of the subholons holonic variables and the local variables associated with the respective superholon, it forwards this information via the UTIL (or COST) message to the highest priority agent and the VALUE messages to lower priority agents. These specific responsibilities of HCOMA are performed by the head agent. Therefore, HCOMA messages relative to the interface with the *OA* are restricted to intra-holon interactions, while DCOP algorithm messages work in the inter-holon scope, connecting neighboring echelons. Thus, HCOMA works with the same mechanism of the original DCOP algorithm, so as maintaining the same computation operations cycle performed by each agent repeatedly (see Algorithm 1). Its main features are in step 2 of this cycle (*perform some computation*), which can be divided into three basic OA-dependent core activities. They are shown in Algorithm 2.

Algorithm 2 HCOMA Cycle

1: **compute** the holonic and local constraints via OA;
2: **assign** the value of the holonic and local variables via VALUE messages by the respective head, which receives from OA;
3: **calculate** using OA the associated utility, which is sent by the respective head via the appropriate message (UTIL or COST);
4: **return**

Under the assumption made in the definition of HCOP that each *OA* is correct and complete, if the DCOP algorithm used is correct and complete, then the resulting HCOMA will also be. As to the HCOMA complexity in time and space, it is the worst case between DCOP algorithm

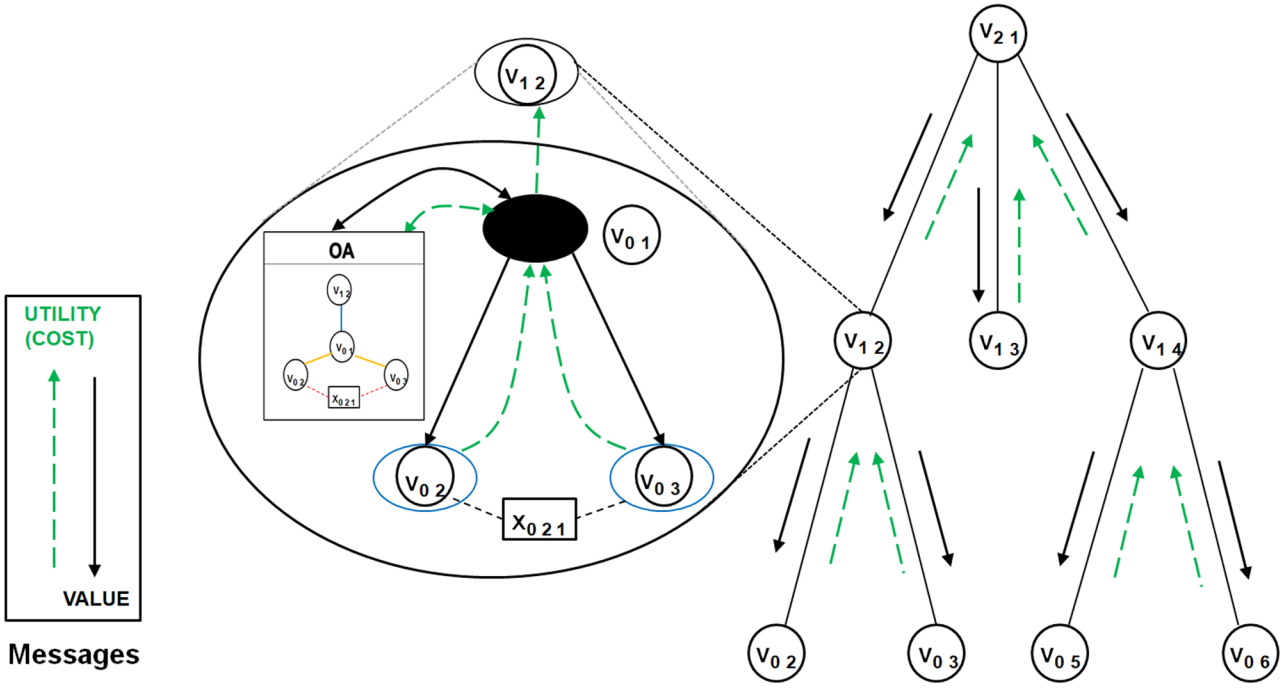


Fig. 5 Relationship between HCOMA and DCOP Algorithm

complexity and that of *OA*. However, the complexity associated with quantities related to the distributed nature, such as the number or size of messages, it is inherited from DCOP algorithm, since each *OA* is centralized by definition. Figure 5 illustrates HCOMA mechanism using DCOP algorithm.

In this work HCOMA is applied to extend both chosen DCOP algorithms in order to solve HCOP. The discussion about the comparison between these two meta-algorithms is presented next.

4.5.2 Comparison Between HCOMAs

The meta-algorithm was implemented in two versions from two traditional DCOP algorithms, which belong to categories, using different solution strategies: ADOPT-based and DPOP-based HCOMAs. They were compared through experiments in order to evaluate their performance as well as to analyse their differences and features in Marcellino (2013). For this purpose it was used a simple DisCSP (Distributed Constraint Satisfaction Problem), called Max-DisCSP, which was adapted to become DCOP by minimizing the number of violated constraints. In turn, it was reformulated to become HCOP by introducing holonic properties in instances of the problem. These properties are numbers of holons, variables domain size, number of local variables and number of echelons. It was possible to realize a greater predominance of DPOP-based HCOMA performance relative to the other meta-algorithm in all of these parameters, which were controlled throughout the experiments. DPOP-based meta-algorithm was faster in all the cases, especially for larger numbers of holons.

ADOPT has the advantage of synchronous processing, which could result in a greater parallelism when dealing with several holons. However, it presents a behavior which mixes the various echelons of the holarchy during its solution process. Thus, each OA, in general, is called multiple times, so that it may be probably an important reason for its poor performance to solve HCOP. On the other hand, DPOP seems more suited to the holonic organization features, for it is based on dynamic programming, which is also recursive. Furthermore, it invokes each OA fewer times, which suggests a higher efficiency. The experiments results pointed out that DPOP-based HCOMA is a better choice to solve HCOP. Therefore, it is also employed in the experiments associated with the more complex HCOP which is treated in this work (see Sect. 2).

5 Modeling Integrated Supply Chain Planning as HCOP

As mentioned before, the integrated planning of the process industry supply chain is a candidate problem to be modeled as HCOP, specifically the one associated with the oil industry, which is described briefly in the first subsection. Its organization model is discussed next, while time modeling,

the chosen performance metrics and its objective function are presented in the following subsections.

5.1 Problem Description

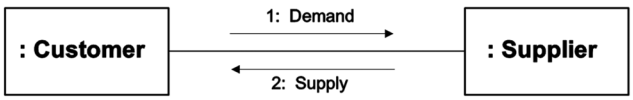
The problem is about the oil industry supply chain, which starts at the crude oil extraction and finishes when its derivative products are delivered to distribution companies, which are considered here as final customers. The supply of crude oil and its derivatives must be performed preferentially by the oil company itself, which may be a single verticalized oil enterprise or a set of cooperating companies of the oil business[2]. Henceforth it will be called extended enterprise (EE) the general situation that comprises both cases. If necessary or eligible, the EE can purchase from the spot market (SM), which satisfies any extra demands of crude oil and its derivatives at higher prices. In the same way, SM can buy any exceeding inventories of those items at lower prices. The EE operation area is spread geographically, and this physical space is visualized as a partition of regions, which are in turn grouped into continents, which finally are gathered into a global area. A region is divided into trading areas or oil extraction areas. A refinery, or distribution terminal, is responsible for each of the first ones, which serves specific final customers (distribution companies), while each of the latter ones is composed of oil extraction platforms, that yield a certain type of crude oil at a rate that may be considered constant for the time scale of this work. Refineries, terminals and oil extraction platforms are called henceforth facility bases, or just bases. The platforms have no capacity to stock the oil which they extract, which must be drained via pipelines or vessels to another base. All areas are connected by transportation modals, which are of two types: oil pipelines and vessels. In addition, each area has a logistics entity, which is responsible for transportation of crude oil (petroleum) and its derivatives. All crude oil types (raw materials) and their derivatives (final products) will hereinafter be referred to simply as products. All transportation is carried out through a set of arcs, which connect two entities of the chain and have an associated maximum transport capacity. This entity can be a base (platform, refinery or terminal), a region, a continent or SM. Because the different entities in the chain are connected through a transport network, they can cooperate with each other in order to supply the different trading areas, rather than concerning themselves with their own areas. Without loss of generality, this work considers as premise that the pipeline transport is used within each region and between them, while the vessel transport interconnects regions within a same continent, as well as continents to each other and to SM.

[2] The scenario of a supply chain formed by competing companies is not considered in the problem modeled in this work.

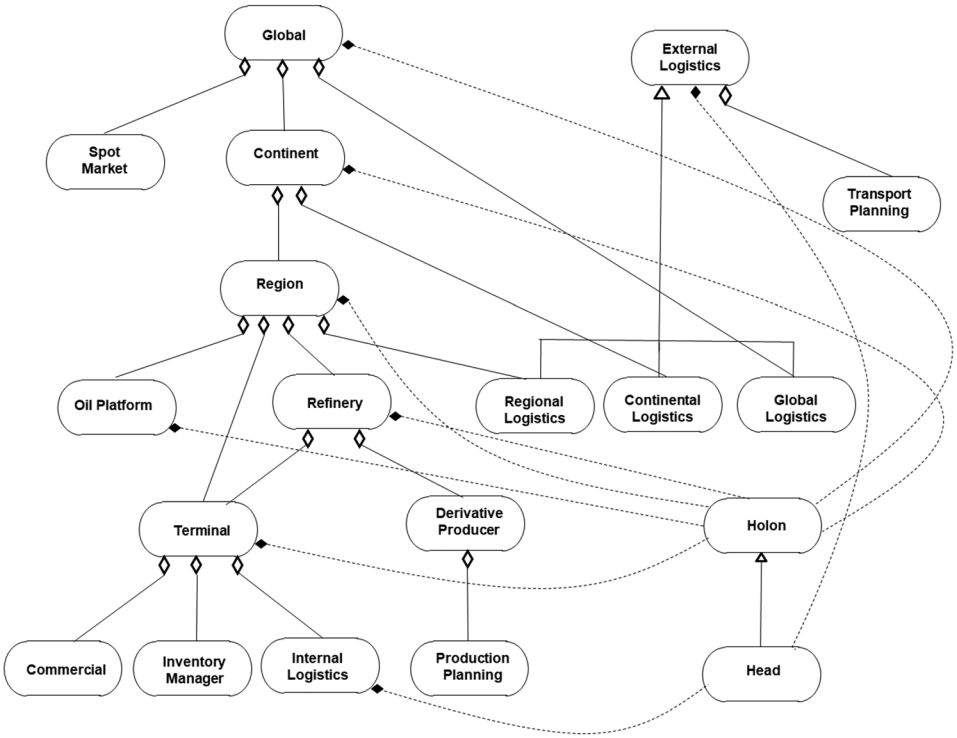
5.2 Organization Model (Internal and Holonic Model)

Since the problem is modeled as HCOP, which is a HMAS, its organizational model can be represented in two dimensions: *holonic organization*, which is common to all holonic systems and an *internal organization*, which is specific to each problem domain (see Sect. 3.1.2). Thus, each agent can have a role in the holonic organization and another role in the internal organization. The holonic organization is composed primarily of the roles head and part, while the internal organization consists of a greater variety of roles, which are associated with physical entities, services and supply chain functions (see Sect. 3.1.2). In addition to the specific roles of internal organization, two generic roles are shared by several of them. The first is the role *supplier*, which refers to any entity that provides products to another entity in the chain, such as a *Refinery*. The other is the role *customer*, which is played by any other entity which needs these products, such as a *Terminal*. The difference between availabilty and need of a product represents the holonic variable of the corresponding holon.

Figure 6 illustrates the general organization model, which represents the relationships between the roles of the internal organization and between the holonic roles using the modeling language Moise+ (Hubner et al. 2002) and the UML Collaboration Diagram (OMG 2015). For example, a *Refinery* has the roles *supplier* and *customer* at the same time. Thus, it is modeled as a *Terminal* with a *Derivative Producer*, since both have a *Commercial* and an *Inventory Manager* areas, in addition to a *Logistics*. The latter is the head of the respective superholon at all levels of the chain, i.e., the *Internal Logistics* for holon *Refinery* or *Terminal*, the *Regional Logistics* for holon *Region* and so on, up to the *Global Logistics* for global holon. Each *Logistics* is responsible for the balance between supply and demand of products between *supplier* and *customer*, and also manages the *Transportation Planning*. Regarding the role *Production Planning*, it is played in the *Refinery* by a software or a team of human experts who support the role *Derivatives Producer*. The role *Commercial*, in turn, is responsible for forecasting the total demand for each derivative in the corresponding commercial area. A *Refinery* can produce multiple derivatives, and it does so according to different production plans, which are characterized by processing a definite quantity of a particular type of crude oil and producing a certain quantity of each resulting derivative. In addition, each *Refinery* or *Terminal* is responsible for the management of its inventories of each product. Thus, the decision variables of the model in



(a) Customer-Supplier Collaboration Diagram



(b) Internal and Holonic Organizational Model

Fig. 6 General Problem Organizational Model

the fundamental echelon ($\eta = 0$) are the *Production Plan* adopted by each *Refinery* during each period of time, and the inventory of each product monitored by the *Inventory Manager* of each *Refinery* or *Terminal* at the end of each period of time.

The holonic agents architecture allows that a complicated process can be broken down into smaller processes, which are in turn decomposed until each of them can be handled by a single agent. The agent head acts as a mediator, which supports communication and control for each level of the decomposition process. Although it would be possible a model with more granularity with agents representing the operational units of the refinery, in this work this descent ends in the *Derivatives Producer* of a *Refinery,* which provides a discrete set of production plans, which are in turn obtained from the correspondent *Production Planning*.

The higher the echelon, the larger the spatial scope of the corresponding holons. Similarly the higher the echelon, the longer the period of time considered by the *Logistics* head in its planning. Thus, the holons result from a spatial and temporal aggregation along growing abstract levels. On the other hand, the problem comprises different OPs: the production optimization of each *Refinery* and the transport optimization of the *Logistics* in each superholon. These latter are associated with growing echelons, and gradually embody larger geographic areas and longer planning time periods. In fact, the *Internal Logistics* is responsible only for a *Refinery* or *Terminal* on a day-by-day basis, the *Regional Logistics* takes care of an entire region with the week as the time unit, and so on up to the *Global Logistics* which focuses on the whole EE with a planning horizon of semesters or even years.

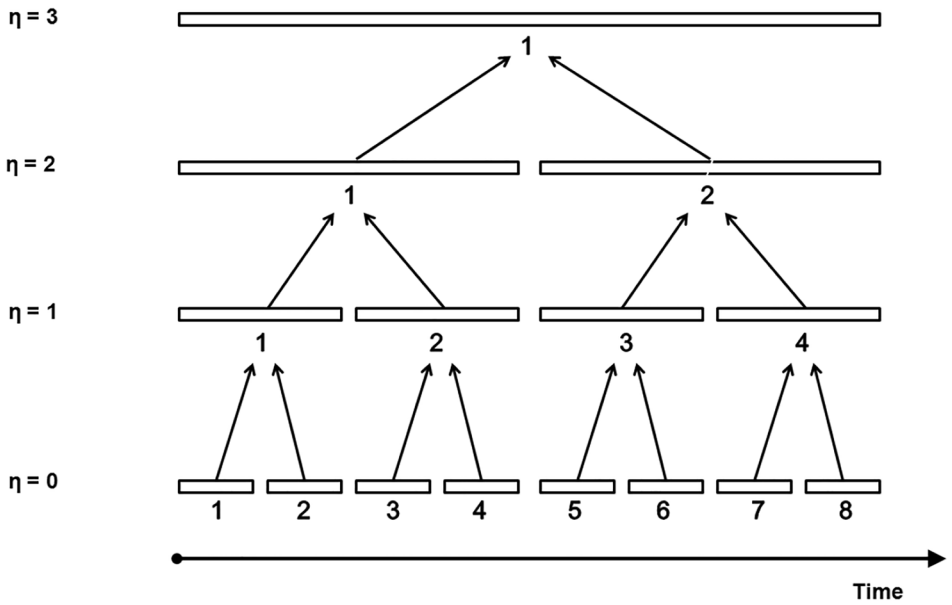


Fig. 7 Temporal Aggregation Example

5.3 Temporal Aggregation

Besides spacial aggregation, the model includes temporal aggregation, treating space and time in the same way. It proposes a multi-period approach (see Sect. 2.2), where the supply chain system evolves over a given time horizon Γ, which is divided in periods P_η, being T_η the total number of periods in the echelon η. The period P_η and the respective $T_\eta$depend on the associated η, but the time horizon Γ is the same for the whole system. Thus, the period P_η increases with the growth of η, while T_η decrease with the growth of η, according to the following invariant relationship:

$$P_\eta \cdot T_\eta = \Gamma \tag{18}$$

Thus, the higher the echelon the longer the planning period. Figure 7 illustrates an example of temporal aggregation.

It is important to look at the past and future relationship in the model, which appears at continental and global echelons. It results from temporal physical constraints associated with the products transport between the entities of the chain. Thus, it is assumed that the products take about a period of time to be transported between different regions within a continent (in the continent echelon), while they take about two periods of time to travel between two continents, or between one continent and SM (in the global echelon). As for the regional echelon, it is assumed that two different bases in the same region are close enough that products can be transported between them within a single period of time.

5.4 Performance Metrics and Objetive Function

Generically, the goal of the solution to the proposed problem turn out to choose a decision strategy such that the performance metric chosen is the best possible. In this work, the performance metrics used are the total profit of EE during the considered time horizon Γ, and the suitable service level to ensure the meeting of the customer demands. In principle, the simultaneous existence of two metrics, one quantitative and other qualitative, could lead the problem to a multi-criteria approach. However, due to the flexibility of SM, which, according to the problem assumptions, can buy or sell quantities of any product, which are large enough in relation to the EE demand and production values. Thus, its final customers are necessarily supplied. Therefore, the objective function of this problem is to maximize the total profit of EE, with customer satisfaction guaranteed automatically.

The total profit of EE can be expressed as:

$$\begin{aligned} Profit &= Revenue - Cost \\ Revenue &= OilRevenue + DerivativeRevenue \\ Cost &= OilCost + DerivativeCost + FreightCost \end{aligned} \tag{19}$$

where

OilRevenue results from the sale of surplus of oil extracted by EE to other oil companies by contracts and also to SM;

DerivativeRevenue results from the sale of all produced oil derivatives to the customers, the sale of surpluses to other continents by contracts and also to SM;

OilCost resulting from the oil extraction cost by EE, the oil purchase from other oil companies by contracts and also from SM;

DerivativeCost is the total production cost of all refineries, the derivatives purchase from other oil companies by contracts and from SM;

FreightCost results from the oil and derivatives transport, being the total of transfering products within each region, between different regions in the same continent, between different continents and from these to SM.

Table 1 Oil Types

Oil Type	Average API	Origin
Light	39	Imported
Medium	25	Internal
Heavy	20	Internal

It should be noted that it makes no sense to consider in the objective function a term associated with the financial storage cost within the holonic model. A holistic and integrated view of this model necessarily takes into account the loss of opportunity of using the stock in relation to the total profit accounted for by the objective function.

6 Experiments

6.1 Introduction

The tool FRODO (Leaute et al. 2009) is an open framework in Java for distributed combinatorial optimization, which was chosen for HCOP experiments in this work, because it contains built-in DCOP algorithms such as DPOP, what makes it easier to develop the meta-algorithms HCOMA. Since it uses XML files with a format which is a superset of XCSP 2.1, it is possible to represent DCOP instances. Thus, the meta-algorithms HCOMA are developed from the corresponding algorithms already available in FRODO, which was adapted to read and interpret HCOP instances as an extended XML file. In addition, FRODO was modified to consider all holonic constraints (see Sect. 4.2) and the oil supply chain integrated planning features. Thus, it was implemented the concepts of product vector space and time, as well as interfaces to the production and the logistic submodel. The latter was developed as a model MILP, using the IBM ILOG CPLEX Optimization Studio version 12.2 as modeling tool (IBM 2018). Due to the recursive nature of the logistic submodel, which is a HCOP feature, it was implemented a special interface between the respective logistic OA and FRODO, using the Java library FacadeOPL (Ferber 2012). This interface allowed the activation and recursive control of that OA from FRODO. The experiments were performed in a PC with a processor Intel Core i3 2.53 GHz and 4 GB RAM, whose results are presented in the following subsections.

Table 2 Oil Platforms

Platform	Oil Type	Extration Rate (M m3/ week)	Region
1	Medium	55	Region II
2	Medium	405	Region I
3	Medium	430	Region I
4	Medium	188	Region I
5	Heavy	262	Region I
6	Heavy	315	Region I
7	Medium	150	Region I

Table 3 Derivative Sale Prices

$ / M m3	lpg	gasoline	jet fuel	diesel	fuel oil
sale price	338	690	871	683	250

6.2 Case Study

The experiments with the HCOP model described in Sect. 5 used a case study based on historical data of oil company PETROBRAS. Although the information is not real due to business confidentiality, it is representative in relation to actual situations. The objective of this case study is to evaluate the feasibility and the advantages of that model of a typical oil supply chain integrated planning. The model integrated the optimization systems for production and logistics and the derived experiments allowed some comparisons between their results and those obtained by conventional approaches for that problem. The production submodel is based on the actual production applications for planning as well as scheduling, which are used in refineries of PETROBRAS. As for the logistic submodel, it was developed specifically for this case study, where it must be executed recursively. However, the latter was based on logistic planning and scheduling systems used by the same company.

6.2.1 Description

The case study in question is simple, but representative. As for the scope and topology considered, it is a subset of the actual supply chain, containing three regions (region I, region II and region III) with 12 refineries and 12 terminals (6 land and 6 marine ones). The oil supply is shared and the refineries collaborate to satisfy the total market demand. The refineries are supplied by two oil groups, which are extracted from 7 oil fields (Table 2), and 1 group of imported oil. The oils were classified into groups according to their origin and API [3] (Table 1). All the relevant entities of the chain and a significant set of products are included. It contains 5 continents, one of them with three regions, and one

[3] The American Petroleum Institute gravity is a measure of how heavy or light a petroleum liquid is compared to water. It is used to compare the relative densities of oil types and the more it grows the higher its quality and price.

Table 4 Oil Derivative Contracts Data

Continent		lpg		gasoline		jet fuel		diesel		fuel oil	
M m3 - $		Qty	Price	Qty	Price	Qty	Price	Qty	Price	Qty	Price
Africa	buy	0	-	0	-	0	-	0	-	0	-
	sell	100	409	0	-	0	-	0	-	0	-
North America	buy	0	-	0	-	0	-	0	-	380	636
	sell	0	-	200	784	0	-	556	777	0	-
Asia	buy	0	-	0	-	0	-	0	-	610	633
	sell	0	-	100	695	140	773	344	782	0	-
Europe	buy	0	-	0	-	0	-	0	-	100	630
	sell	0	-	900	676	0	-	0	-	0	-
Middle East	buy	0	-	0	-	0	-	0	-	0	-
	sell	0	-	0	-	200	764	0	-	0	-

overseas SM. The regions comprise refineries and terminals, whereas only two regions contains oil extraction platforms. Inside the regions, entities are connected by pipelines, but regions and SM are connected to each other by vessels. The refineries produce 5 oil derivatives (Liquefied Petroleum Gas (LPG), Gasoline, Diesel, Jet Fuel and Fuel Oil) by processing three types of crude oil (Table 1). The refineries can operate according to three production plans, which are specific to each refinery: plan A, plan B, and plan C. Although it is not a real situation, it is representative and fits for a proof-of-concept, which is accomplished by comparing the HCOP model with a conventional approach to manage the oil supply chain.

The derivatives sale prices of EE are shown in the Table 3.

The logistic network includes both oil pipeline and vessel modals, which are used to transport oil and its derivatives. The logistic submodel considers transport arcs between each pair of related holons. The planning horizon Γ is 2 months, since the model doesn´t include the strategic level. Thus, the periods P_η are:

- P_1 = one bimester (high tactical level)
- P_2 = one month (low tactical level)
- P_3 = one week (operational level)

It was considered contracts with each of the five continents involved, where products can be bought or sold (see Tables 4 and 5, which show associated quantities and prices). If it is necessary to buy or sell products in SM, there are bounds regarding both operations in accordance with Tables 6 and 7, which also show the corresponding prices for derivatives and oils.

Figure 8 illustrates the scope and holonic organization of the case study. Due to the difficulty of visualization the fundamental echelon is omitted in the figure.

6.2.2 Control Scenario

A controlled experiment is a scientific test done under controlled conditions, where just one (or a few) factors or variables are changed at a time, while all others are kept

Table 5 Oil Contracts Data

Continent		light		medium		heavy	
M m3 - $		Qty	Price	Qty	Price	Qt	Price
Africa	buy	0	-	0	-	0	-
	sell	1300	680	0	-	0	-
North America	buy	0	-	1000	680	1638	640
	sell	0	-	0	-	0	-
Asia	buy	0	-	1000	660	2273	642
	sell	0	-	0	-	0	-
Europe	buy	0	-	0	-	453	640
	sell	0	-	0	-	0	-
Middle East	buy	0	-	0	-	0	-
	sell	1000	667	0	-	0	-

Table 6 SpotMarket Sale/ Purchase Bounds and Prices for Derivatives

	lpg		gasoline		jet fuel		diesel		fuel oil	
M m3 - $	Qty	Price	Qty	Price	Qty	Price	Qty	Price	Qty	Price
buy	1000	328	1000	669	1000	845	1000	663	1000	524
sell	1000	348	1000	711	1000	897	1000	703	1000	556

constant. It is split into two groups : the *experimental group* and the *control group* (sometimes called *comparison group*). The experimental group is given the experimental treatment, while the control group is given either a standard treatment or nothing. The conditions must be exactly the same for all parts of the experiment. The only difference between them must be the item which receives focus.

In this work the control group contains the standard conditions which are used by a conventional centralized planner, considering only routine situations without any unforeseen circumstances. This scenario is called *control scenario* hereafter. Its solution is obtained by taking into account submodels with the main features of the conventional centralized planner and its simplifications, such as the use of linear models both in production and logistics, without considering the operational level closely. On the other hand, the experimental group shows that the HCOP model, which, although deterministic, reacts quickly to disturbances and takes into account the operational level with its nonlinear and scheduling submodels.

Due to the features of the proposed model, it can integrate different optimizers associated with respective subproblems at each level. Thus, its solutions in the control scenario coincide with those obtained by the traditional centralized approach, since they own the same limitations, such as the use of linear models. Thus, it is not possible the direct comparison between the new model and the conventional one with respect to the control scenario. Therefore, the comparisons between them can be made only concerning disturbances in the control scenario or submodel improvements, which constitute the control experiments presented in the next subsections.

The control scenario solution is shown next by the associated profit as well as the total export and import quantities of each product regarding contracts and SM.

Control Scenario

Profit = 8.859.130$

	lpg	gasoline	jet fuel	diesel	fuel oil	light	medium	heavy
Import	162	1226	385	900		2391		
Export					1104		2161	4609

6.2.3 Control Scenario Analysis

The control scenario was used for the analysis of quantitative performance and feasibility of the proposed model. It was observed both the behavior of processing time and the evolution of the search for the optimal solution. The latter was divided into the analysis of the objective function in the form of total profit and the solution itself, which is represented by the emergent variable of the global superholon EE, where each vector component is the extra purchase or sale of the respective product from or to SM. This same solution representation is used in the other experiments discussed in this section.

The important parameters for a HCOP instance are the number of holons, the number of local variables, the number of echelons and the domain size (Marcellino 2013). Since the experiments of this work consist of a single instance of the problem, these parameters were unchanged along all experiments, except the size of the domain, which is the only one that does not depend on the topology and network constraints of the problem. Therefore, the results of the control scenario analysis are valid for all experiments. They are shown by the three graphs of Figs. 9 and 10, which depend on the domain size of all emergent variables using the domain size invariance (see Eq. 17). The domain size is a quantity of paramount importance to the constraint problems in general, because, on one hand, it is closely connected with the degree of difficulty in solving the problem, and, on the other side, it interferes with the accuracy and quality level of the problem solution.

Table 7 SpotMarket Sale/ Purchase Bounds and Prices for Oils

	light		medium		heavy	
M m3 - $	Qty	Price	Qty	Price	Qty	Price
buy	-	-	2000	630	1000	600
sell	3000	860	-	-	-	-

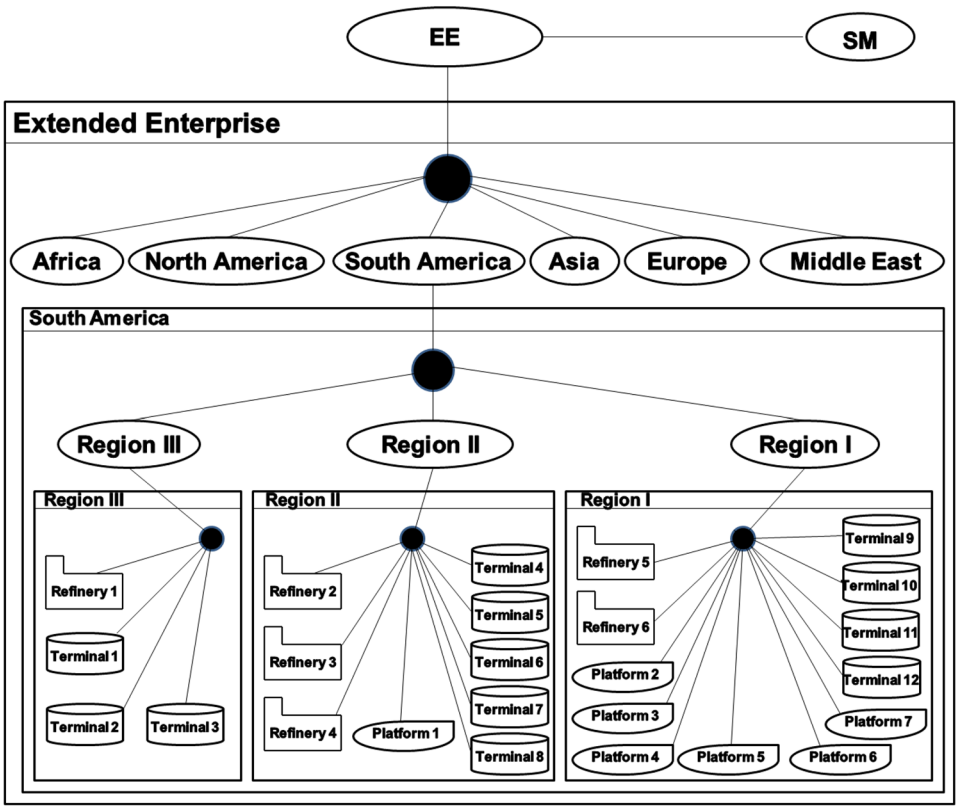


Fig. 8 Overview of the Scope and Holonical Organization of the Case Study

Processing Time The graph of Fig. 9(a) shows the evolution of total processing time. In the time analysis of the model for a basic HCOP (Marcellino 2013), it was observed that it varies linearly with the domain size of holonic variables, while the other parameters are kept. That change is exponential with the number of holons in that work, but it was fixed in all experiments here, since the problem topology is not altered. Thus, it was expected the time behavior to be linear. On the other hand, since the domain size is closely connected with the difficulty in solving the problem, it is expected a growth of processing time with increasing domain size. However, it was observed that the total processing time tends asymptotically to a certain value, which is about 10 minutes. Despite the case study is not completely real, its scope and data are representative and meaningful. Thus, even with few computational resources. the total processing time of each scenario experiment is acceptable. Therefore, the quantitative analysis with respect to processing time is favorable to the proposed model.

Objetive Function : Total Profit The graph of Fig. 9(b) shows the search for the optimal total profit as a function of domain size. In spite of the instabilities that occured for small values of the domain size, it can be seen that the profit function

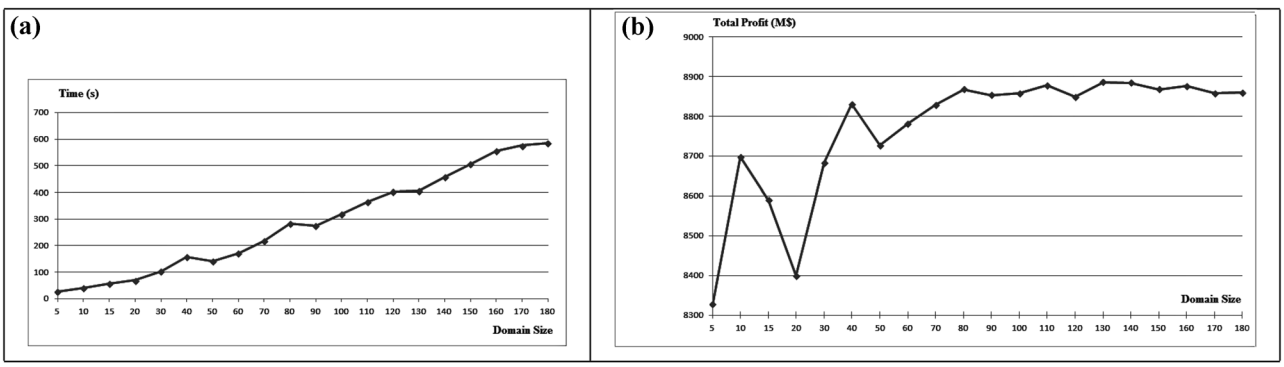


Fig. 9 Scenario Graphs by (**a**) Processing Time (**b**) Total Profit

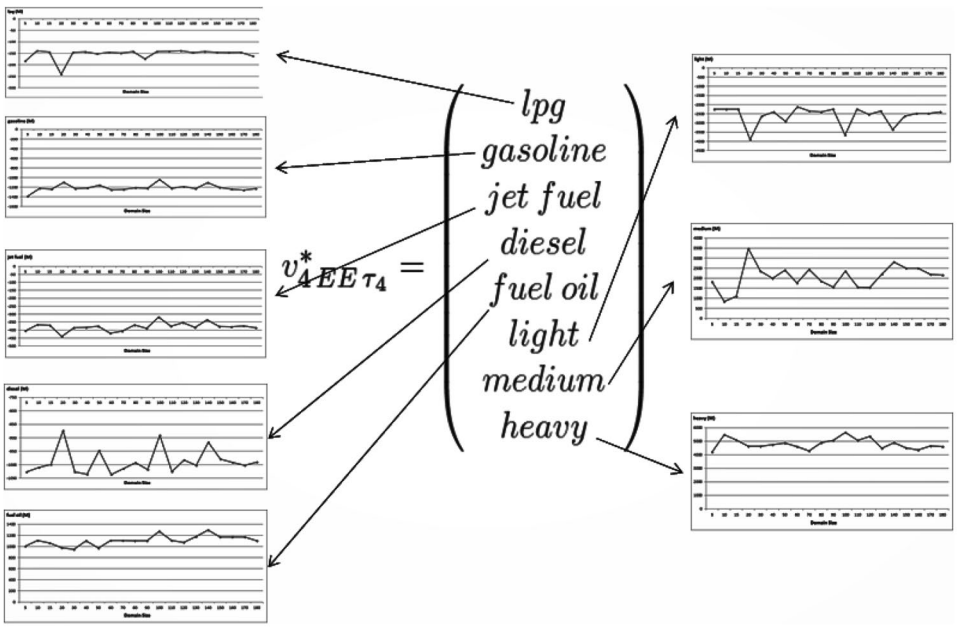


Fig. 10 Scenario Graphs by Product Vector Solution

converges as these values increase, showing the feasibility of the optimization model. This asymptotic value of the objective function represents the optimum according to the proposed model, i.e.:

$$Profit^* = lim_{domainSize \to \infty} Profit(domainSize) \tag{20}$$

Vector Solution This time the graph of Fig. 10 shows the search for the optimal solution in terms of $q^p_{4\,EE\,\tau_4}$, which is the component of the vector emergent variable $v_{4\,EE\,\tau_4}$ associated with the global holon EE as function of the domain size. This variable represents the net quantity of each product p relative to EE, which can be an availability (+) or need (-) in relation to SM. It can also be observed a convergence relative to each vector component corresponding to the respective product, derivative or oil. Thus, the asymptotic value associated with each product p represents the solution sought:

$$v_{4\,EE\,\tau_4}{}^* = lim_{domínioSize \to \infty} v_{4\,EE\,\tau_4}(domínioSize) \tag{21}$$

And the value found is the following:

$$v_{4\,EE\,\tau_4}{}^* = (-62, -26, -45, -90, 14, -91, 161, 245)$$

The existence of convergence, especially in cases of the objective function and the vector solution, was critical to the conclusion that the proposed model is feasible. In a hypothetical situation, considering the functions associated with the domain size for these two cases, if we suppose, by contradiction, that these functions did not converge with the evolution of the domain size, then there would be no acceptable criterion to define a single output value associated with profit and optimal solution vector, respectively. Therefore, the model would not be feasible as a optimization model. On the other hand, the test with the parameter domain size larger than 180 indicated a limit on the precision of solutions achieved by the model for this case study. Probably it is due to the accuracy of the input information and the optimization submodels used, as well as to the decision to consider the same domain size for all the emergent variables of the problem.

6.2.4 Experimental Scenario Analysis

The previous analyses are quantitative, which confirms the feasibility of the optimization model due to the convergence found. In this new phase of experiments, qualitative analyses are performed by comparisons between the control scenario and other experimental scenarios. All experiments use the maximum value of domain size (180), where the solution quality is the best obtained, representing the model optimal value. The experimental scenarios are divided into two cases:

- case I :
  - It consists of unforeseen cases, which affect the problem solution;
  - Except for the previous item and its consequences, it uses the control scenario;
  - It uses the same submodels employed in the control scenario.

- case II :
  - It uses the control scenario;
  - It uses one submodel different from the control scenario.

They are briefly described next.

Case I: Unforeseen Cases Since the proposed model is distributed, it may deal in a more realistic way with situations such as changes in the production capacity of a refinery, the transport capacity of a pipeline or a vessel, the storage capacity of oil and its derivatives at a terminal, the oil extraction rates of a platform, the forecasted demand for a derivative product, etc. In the experiments it was considered the first three cases, whose descriptions and results are presented.

- **Case IA : Refinery Accident**
  It was created a scenario similar to the control one (see Sect. 6.2.2), but with only one change. There was a problem of contamination of the diesel product at the refinery located in Region III. It compromised the entire production of this derivative in this refinery, consequently reducing the associated total production of EE in the first week of the planning period.
  Experimental Scenario : Unforeseen Production Capacity Reduction
  Profit = 8.726.226$

	lpg	gasoline	jet fuel	diesel	fuel oil	light	medium	heavy
Import	165	1237	390	1118		2391		
Export					1104		2399	4371

The proposed model is able to evaluate quickly the unexpected impact on the whole system, estimating the loss entailed by it (132,904 M $) by using the production submodel. The import of derivatives remained practically constant, except for diesel, which was directly affected by a larger amount imported from SM. In addition, it is possible to notice an adaptation of EE to the unforeseen situation, altering the oil export profile, probably due to the effort to increase diesel production.

- **Case I-B : Pipeline Accident**
  It was also created a scenario similar to the control one (Sect. 6.2.2), but this time there was an operational problem with the pipeline between terminal 3 of Region III and refinery 2 of Region II (interregional pipeline). This accident made the pipeline transport capacity to fall by half in the first week of the planning horizon with the resulting decline in refinery performance, which operated at half load during this period.
  Experimental Scenario : Unforeseen Transport Capacity Reduction
  Profit = 8.693.164$

	lpg	gasoline	jet fuel	diesel	fuel oil	light	medium	heavy
Import	154	1257	375	1047		2521		
Export					1076		2161	4741

The model is able to assess the unexpected impact throughout the system quickly, estimating the loss entailed by it (165.966 $) by using the logistic submodel. Coincidentally, the problem affected the diesel again. Unlike the centralized model which would provide its results only later, the proposed model enables the triggering of the necessary measures, such as the preparation for diesel import growth to start earlier, reducing losses.

- **Case I-C : Platform Accident**
  This time there was an operational problem with the platform 1 of Region II, which interrupted completely its oil extraction operation during the first week of the planning horizon.
  Experimental Scenario : Unforeseen Oil Extraction Rate Reduction
  Profit = 8.610.924$

	lpg	gasoline	jet fuel	diesel	fuel oil	light	medium	heavy
Import	143	1213	363	999		2481		
Export					1174		1453	4963

The model is able to assess the impact of the unexpected throughout the system, estimating the loss entailed by it (248.206 $) by using the oil extraction submodel. This time the problem affected the export of medium type oil, which is extracted by the impaired platform.

Case II : Submodel Improvement

- **Case II-A : Nonlinear Production Submodel**
  The control scenario considers for each refinery a conventional and simple production submodel. However, in reality, the refinery operation needs to be modeled taking into account its nonlinear behavior. This experiment uses a production submodel a little closer to reality, dealing with a part of the nonlinearities of the integration between the internal units of a refinery. Figure 11 illustrates a general refinery internal configuration. This production submodel is still simplified, but the nolinearities considered, which are dependent on the API of the oil used, are sufficient to yield differences with relation to the linear submodel. This latter is employed by the conventional centralized planner.

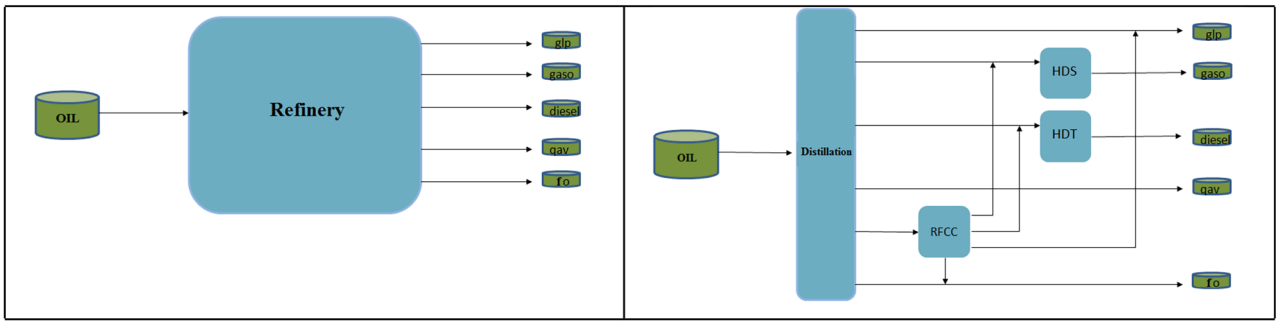


Fig. 11 Nonlinearities from the Integration between the Refinery Internal Units

Experimental Scenario : Nonlinear Production Submodel

Profit = 8.036.426$

	lpg	gasoline	jet fuel	diesel	fuel oil	light	medium	heavy
Import	141	1208	364	1194		3925		
Export					985		2757	5135

In this experiment the model estimates a difference of 822,704 M $ in the expected profit due to an unrealistic assessment made using the linear submodel. However, since the nonlinear submodel leads to more realistic results because it provides a more accurate description of the processes at the operational level, this forecast could be used in the oil contract profile to narrow the gap or even increase profit. The results show a larger amount of light oil import, which is offset by the export of available heavy oil.

- **Case II-B : Transport Scheduling Submodel**

  In general, the conventional planning models are centralized (see Sect. 2.3) and don´t include scheduling submodels in both production and transport at the operational level. The motivation to use the proposed distributed model is to explore the ability to deal with local submodels, which can prevent an optimistic and unreal forecast, making it possible to foresee infeasibilities in the planning solution at the higher levels. Thus, the conventional prediction, which was used in the control scenario, didn´t consider anything related to the scheduling activities. In this experiment it is treated the same scenario, but taking into account a transport scheduling problem in supplying oil to the refinery 3 and refinery 4 in the Region II, where there is a bottleneck due to a shared pipeline between the marine terminal and each of these refineries, as it can be seen in Fig. 12. Information associated with these specific scheduling activities were obtained by the actual offline scheduling application, whose results were embedded into the main model using a offline surrogate model (see Sect. 2.2).

  Experimental Scenario : Scheduling at the Operational Level

  Profit = 8.508.064$

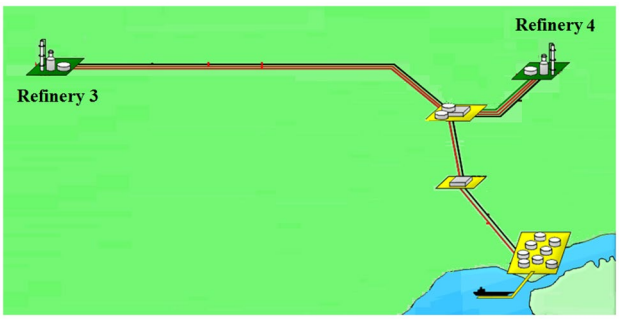


Fig. 12 Bottleneck in the Oil Distribution between Refinery 3 and Refinery 4

	lpg	gasoline	jet fuel	diesel	fuel oil	light	medium	heavy
Import	142	1203	387	1061		2481		
Export					931		1442	5113

The proposed model is able to assess the impact of the scheduling problem delay. This model estimated a difference of $ 351,066 in expected profit due to an unrealistic assessment made with the conventional model by using a more realist scheduling submodel.

7 Conclusions and Future Work

The modeling of the integrated oil industry supply chain planning presents great challenges in terms of scalability, since the chain entities are geographically distributed and with different time scales, algorithm availability, handling nonlinearities, consideration of uncertainties and global optimization. As yet there is no available solution to face that problem. The major contribution of this work is to propose a

model based on holonic agents and constraint optimization programming, which is called Holonic Constraint Optimization Problem (HCOP) in order to get closer to such goals. It also provides a distributed architecture with the integration of available local optimizers and a meta-algorithm called HCOMA, where you can choose the preferred DCOP algorithm to solve the problem. Since a previous work on this subject recommended DPOP as the most suitable DCOP algorithm to solve HCOP, it was chosen for HCOMA to solve the case study problem based on historical data of the PETROBRAS oil company, representing a typical oil supply chain. The experiments worked as a proof of concept, confirming the feasibility of the proposed model due to obtaining convergence in the solution process, as well as assessing the benefits for its eventual use as the basis for a future integration tool aiming at prediction and simulation of scenarios. It has proven to perform well in both the solution quality and the time to reach it. Its solution strategy is different from the traditional ones. In general, these depend on large centralized models, which are computationally intractable. On the other hand, the holonic organization considers the optimal choices which are made at the operational level, while searching for the best global solution at higher levels. Besides, HCOP approach has modeling flexibility, since the holonification process naturally takes into account spatial and temporal decompositions and makes it easier to employ the available local optimization algorithms, which are associated with the respective subproblems resulting from the complete problem partition. Typically conventional centralized planners use linear models and a simplified representation of the entire chain, leading to a suboptimal or an unrealistic optimal evaluation. They ignore nonlinear submodels in both the production and transport areas, for instance, which may be incorporated by the proposed model. Thus, this latter may provide more accurate descriptions of the processes at the operational level. Also, it reacts quickly to environmental disturbances due to being distributed.

This work selected an uniform domain size for all variables as a first approach. In future work, it is worth investigating the variation of the domain size selectively concerning different echelons. Such research may deepen our understanding of the HCOP model solution process and then lead to the evolution of the solution method. In this way, other experiments can be performed to take advantage of part of a previous solution before the occurrence of disturbances in the form of unforeseen events. Thus, it can be taken as an initial partial solution, reducing the search effort and the time to obtain the new solution. With the same objective of increasing search performance, it is important to focus on considering the problem features before starting an exhaustive search. In this sense, the new process can make the direction intercalation relative to the search path between consecutive echelons. Such a solution strategy may allow tackling problem configurations more complex than the case study, getting closer to the real situation in the future.

Although the DPOP algorithm seems more suited to the HCOP solution intuitively, it is worth getting theoretical ground for that fact through future work, rather than accepting just empirical evidence. Similarly, despite the choice of ignoring the strategic level in this work, the proposed model has features that make it valuable for a future integration attempt between that level and the others considered so far.

Acknowledgements The authors gratefully acknowledge PETROBRAS for authorizing the publication of the information herein. In addition, the opinions and concepts presented are the sole responsibility of the authors.

References

Abdoos M, Esmaeili A, Mozayani N (2012) Holonification of a network of agents based on graph theory. In: Jezic G, Kusek M, Nguyen NT, Howlett RJ, Jain LC (eds) Agent and Multi-Agent Systems Technologies and Applications. Springer, Berlin Heidelberg, pp 379–388

Abdoos M, Mozayani N, Bazzan AL (2013) Holonic multi-agent system for traffic signals control. Eng Appl Artif Intell 26(5):1575–1587. https://doi.org/10.1016/j.engappai.2013.01.007, http://www.sciencedirect.com/science/article/pii/S0952197613000171

Ajili F, Wallace M (2003) Constraint and integer programming: Toward a unified methodology. Hybrid Problem Solving in ECLiPSe

Baptiste P, Laborie P, Pape CL, Nuijten W (2006) Chapter 22 - constraint-based scheduling and planning. In: Rossi F, van Beek P, Walsh T (eds) Handbook of Constraint Programming, Foundations of Artificial Intelligence, vol 2, Elsevier, pp 761–799. https://doi.org/10.1016/S1574-6526(06)80026-X, http://www.sciencedirect.com/science/article/pii/S157465260680026X

Beamon BM (1998) Supply chain design and analysis: Models and methods. Int J Prod Econ 55(3):281–294

Bellman R (1957) Dynamic programming. Princeton University Press

Brandes U, Delling D, Gaertler M, Görke R, Hoefer M, Nikoloski Z, Wagner D (2007) On finding graph clusterings with maximum modularity. In: Brandstädt A, Kratsch D, Müller H (eds) Graph-Theoretic Concepts in Computer Science. Springer, Berlin Heidelberg, pp 121–132

Burckert HJ, Fischer K, Vierke G (1998) Transportation scheduling with holonic mas - the teletruck approach. In: Third International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM'98), pp 577–590

Burke DA (2008) Exploiting problem structure in distributed constraint optimisation with complex local problems. PhD thesis, National University of Ireland, Cork

Chima CM, Hills D (2007) Supply-chain management issues in the oil and gas industry. J Bus 5(6):27–36

Chopra S, Meindl P (2012) Supply Chain Management: Strategy, Planning, and Operation. Prentice Hall

Dechter R, Cohen D et al (2003) Constraint processing. Morgan Kaufmann

Dimitriadis AD, Shah N, Pantelides CC (1997) Rtn-based rolling horizon algorithms for medium term scheduling of multipurpose plants. Comput Chem Eng 21:1061

Eichman DA (2000) Creating a high-performance downstream petroleum supply chain. Achieving Supply Chain Excellence through Technology pp 229–232

Faltings B, Yokoo M (2005) Introduction: special issue on distributed constraint satisfaction. Artif Intell 161(1–2):1–5

Ferber DF (2012) Facadeopl. https://github.com/danielferber/FacadeOPL
Ferber J (1995) Les systèmes multi-agents. Vers une intelligence collective, InterEditions
Fioretto F, Pontelli E, Yeoh W (2018) Distributed constraint optimization problems and applications: A survey. J Artif Intell Res 61:623–698
Forrester JW (1958) Industrial dynamics. a major breakthrough for decision makers. Harv Bus Rev 36(4):37–66
Garcia DJ, You F (2015) Supply chain design and optimization: Challenges and opportunities. Comput Chem Eng 81:153–170. https://doi.org/10.1016/j.compchemeng.2015.03.015, http://www.sciencedirect.com/science/article/pii/S0098135415000861, special Issue: Selected papers from the 8th International Symposium on the Foundations of Computer-Aided Process Design (FOCAPD 2014), July 13-17, 2014, Cle Elum, Washington, USA
Gerber C, Siekmann J, Vierke G (1999) Holonic multi-agent systems. Research Report 99(3)
Giret A, Botti V (2004) Holons and agents. J Intell Manuf 15:645–659
Graves S (1982) Using lagrangean techniques to solve hierarchical production planning problems. Management Sci 28:260
Grossman IE, van den Heever SA, Harjunkoski I (2001) Discrete optimization methods and their role in the integration of planning and scheduling. In: Proceedings of Chemical Process Control Conference 6, Tucson, USA
Grossmann IE (2014) Challenges in the application of mathematical programming in the enterprise-wide optimization of process industries. Theor Found Chem Eng 48(5):555–573. https://doi.org/10.1134/S0040579514050182
Hilaire V, Koukam A, Rodriguez S (2008) An adaptative agent architecture for holonic multi-agent systems. ACM Transactions on Autonomous and Adaptive Systems 3(1)
Hms PR (1994) Hms requirements. http://hms.ifw.uni-hannover.de/:HMS Server
Hobbs JR (1990) Granularity. In: Kleer Jd (ed) Weld DS. Morgan Kaufmann, Readings in Qualitative Reasoning About Physical Systems, pp 542–545
Hooker JN (2007) Integrated methods for optimization, vol 100. Springer Science & Business Media
Hubner JF, Sichman JS, Boissier OA (2002) Model for the structural, functional, and deontic specification of organizations in multiagent systems. In: Simposio Brasileiro de Inteligência Artificial (SBIA), The AAAI Press/MIT Press, pp 118–128
IBM (2018) Ibm ilog cplex optimization studio. http://ibm.com/products/ilog-cplex-optimization-studio
Jennings N (2000) On agent-based software engineering. Artif Intell 117(2):277–296
Koestler A (1967) The Ghost in the Machine, 1st edn. Hutchinson & Co, London
Labarthe O, Espinasse B, Ferrarini A, Montreuil B (2007) Toward a methodological framework for agent-based modelling and simulation of supply chains in a mass customization context. Simul Model Pract Theory
Lasschuit W, Thijssen N (2004) Supporting supply chain planning and scheduling decisions in the oil and chemical industry. Comput Chem Eng 28(6–7):863–870
Leaute T, Ottens B, Szymanek R (2009) FRODO 2.0: An open-source framework for distributed constraint optimization. In: Proceedings of the IJCAI'09 Distributed Constraint Reasoning Workshop (DCR'09), Pasadena, California, USA, pp 160–164
Lima C, Relvas S, Barbosa-Póvoa APF (2016) Downstream oil supply chain management: A critical review and future directions. Comput Chem Eng 92:78–92
Lynch NA (1996) Distributed Algorithms. Morgan Kaufmann
Magalhaes MVO, Moro LFL, Smania P, Hassimotto MK, Pinto JM, Abadia GJ (1998) Sipp. a solution for refinery scheduling. In: 1998 NPRA Computer Conference
Maravelias C, Grossmann IE (2004) A hybrid milp/cp decomposition approach for the continuous time scheduling of multipurpose batch plants. Comp and Chem Eng 28:1921
Maravelias C, Sung C (2009) Integration of production planning and scheduling: Overview, challenges and opportunities. Comput Chem Eng 33:1919–1930
Marcellino FJM (2013) Planejamento integrado da cadeia de suprimentos da industria do petroleo baseado em agentes holonicos. PhD thesis, Escola Politecnica, Universidade de Sao Paulo, Sao Paulo(Brazil)
Marcellino FJM, Sichman JS (2010a) A holonic multi-agent model for oil industry supply chain management. In: Ibero-American Conference on Artificial Intelligence, Springer, pp 244–253
Marcellino FJM, Sichman JS (2010b) Oil industry supply chain management as a holonic agent based distributed constraint optimization problem. In: Workshop on Artificial Intelligence and Logistics 2010 in 19th European Conference on Artificial Intelligence, Lisbon, Portugal
Marcellino FJM, Sichman JS (2011) Hcop: Modeling distributed constraint optimization problems with holonic agents. In: Workshop on Artificial Intelligence and Logistics in IJCAI 2011, Barcelona, Spain, p 37
Modi PJ, Shen W, Tambe M, Yokoo M (2003) An asynchronous complete method for distributed constraint optimization. In: AAMAS03
Moise G (2008) An agent-holon oriented methodology to build complex software systems. Int J Comput
OMG (2015) Omg unified modeling language (omg uml) version 2.5. www.omg.org/spec/UML/2.5/PDF
Papadimitriou CH, Steiglitz K (1982) Combinatorial optimization, vol 24. Prentice Hall Englewood Cliffs
Perea E, Grossmann IE, Ydstie E, Tahmassebi T (2001) Dynamic modeling and decentralized control of supply chains. IEC Res 40:3369
Petcu A (2009) A class of algorithms for distributed constraint optimization, vol 194. Ios Press
Petcu A, Faltings B (2004) A distributed, complete method for multi-agent constraint optimization. In: Proceedings of the Fifth International Workshop on Distributed Constraint Reasoning (DCR2004) in CP 2004
Petcu A, Faltings B (2005) A scalable method for multiagent constraint optimization. In: Proceedings of the International Joint Conference on Artificial Intelligence, pp 266–271
Rodriguez S, Hilaire V, Koukam A (2005) Formal specification of holonic multi-agent systems framework. In: Sunderam VS, van Albada GD, Sloot PMA, Dongarra J (eds) Computational Science - ICCS 2005. Springer, Berlin Heidelberg, pp 719–726
Sahinidis NV (2004) Optimization under uncertainty: state-of-the-art and opportunities. Comput Chem Eng 28(6):971–983. https://doi.org/10.1016/j.compchemeng.2003.09.017, URL http://www.sciencedirect.com/science/article/pii/S0098135403002369, fOCAPO 2003 Special issue
Schiex T, Fargier H, Verfaillie G et al (1995) Valued constraint satisfaction problems: Hard and easy problems. IJCAI 1(95):631–639
Shapiro JF (2006) Modeling the Supply Chain. Duxbury Press, Pacific Grove CA
Suda H (1989) Future factory system in japan. Journal of Advanced Automation Technology 1
Tsang EPK (1993) Foundations of Constraint Satisfaction. Computation in cognitive science, Academic Press
Ulieru M, Geras A (2002) Emergent holarchies for e-health applications: a case in glaucoma diagnosis. In: IEEE 2002 28th Annual Conference of the Industrial Electronics Society IECON 02, IEEE, vol 4, pp 2957–2961
Van Den Heever SA, Grossmann IE (1999) Disjunctive multiperiod optimization methods for design and planning of chemical process systems. Comput Chem Eng 23(8):1075–1095
Vecchietti A, Grossmann IE (2000) Modeling issues and implementation of language for disjunctive programming. Comput Chem Eng 24:2143–2155

Versteegh F, Salido MA, Giret A (2010) A holonic architecture for the global road transportation system. J Intell Manuf 21(1):133–144
Yokoo M, Hirayama K (1998) Distributed constraint satisfaction algorithm for complex local problems. In: Proceedings International Conference on Multi Agent Systems (Cat. No. 98EX160), IEEE, pp 372–379
Yokoo MC, Durfee EH, Ishida T, Kuwabara K (1992) Distributed constraint satisfaction for formalizing distributed problem solving. In: International Conference on Distributed Computing Systems, pp 614–621
Yuan Y, Liang TP, Zhang JJ (2003) Using agent technology to support supply chain management: potentials and challenges. In: Supply Chain Transformation in the eBusiness Environment: Issues, Solutions and the Future, the First Annual Symposium on Supply Chain Management
Zelinkovsky R (1999) Transport system. US Patent 5,928,294