D3.4 – First report on methods for distributed optimisation of plants with shared resources

Lukas Samuel MAXEINER\textsuperscript{a} (TUDO)
Simon WENZEL\textsuperscript{a} (TUDO)
José Luis PITARCH\textsuperscript{b} (UVA)
María Paloma MARCOS\textsuperscript{c} (UVA)
\textsuperscript{a}Research Assistants - Germany
\textsuperscript{b}Research Associate - Spain
\textsuperscript{c}Research Assistant - Spain

April 2018

www.copro-project.eu
The goal of CoPro is to develop and to demonstrate methods and tools for process monitoring and optimal dynamic planning, scheduling and control of plants, industrial sites and clusters under dynamic market conditions. CoPro pays special attention to the role of operators and managers in plant-wide control solutions and to the deployment of advanced solutions in industrial sites with a heterogeneous IT environment. As the effort required for the development and maintenance of accurate plant models is the bottleneck for the development and long-term operation of advanced control and scheduling solutions, CoPro will develop methods for efficient modelling and for model quality monitoring and model adaption.
Abstract
This document is the first report on methods for distributed optimisation of plants with shared resources. It covers the fundamentals and the state of the art of distributed optimisation for the optimal allocation of shared resources in the process industry. Starting from the two main concepts of Lagrangian relaxation and the Alternating direction method of multipliers (ADMM), the challenges for an application of such methodologies to industrial use cases are highlighted. Challenges are the incorporation of discrete decisions, uncertainty, and the privacy among constituent systems, which are laid out as future research directions that are going to be treated within the scope of the project.

Revision History
The following table describes the main changes done in the document since it was created.

<table>
<thead>
<tr>
<th>Revision</th>
<th>Date</th>
<th>Description</th>
<th>Author (Organisation)</th>
</tr>
</thead>
<tbody>
<tr>
<td>v0.1</td>
<td>Feb. 19, 2018</td>
<td>Creation of the document.</td>
<td>Simon WENZEL (TUDO)</td>
</tr>
<tr>
<td>v0.2</td>
<td>Mar. 12, 2018</td>
<td>Added chapters 1,2,3,4.</td>
<td>S. WENZEL &amp; L.S. MAXEINER (TUDO)</td>
</tr>
<tr>
<td>v0.3</td>
<td>Mar. 21, 2018</td>
<td>Lenzing case study added.</td>
<td>M.P. MARCOS &amp; J.L. PITARCH (UVA)</td>
</tr>
<tr>
<td>v0.8</td>
<td>Mar. 28, 2018</td>
<td>Added conclusion.</td>
<td>Lukas S. MAXEINER (TUDO)</td>
</tr>
<tr>
<td>v0.9</td>
<td>Apr. 5, 2018</td>
<td>Incorporation of the feedback from CERTH.</td>
<td>Lukas S. MAXEINER (TUDO)</td>
</tr>
<tr>
<td>v1.0</td>
<td>Apr. 6, 2018</td>
<td>Internal review</td>
<td>S. WENZEL &amp; L.S. MAXEINER &amp; S. ENGELL (TUDO)</td>
</tr>
<tr>
<td>v1.1</td>
<td>Apr. 18, 2018</td>
<td>Changes from consortium</td>
<td>Christine Maul (COV)</td>
</tr>
<tr>
<td>v1.2</td>
<td>Apr. 30, 2018</td>
<td>Final Revision</td>
<td>S. WENZEL &amp; L.S. MAXEINER &amp; S. ENGELL (TUDO)</td>
</tr>
</tbody>
</table>

Disclaimer
THIS DOCUMENT IS PROVIDED “AS IS” WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NONINFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, OR ANY WARRANTY OTHERWISE ARISING OUT OF ANY PROPOSAL, SPECIFICATION OR SAMPLE. Any liability, including liability for infringement of any proprietary rights, relating to use of information in this document is disclaimed. No license, express or implied, by estoppels or otherwise, to any intellectual property rights are granted herein. The members of the project CoPro do not accept any liability for actions or omissions of CoPro members or third parties and disclaims any obligation to enforce the use of this document. This document is subject to change without notice.
# Table of Contents

1 Introduction
   1.1 Distributed production systems in the process industries ........................................... 1
   1.2 Potential of better coordinated production in the process industries ............................. 1
   1.3 Motivation for the use of distributed coordination methods ......................................... 2
   1.4 Structure of the report .................................................................................................... 2

2 Problem formulation .............................................................................................................. 3

3 Decomposition methods
   3.1 Algorithms for distributed optimisation of plants with shared resources ........................ 5
   3.2 Market-based Coordination for Continuous Production Units ........................................ 7
      3.2.1 Problem Description and Coordination Principle .................................................... 7
   3.3 Sharing of resources using the Alternating Direction Method of Multipliers .................... 9

4 Extension of the existing methods .......................................................................................... 11
   4.1 Extensions for discrete decisions .................................................................................... 11
      4.1.1 Coordination algorithm .......................................................................................... 11
      4.1.2 Illustrative example 1: Load balancing among coupled production units ................ 12
      4.1.3 Illustrative example 2: Coordination of semi-batch reactors .................................. 13
      4.1.4 Conclusions and future work ................................................................................. 14
   4.2 Extension to the connection of systems to external markets ........................................... 15
      4.2.1 Augmented sub-gradient update ............................................................................. 15
      4.2.2 Procurement by the coordinator ............................................................................. 20
      4.2.3 Conclusion and outlook ........................................................................................ 24

5 CoPro applications ................................................................................................................. 25
   5.1 Distributed optimisation and optimal procurement for the Covestro gas networks .......... 25
   5.2 Distributed optimisation in the Lenzing evaporation park ................................................. 26
      5.2.1 Cooling-water optimisation .................................................................................... 27
      5.2.2 Load-allocation optimisation ................................................................................. 28
      5.2.3 Proposal for a joint distributed optimisation ......................................................... 29

6 Summary and future work ...................................................................................................... 30

References ............................................................................................................................. 31
List of Figures

1. Illustration of the topology of distributed production plants. ........................................ 1
2. Illustration of a joint system. .............................................................................................. 3
3. A typical management structure of a large industrial complex with distributed optimisers. .. 8
4. Operating ranges of the plants. ............................................................................................ 12
5. Simulation results illustrative example 1 ............................................................................ 13
6. Trajectories indicating the usage of the constrained feed. ................................................. 14
7. Evolution of final times and prices. ..................................................................................... 14
8. Site-wide optimisation problem. ........................................................................................ 15
9. Optimal response of the systems to the transfer prices in the one dimension. .................... 16
10. Resulting trajectories of characteristic quantities. ............................................................. 18
11. Portfolio tariff. ................................................................................................................... 21
12. Transfer prices against iterations. ...................................................................................... 23
13. Resource balances against iterations. ................................................................................ 23
14. Spinbath evaporation plant with surface condenser ........................................................... 26
1 Introduction

In the process industries, different plants on a site are often coupled by tight interconnections. These interconnections can be, e.g., flows of material or energy, and they physically link the individual production steps. In addition, production plants, production complexes, or whole production sites are linked to external markets on which quantities of energy and materials are exchanged with suppliers or customers. Such production systems can be considered as systems-of-systems (SoS) according to Maier [1] and can be found in many fields, not restricted to the process industries. Key characteristics of such systems are some form of coupling, operational and managerial independence as well as limited geographic distribution. Coordinating these distributed production systems in the process industries offers a great potential with respect to energy and resource efficiency [2]. Within the CoPro project, decomposition methods for the coordination of interconnected systems will be extended to suit the needs of the process industries, which are characterised by the interplay of continuous streams of energy and material and discrete decisions.

1.1 Distributed production systems in the process industries

Typically, the production in the process industries is done in medium to large sized production sites. These production sites often operate their own facilities for providing steam and electricity, where the energy is then shared between the different companies. The necessary networks for the transportation of energy and materials are often provided and maintained by site-operators as a service. Due to an optimised flow of the energy and material between the different processing steps, e.g., individual units are heat integrated or the off-gas from some processes is burned in the on-site power plant, there exist tight physical interactions between the different unit operations in a process (see Fig. 1). These unit operations can be linked on different levels, i.e., within a plant, within a complex, or across the plants. The extreme, yet not uncommon, situation is when processes are even coupled across the borders of production sites. Then, the material (or energy) is transferred via pipelines or by means of ships, trains, or trucks.

![Figure 1: Illustration of the topology of distributed production plants that are coupled by a common distribution grid [3].](image)

1.2 Potential of better coordinated production in the process industries

With respect to the energy and resource efficiency of an overall production site that incorporates all units, plants and complexes (subsystems), an integrated optimisation of the production offers a great potential. The objective of such an optimisation can be the reduction of the total cost of operation, the reduction of the carbon footprint of the site, or combinations of these and other performance measures. The constraints of such optimisation problems include on the top level the balancing of the linking nodes between the individual
systems. The streams of energy and material that link the various processes will in the following be called shared resources.
An even greater potential for an increase in efficiency is to couple the optimisation of the whole production system to external influences such as varying energy prices, weather conditions, or the current market situation. An overall optimised operation can be to optimally shift time-varying operating loads, jobs, or target amounts to favourable points in time. For instance, an energy consuming cooling or compressing unit operation can be shifted to times when the weather forecast is predicting a windy and sunny day on which the electricity price will most likely drop.
Taking all these influences into account, a well coordinated overall system will make the most out of the existing infrastructure and of the available sources of power and material by using the existing resources in the most favourable manner and contribute significantly to achieving global resource efficiency goals.

1.3 Motivation for the use of distributed coordination methods

An integrated optimisation of the overall system subject to a balanced shared resource network however is often not applicable mainly due three reasons:

- the overall problem becomes too large to be solved efficiently,
- centralised solutions are vulnerable to missing or unreliable information,
- the plants may belong to different business units or even to different companies, which do not want to share their models and economic performance indicators.

Yet, finding an overall optimal operating point is still needed to increase the efficiency of existing production systems. In this context, distributed coordination methods can be used to recover this optimal operating point. There are different coordination schemes that aim at recovering the centralised solution of the overall optimisation problem in a distributed manner. The methods which will be discussed in Section 3 mainly differ in the quality and quantity of the exchanged information between the subsystems of the overall system. Which method is best suited depends on the specific case. If the main challenge is to efficiently solve a large optimisation problem, where in principle all pieces of information are available on the upper layer, then one would opt for a method that shares qualitatively rich information between the systems while reducing the frequency of the exchange. If however, the aim is to keep a certain level of confidentiality among the subsystems, e. g., because of managerial reasons, then one would choose a scheme, where only little information is exchanged and the systems remain semi-autonomous with managerial independence.

1.4 Structure of the report

In this report, at first the general formulation of the optimisation problem will be described. Afterwards, suitable decomposition methods will be reviewed and extended according to the needs of the CoPro project. Then application examples will be provided that illustrate the functionality of the coordination schemes. Finally, the directions for future work will be outlined.
2 Problem formulation

There are various criteria that can be used to measure the quality of operation for individual processing units or plants. Not only is every plant subject to economical performance indicators, but also maximising production quantity, resource efficiency, reliability, reduction of environmental impact such as the carbon footprint or CO₂ emissions are targeted. Based on these considerations, for every unit or plant, quantitative objectives can be formulated and combined using suitable weights.

From a mathematical point of view, one can implement the trade-off between different criteria for the operation of a single unit or plant \( i \) into an objective function \( f_i \) in order to quantify the success in the direction of the predefined goals. The value of this objective depends on the different decisions \( x_i \) that are taken.

\[
\max_{x_i} f_i(x_i). \tag{2.1}
\]

In general, not all decisions can be taken freely, since there are several non-negotiable conditions that have to be met. These can be technical limitations of the equipment or thermodynamic laws, requirements given by a regulator, or by contracts with suppliers or customers. All these constraints can be aggregated as a set of constraints \( g_i(x_i) \leq 0 \).

The solution of Eq. (2.1) while satisfying the set of constraints are optimal decisions \( x_i^* \) as well as a value of the achieved objective \( f^* \). Considering a purely economic objective often results in a linear function, where the goal is to use as few resources as possible to generate as much product as possible and the objective value returns the profit. Other objectives, for instance the tracking of certain reference values, are typically of quadratic form, where certain purities or other constraints must be met. This mathematical structure of the objective is important when considering suitable algorithms for the solution of the optimisation problem at hand.

The different decisions \( x_i \) can either be continuous variables such as temperatures, pressures, or flows of resources, e.g., amounts of product or steam, or discrete decisions as for example which recipe, equipment, contract or tool to select. Depending on the type of the decisions \( x_i \) and their number, the difficulty to find the best solution varies significantly.

For a joint optimisation, the interconnecting streams have to be the same for all systems to fulfil mass and energy balances. See for example the stream \( r \) connecting Plant 1 and Plant 2 in the top of Fig. 2. If those variables become degrees of freedom for the respective plants, then each plant can set the stream locally. Hence, the system is distributed. In order to ensure that the bottom of Fig. 2 corresponds to the top, the independent systems are coupled via global constraints. If \( N \) individual units or plants are connected via networks of shared resources, to which they are either suppliers or consumers, changes in the operating point of one system will have an impact on the others, if the operating point changes the shared resource utilisation. This balance of the shared resource networks in a system can be expressed by the following constraint:

\[
\sum_{i=1}^{N} r_i = r_c. \tag{2.2}
\]
If the networks only contain participating units, then the networks must be balanced, i.e., \( r_e = 0 \). If, however, a certain amount of external resources \( r_e \) can be used, streams can leave the system under consideration, or there are external units that are not part of the optimisation, \( r_e \neq 0 \). The general mathematical form for problems where the objective is the sum of the individual objectives can be expressed as:

\[
\begin{align*}
\min_{\mathbf{x}_i \forall i} & \quad \sum_{i=1}^{N} f_i(\mathbf{x}_i), \\
\text{s.t.:} & \quad g_i(\mathbf{x}_i) \leq 0 \quad \forall i, \\
& \quad r_i = h_i(\mathbf{x}_i) \quad \forall i. \\
& \quad \sum_{i=1}^{N} r_i = r_e.
\end{align*}
\]

Here, the mapping of the decision variables \( \mathbf{x}_i \) to the contribution \( r_i \) to the networks is given by \( h_i \). For simplicity, in the remainder of this deliverable, a linear mapping for \( h_i \) is used as notation, i.e., \( r_i = A_i \mathbf{x}_i \) with \( A_i \in \mathbb{R}^{m \times n_{x_i}} \) with \( m \ll n_{x_i} \). Depending on the mathematical structures of the objective function and the constraint functions as well as on the domain of the decision variables, the problem can belong to different classes of optimisation problems, for which the effort to find solutions differs significantly. For continuous decision variables \( \mathbf{x}_i \), it can be a linear program (LP: \( f_i(\mathbf{x}_i), g_i(\mathbf{x}_i) \) linear), quadratic program (QP: \( f_i(\mathbf{x}_i) \) quadratic, \( g_i(\mathbf{x}_i) \) linear), or non-linear program (NLP: \( f_i(\mathbf{x}_i) \) or \( g_i(\mathbf{x}_i) \) non-linear). If some decisions are discrete, then the class changes to mixed integer and continuous, where again the same differentiations can be made with respect to the objective and the constraints. In the following chapter, different decomposition methods for solving (2.3) are explained and it is indicated for which problem class they can be used.
3 Decomposition methods

Decomposition methods are applied to divide the original problem into sub-problems, the solutions of which depend on coordination variables. Most methods use a two layer hierarchical approach, where communication between the subsystems is enabled or managed by an upper layer. The upper layer ensures the satisfaction of the global constraints in order to find an optimal state of the overall system. Therefore, it is also called coordination layer. The following state of the art overview is based on the deliverable 2.3 from the DYMASOS project [4].

3.1 Algorithms for distributed optimisation of plants with shared resources

Different techniques can be found in the literature for distributed and hierarchical optimisation [5] depending on the degree of autonomy of the constituent systems, e.g., whether systems assign some of their degrees of freedom using a local management (optimisation), and based on the overall system architecture, e.g., whether there exists a central entity that can coordinate the local decisions by issuing some global reference signals.

One of the first decomposition strategies developed is the Dantzig-Wolfe (DW) decomposition [5] for linear programs. This method can be classified as a distributed simplex algorithm and can be used for the efficient solution of linear programming problems where the constraints structure is sparse such that there exist subsets of constraints of joint (where almost every variable occurs) and local (where only few variables occur) character. This technique is appealing for the practical implementation as it was shown to be very efficient [6]. The downside of the DW algorithm is that the coordinator requires information about the local constraints of the subsystems so that privacy restrictions cannot be respected if this method is in use.

Distributed optimisation of plants with shared resources can be realised through application of dual decomposition approaches [7]–[10]. These techniques are also suitable for the application in model predictive control (MPC) of interconnected units [11]–[13]. The general procedure can be interpreted by market theory and resembles a Walrasian auction in which an upper level coordinator takes the role of an auctioneer and the local optimisers are the agents that take part in the auction [14]. The auctioneer iteratively sets prices until the demand and the supply of the agents match. To apply this interactive process, the local objectives must depend on the prices of shared resources. The Lagrange multipliers can thus be interpreted as transfer prices that either incur costs for the local plants, when these consume resources, or they create a reward for the production of resources [15].

In price-based distributed optimisation approaches, prices for the resources enter into the local optimisation problems as linear penalty or reward terms. A central coordinator influences the local optimisation by manipulating the price in the same way as in an open market to steer the auction towards the point where the supply meets the demand. These algorithms usually only require information about the local utilisation of the shared resources, which makes them attractive for cases where the different subsystems do not want to make the detailed knowledge of their internal operations public.

The current practice for balancing of production and consumption of shared resources in the process industries is through direct negotiation of medium to long term prices of the shared resources between the management of the different companies, business units, or plants, resulting in an internal micro-market. This points to the potential of such distributed optimisation schemes of this well known method [14].

In [8] and [16], the authors propose a solution based on multi-agent systems for a distributed resource allocation in technical or industrial environments. They refer to a mathematical formulation of the resource allocation problem as an optimisation problem and use this formulation to derive the local behaviour and the interaction behaviour of the agents. The novelty presented here is the combination of market-based mechanisms for resource allocation and process control in continuous production.

In [9], the authors propose a distributed optimisation scheme for MPC applications that can be formulated as a large-scale quadratic program with linking equality constraints. The price adjustment algorithm is based on the Newton’s method and employs sensitivity analysis and active set change identification techniques. The proposed decentralised MPC framework is tested across several case study examples and the results are promising.
Deliverable 3.4
First report on methods for distributed optimisation of plants with shared resources

in providing an acceptable on-line calculation speed for solving industrial plant-wide MPC optimisation problems.

In [17], the authors present a coordinated-distributed optimisation and control scheme for discrete-time linear process systems. This scheme employs local model predictive controllers that can be coordinated to achieve the centralised optimal performance. For coordination of the local controllers, the authors propose price-based mechanisms and they combine price-driven decomposition [18] with Newton’s method along with a sensitivity analysis technique to update the Lagrange multipliers. Through simulations performed on a fluid catalytic cracking benchmark process coupled by inputs and the states, the authors show that the proposed method provides rapid convergence towards the plant-wide optimum.

Similar work has been done in [19] where the authors present successful implementation of the decentralised NMPC coordination architecture on a laboratory scale oxygen distribution network. In [20], they continue the work and compare different tuning techniques for the price coordination, where it is shown that using market-based algorithms is superior compared to the PI algorithm or heuristic algorithms.

In [21], the authors present the mathematical conditions under which a system-level optimisation of supply and demand scheduling can be implemented as a distributed optimisation in which users and suppliers as well as the load serving entities are decision makers with well-defined partial objectives using price-based coordination. The requirements for convergence of the global objectives and the local/individual objectives are defined and novel algorithms for coordinating of the supply-demand problems on different time scales like day ahead horizon and real time adjustment are developed.

Further studies that touch on the topic of distributed optimisation of plants with shared resources grouped by problem type can be found in:

- [22]–[25] for problems involving linear models,
- [3], [26]–[35] for non-linear problems,
- [36], [37] for limited communication bandwidth,
- [38]–[41] for general cooperative problems and consensus problems.

While price-based coordination reduces the size of the problem of site-wide optimisation and ensures a certain level of confidentiality of the information among the constituent plants, one of the main obstacles towards the industrial application of the technique is the guaranteed rate of convergence [20].

In recent years, different approaches for solving plant-wide optimisation problems have been proposed and one of the most promising algorithms is the Alternating Direction Method of Multipliers (ADMM). ADMM is popular for different types of decomposition problems and an extensive summary of the characteristics and the applications is presented in [15]. The main advantage of ADMM lies in the fact that the convergence of the algorithm can be proven under mild assumptions. ADMM does not require strict convexity of the objective functions of the local optimisation problems as opposed to dual decomposition. ADMM is also guaranteed to converge for non-linear convex models [42]. The difference of ADMM w.r.t. dual decomposition is the modification of local optimisation problems where an augmented Lagrangian formulation is employed. Since the method originates from the 1950s and was fully developed in the 1970s, the authors in [43] argue that ADMM is well suited for distributed convex optimisation, and in particular to large-scale problems. The performance of ADMM versus different state-of-the-art decomposable convex quadratic programming methods employing duality has been investigated in [43].

Although ADMM is also used to solve general large-scale linearly constrained optimisation problems, the majority of theoretical results relate to quadratic and convex functions. Recently, some authors analysed the convergence properties of ADMM for solving certain class of non-convex consensus and sharing problems. In [26] the authors show that the classical ADMM converges to the set of stationary solutions, if the penalty parameter in the augmented Lagrangian is chosen to be sufficiently large [44]. Faster convergence than ADMM is promised by the augmented Lagrangian based algorithm for distributed non-convex optimisation (ALADIN), where on the lower level first and second order information is used for the objectives as well as for the constraints. On the coordinator level, the variables are updated by solving a
QP [45]. This algorithm works well as long as all information from the local systems is known to the coordinator. An extension with proof is given in [46].

Sources that use price-based decomposition techniques to specifically address use cases from the process industries can be found in [3], [8], [9], [19], [20], [28]–[35], [47], [48].

3.2 Market-based Coordination for Continuous Production Units

This chapter introduces distributed optimisation based on dual decomposition, i.e., price-based coordination, explains the principle of such mechanisms and illustrates the application and implementation of such schemes from a mathematical perspective.

3.2.1 Problem Description and Coordination Principle

We study the problem of optimal management of a system that consists of \( N \) subsystems that can be expressed in the following form

\[
\begin{align}
\min_{\mathbf{x}} & \quad \sum_{i=1}^{N} f_i(\mathbf{x}_i) \\
\text{s.t.} & \quad g_i(\mathbf{x}_i) \leq 0, \\
& \quad \sum_{i=1}^{N} A_i x_i = r_e, \quad \forall i \in \{1, \ldots, N\}.
\end{align}
\]

The objective of plant \( i \) is denoted by \( f_i : \mathbb{R}^{n_x_i} \to \mathbb{R} \). For linear objectives \( f_i = \mathbf{h}_i^T \mathbf{x}_i \), where \( \mathbf{h}_i \in \mathbb{R}^{n_x_i} \) contains the weights or prices. In the quadratic case, symmetric positive definite matrices \( \mathbf{H}_i \in \mathbb{R}^{n_x_i \times n_x_i} \) are used to penalise deviation from a reference given by \( \mathbf{x}_{i,Ref} \). The objective is then given by \( f_i = (\mathbf{x}_i - \mathbf{x}_{i,Ref})^T \mathbf{H}_i (\mathbf{x}_i - \mathbf{x}_{i,Ref}) \). In the case of linear inequality constraint matrices, i.e. \( g_i(\mathbf{x}_i) = \mathbf{G}_i \mathbf{x}_i \), with \( \mathbf{G}_i \in \mathbb{R}^{n_g \times n_x_i} \), the problems are LPs or QPs for \( \mathbf{x}_i \in \mathbb{R}^{n_x_i} \). All other forms of \( f_i(\mathbf{x}_i) \) or \( g_i(\mathbf{x}_i) \) are called NLPs for \( \mathbf{x}_i \in \mathbb{R}^{n_x_i} \). The decision variables \( \mathbf{x} := (\mathbf{x}_1, \ldots, \mathbf{x}_N) \) of the local problems, i.e., states of the systems, operational decisions or inputs in the presence of local dynamics and constraints. Note the additive nature and thus separable of the objective function (3.1a). This structure implies that the decision of the \( i \)-th agent can be computed independent (e.g., in parallel fashion) of the rest of the agents.

The conditions (3.1b) represent the local constraints of each individual agent (plant). The constraints (3.1c) represent couplings among the subsystems with \( A_i \in \mathbb{R}^{m \times n_x_i} \) with \( m \ll n_x_i \). These constraints pose the obstacle for complete decentralisation or independent solution of the problem. Often, a central site management of an industrial production complex allocates shared resources among different competing agents, which have managerial autonomy in terms of their local operation (see Fig. 3). The centralised optimisation problem is depicted in Fig. 3. A centralised optimisation of the production site thus involves the evaluation of the local economic costs as well as of a global constraint reflecting the network balance.

The shared resource constraints in (3.1c) comprises the sum of the \( N \) shared resource utilisation vectors \( A_i x_i \in \mathbb{R}^{n_{\text{Glob}}} \) and the external flows \( r_e \) that are leaving or entering the balance space. Such external flows can be intermediate product streams that are sold to other parts of the production site, which are not considered in the balance space, as well as imported or exported utilities like electric power from or to the grid. Here, a supply of a plant to the shared resource network is denoted with a negative sign (\( A_i x_i[j] < 0 \)), while a consumption is represented by a positive quantity (\( A_i x_i[j] > 0 \)), with \( j \) as the index of a particular shared resource.
Problem Decomposition and Hierarchical Coordination by Price-based Mechanisms

The optimality conditions of the problem (3.1) can be formulated using the Lagrange function

\[
L(x_1, x_2, \ldots, x_N, \lambda, \nu_1, \nu_2, \ldots, \nu_N) = \sum_{i=1}^{N} (f_i + \lambda^T A_i x_i + \nu_i^T g_i) - \lambda^T r_e,
\]

where \( \nu_i \) and \( \lambda \) represent the vectors of Lagrange multipliers associated with local and global constraints. Using (3.2), the optimality conditions of problem (3.1) result in

\[
0 = \nabla x_i L = \nabla x_i f_i + A_i^T \lambda + \nabla x_i g_i^T \nu_i, \quad (3.3a)
\]

\[
0 = \nabla \lambda L = \sum_{i=1}^{N} A_i x_i - r_e, \quad (3.3b)
\]

\[
0 = \nu_i g_i, \quad g_i \leq 0, \quad \nu_i \geq 0, \quad \forall i \in \{1, \ldots, N\}. \quad (3.3c)
\]

If the constraints (3.3b) are relaxed, the optimisation problem (3.1) can be decomposed into \( N \) local optimisation problems of the form

\[
\min_{x_i} f_i + \lambda^T A_i x_i
\]

s.t. \( g_i \leq 0 \),

\[
\min \mathcal{L}(x, \lambda) = \min_{x} \mathcal{L}(x, \lambda). \quad (3.5)
\]

To find the update of the price vector \( \lambda^{k+1} \) the old price vector is corrected using the gradient of the dual function,

\[
\lambda^{k+1} = \lambda^k + \alpha^k \nabla d(\lambda^k), \quad (3.6)
\]
where $\alpha^k \in \mathbb{R}$ is the step size parameter. The choice of $\alpha^k$ is critical for convergence. For strictly convex problems, convergence can be proven, if the following holds [49]:

$$\lim_{k \to \infty} \alpha^k = 0$$

(3.7)

$$\sum_{k=1}^{\infty} \alpha^k = \infty$$

(3.8)

In practice, the goal is to find a sufficiently small $\alpha^k$ such that the number of required iterations to reach a certain accuracy $\| \sum_{i=1}^{N} A_i x_i - r_e \| < \epsilon$ is small. It is not always straightforward to find this direction $\nabla d(\lambda^k)$ (e.g., when the functions are not differentiable). In this case the update step can be defined using the subgradient $\partial d$.

$$\lambda^{k+1} = \lambda^k + \alpha^k \partial d(\lambda^k).$$

(3.9)

For the problem at hand a subgradient is the residual of the constraint (3.1c), which can be written as

$$\partial d(\lambda^k) = \sum_{i=1}^{N} A_i x^*_i - r_e,$$

(3.10)

where $x^*_i$ denotes the optimal operating point of subsystem $i$ for the price vector $\lambda^k$. From Eq. (3.10) it can be seen that the prices are updated as long as the constraints are not satisfied. As soon as they are fulfilled, the equilibrium price $\lambda^*$ is attained—similar to the market-clearing price in market theory.

In summary, the principle of the price-based coordination is the following: The coordinator transfers the price vector $\lambda$ to the subsystems and collects the responses $A_i x^*_i$, $\forall i \in \{1, \ldots, N\}$. Based on this information it decides about the new update and this process repeats itself until the primal feasibility condition (3.1c) is fulfilled, i.e., the optimality conditions (3.3a)–(3.3c) are satisfied.

### 3.3 Sharing of resources using the Alternating Direction Method of Multipliers

Problem (3.1) can also be solved using the Alternating Direction Method of Multipliers which originally was motivated by the goal to ensure convergence for a larger class of problems [15], [50].

**Alternating Direction Method of Multipliers**

If we form the augmented Lagrangian (we neglect the local constraints $g_i$ in this part for clarity), the satisfaction of the constraints is enforced by additionally using a quadratic penalty, we get the following Lagrangian:

$$\mathcal{L}(x, \lambda) := \sum_{i=1}^{N} f_i(x_i) + \lambda^T \left( \sum_{i=1}^{N} A_i x_i - r_e \right) + \frac{\rho}{2} \left\| \sum_{i=1}^{N} A_i x_i - r_e \right\|_2^2.$$

(3.11)

Since this Lagrangian, is not separable, new variables are introduced and the problem is rewritten as

$$\min_x \sum_{i=1}^{N} f_i(x_i)$$

s.t. $A_i x_i = z_i,$

$$\sum_{i=1}^{N} z_i - r_e = 0,$$

(3.12)

where the resulting augmented Lagrangian,

$$\mathcal{L}(x, \lambda, z) := \sum_{i=1}^{N} \left( f_i(x_i) + \lambda^T (A_i x_i - z_i) + \frac{\rho}{2} \left\| A_i x_i - z_i \right\|_2^2 \right) + I \left( \sum_{i=1}^{N} z_i - r_e \right).$$

(3.13)
is now separable in decision variables $x_i$. The indicator function $I$ returns 0, if the content of the brackets evaluates to 0 and $\infty$ otherwise. The Lagrangian is a function of the local decision variables, the Lagrange multiplier associated with the satisfaction of the global constraints, and the variables $z_i$. The variables $z_i$ can be interpreted as vectors of proposed shared resource utilisation for system $i$, with $\dim(z_i)$ equal to the number of shared resources and $N$ the number of subsystems. The price vector $\lambda$ and the penalty term $\rho \in \mathbb{R}_+$ are the dual variables of the Lagrangian relaxation and the weight of the augmented Lagrangian term in the objective. From the Karush-Kuhn-Tucker conditions we get the iteration scheme with the local minimisation of the extended Lagrangian function

$$x_i^{k+1} = \arg\min_{x_i} f_i(x_i) + \lambda^k T (A_i x_i - z_i^k) + \frac{\rho}{2} \left\| A_i x_i - z_i^k \right\|^2_2,$$

and update rules for $z_i$ and $\lambda$:

$$z_i^{k+1} = A_i x_i^k + \frac{1}{N} \left( \sum_{i=1}^{N} A_i x_i^k - r_e \right),$$

$$\lambda^{k+1} = \lambda^k + \rho \left( \sum_{i=1}^{N} A_i x_i - r_e \right).$$

The update of the $z_i$ variables can be interpreted as a projection step onto the feasible set, while the update of the prices is still a subgradient update as in Section 3.2.1.

It has been shown that the convergence rate of ADMM is $O\left(\frac{1}{k}\right)$ in the worst case [51]. In contrast to dual decomposition, where a careful selection of $\alpha$ was required to ensure convergence, ADMM converges regardless of the choice of $\rho$, only the rate of convergence is influenced as long as the problem is convex. If the local objective functions are strongly convex, [52] states that linear convergence is guaranteed for the algorithm presented in Algorithm 1. For more details about the ADMM approach, the reader is referred to the benchmark

**Algorithm 1** Algorithm for price-based coordination using ADMM.

```plaintext
1: Data: $\rho$, $\lambda^0$, $k_{max}$, $\epsilon$, $N$
2: $z^0 := 0$, $k := 0$
3: Result: $\lambda^*$, $x^*_i$
4: while $k < k_{max}$ do
5:   for $i = 1 : N$ do
6:     $x_i^{k+1} := \arg\min_{x_i} L_i (x_i, \lambda^k, z^k_i)$
7:   end for
8:   if $\left( \left\| \sum_{i=1}^{N} A_i x_i^{k+1} - r_e \right\|_\infty < \epsilon \text{ and } \rho \left\| \sum_{i=1}^{N} A_i x_i^{k+1} - \sum_{i=1}^{N} A_i x_i^k \right\|_\infty < \epsilon \right)$ then
9:     $\lambda^* := \lambda^k$, break
10:    else
11:     $z_i^{k+1} := A_i x_i^k + \frac{1}{N} \left( \sum_{i=1}^{N} A_i x_i^k - r_e \right)$
12:    end if
13:    $\lambda^{k+1} := \lambda^k + \rho \left( \sum_{i=1}^{N} A_i x_i - r_e \right)$
14:    $k = k + 1$
15: end while
```

in [43] and the survey in [15].
4 Extension of the existing methods

In order to tackle the goals of the CoPro project, a first report on the advances made with respect to solving the challenges that were defined at the beginning of the project is given. The previously explained methods are then modified to better address these challenges of discrete variables and two different ways on how external markets can be included in the schemes are introduced.

4.1 Extensions for discrete decisions

One of the most important challenges when dealing with the real world use cases in CoPro is the presence of discrete decisions, where for example equipment can be on or off, a certain number of equipments can be used in parallel, or if the parallel usage of different equipment is mutually exclusive. For the case that only few discrete decisions have to be made, in the following section a coordination algorithm is proposed that finds good feasible solutions for the problem classes of MILPs or MINLPs using ADMM. For these kinds of problems, if ADMM converges, the found solution is at least locally optimal. Convergence itself however cannot be proven and the following results are based on empirical observations.

4.1.1 Coordination algorithm

Using ADMM to find the equilibrium prices by iteratively adjusting the prices for the shared resources, the optimisation problem for each subsystem can be formulated as follows, where for better distinction of the different types of decision variables different symbols $\mathbf{x}$ and $\mathbf{d}$ are used.

$$
\mathbf{x}_i^{(k+1)}, \mathbf{d}_i^{(k+1)} = \arg \min_{\mathbf{x}_i, \mathbf{d}_i} f_i(\mathbf{x}_i, \mathbf{d}_i) + \lambda^{(k)} \mathbf{A}_i \mathbf{x}_i + \frac{1}{2} (\mathbf{A}_i \mathbf{x}_i - \mathbf{z}_i^{(k)})^\top \text{diag}(\rho^{(k)}) (\mathbf{A}_i \mathbf{x}_i - \mathbf{z}_i^{(k)}) , \quad (4.1)
$$

s.t.: $g_i(\mathbf{x}_i, \mathbf{d}_i) \leq 0$.

where $\mathbf{x}_i \in \mathbb{R}^{n_x}$ and $\mathbf{d}_i \in \mathbb{Z}^{n_d}$. The superscript $(k)$ indicates that the values are fixed on the subsystem level during the optimisation in the $(k+1)$-th iteration. If the shared resources are subject to equality constraints, the variables that were fixed in the individual optimisation of the subsystems are updated according to the following scheme [15]:

$$
\lambda^{(k+1)} = \begin{cases} 
\lambda^{(k)} + \xi \rho^{(k)} & \text{if } \sum_{i=1}^N \mathbf{A}_i \mathbf{x}_i^{(k)} - \mathbf{r}_e < 0, \\
\lambda^{(k)} + \rho^{(k)} & \text{else},
\end{cases} \quad (4.2)
$$

$$
\mathbf{z}_i^{(k+1)} = \mathbf{A}_i \mathbf{x}_i^{(k+1)} + \left( \sum_{i=1}^N \mathbf{A}_i \mathbf{x}_i^{(k)} - \mathbf{r}_e \right) N^{-1}. \quad (4.3)
$$

The factor $\xi \in (0, 1]$ is introduced to damp oscillations that can occur for large values for $\rho$. The convergence rate of ADMM is improved using different penalty parameters $\rho[j]$ for the different global constraints and an automatic adjustment of the step size parameter based on primal and dual feasibility, $\Phi_{\text{Primal}}$ and $\Phi_{\text{Dual}}$ [53].

$$
\Phi_{(k+1)}^{\text{Primal}} = \text{abs} \left( \sum_{i=1}^N \mathbf{A}_i \mathbf{x}_i^{(k)} - \mathbf{r}_e \right), \quad (4.5)
$$

$$
\Phi_{(k+1)}^{\text{Dual}} = \text{diag}(\rho^{(k)}) \text{abs} \left( \sum_{i=1}^N \mathbf{A}_i \mathbf{x}_i^{(k)} - \sum_{i=1}^N \mathbf{A}_i \mathbf{x}_i^{(k)} \right), \quad (4.6)
$$

$$
\rho[j]^{(k+1)} = \begin{cases} 
\rho[j]^{(k)} r_{\text{incr}}, & \text{if } \Phi_{\text{Primal}}[j]^{(k+1)} > \beta \Phi_{\text{Dual}}[j]^{(k+1)}, \\
\rho[j]^{(k)} r_{\text{decr}}, & \text{if } \Phi_{\text{Primal}}[j]^{(k+1)} < \beta \Phi_{\text{Dual}}[j]^{(k+1)}, \\
\rho[j]^{(k)}, & \text{else}.
\end{cases} \quad (4.7)
$$
Here, [j] is an index for an element of the vector. If the shared resources are inequality constrained, the prices \( \lambda^{(k+1)} \) and references \( z_i^{(k+1)} \) have to be adjusted as follows:

\[
\lambda_{[j]}^{(k+1)} = \max \left( \lambda_{[j]}^{(k+1)}, 0 \right), \quad (4.2a)
\]

\[
z_{[j]}^{(k+1)} = A_i x_{[j]}^{(k+1)} \quad \text{if} \quad \sum_{i=1}^{N} A_i x_{[j]}^{(k+1)} - r_c [j] < 0. \quad (4.3a)
\]

Additionally, the primal feasibility \( \Phi_{\text{Primal}}^{(k+1)} \) is calculated using Eq. (4.8) instead of Eq. (4.5):

\[
\Phi_{\text{Primal}}^{(k+1)} = \sum_{i=1}^{N} A_i x_i^{(k+1)} - r_c. \quad (4.8)
\]

The convergence criterion is that for consecutive solutions both \( \Phi_{\text{Primal}} \) and \( \Phi_{\text{Dual}} \) are element wise smaller than a predefined tolerance. In the non-convex case there is no proof, however, proper tuning of the parameters \( \xi, \beta, \tau_{\text{incr}}, \tau_{\text{Decr}} \) often leads to convergence.

### 4.1.2 Illustrative example 1: Load balancing among coupled production units

A load balancing problem with three coupled production units is considered. All three plants have different operating ranges as depicted in Fig. 4. Among the plants, the shared resource has to be allocated, where Plant 1 is a producer and the other two are consumers of this shared resource. A relevant case in terms of resource allocation in industrial practice is the existence of discrete decisions on the plant level. This situation occurs for Plant 1, which can choose between two operating ranges.

![Figure 4: Operating ranges of the plants.](image)

(a) Plant 1 has two operating ranges \(-8 \leq A_1 x_1 \leq -1 \lor -15 \leq A_1 x_1 \leq -10 \) and \( A_1 x_{1,\text{set}} = -12 \).

(b) Plant 2 encountered a shift of the upper bound \( 1 \leq A_2 x_2 \leq 3 \leftarrow 10 \), \( A_2 x_{2,\text{set}} = 9 \).

(c) Plant 3 has one operating range \( 1 \leq A_3 x_3 \leq 4 \) with \( A_3 x_{3,\text{set}} = 4 \).

**Mathematical problem description**

The models of the three plants are linear stationary models \( y_i : \mathbb{R} \to \mathbb{R} \) with quadratic cost functions \( f_i \) of the following form:

\[
f_i(x_i) = -y_i(x_i) + \gamma A_i (x_{i,\text{ref}} - x_i)^2, \quad \gamma = 10. \quad (4.9)
\]

The models of the individual plants, denoted by subscript \( i \), are expressed as:

\[
y_1(x_1) = \alpha_{1.0} d_1 + \alpha_{1.0}' (1 - d_1) + (\alpha_{1.1} d_1 + \alpha_{1.1}' (1 - d_1)) x_1, \quad (4.10a)
\]

\[
y_2(x_2) = \alpha_{2.0} + \alpha_{2.1} x_2 \quad (4.10b)
\]

\[
y_3(x_3) = \alpha_{3.0} + \alpha_{3.1} x_3. \quad (4.10c)
\]

The discrete variable \( d_1 \) determines whether line 1 or line 1 and 2 together are used. The reference operating points \( A_i x_{i,\text{ref}} \) and the bounds on \( A_i x_i \) are given in Fig. 4. The overall optimisation problem for the load balancing task can be formulated as in Eq. (2.3) with Eq. (2.3d) being an equality constraint with \( r_c = 0 \).
Simulation results

![Graphs showing simulation results](image)

**Figure 5:** On the left: Resource consumption/production over iterations with the final values indicated by the dotted line. On the right: Shared resource residuals over number of iterations with the dotted line indicating the predefined tolerance.

The simulation results for the load balancing are shown in Fig. 5. On the left, it can be seen that at the start of the coordination the plants operate at their reference operating points or at a constraint. In this situation, the network of shared resources is not balanced as shown on the right. In the course of the coordination the prices $\lambda^{(k)}$ are adjusted such that the plants change their operating point gradually. It can be seen that Plant 1, which starts at $A_1 x_1 \approx -12$, responds with a jump after iteration $k = 10$. This results from the discrete decision to switch off the second production line and thus to change to a different operating range. Finally, the three plants adjust their operating points such that the residual at this point is below $10^{-4}$.

### 4.1.3 Illustrative example 2: Coordination of semi-batch reactors

This illustrative example considers the optimal distribution of resources between semi-batch reactors. In multi-reactor batch-plants scheduling is used to determine the sequence and the duration of the batches. This is usually done using simple models which do not take into account the profiles of consumption of shared resources during the reaction. To overcome this, coordinated model predictive control can be used, where each reactor maximises its throughput by adapting the future trajectories. Ensuring that the available shared resources are not exceeded, the inequality constrained version of the algorithm presented in section 4.1.1 is applied to share resources in each discrete time interval. However, the batch times vary significantly depending on the availability of the shared resources. Thus, the challenge is to find the optimal number of necessary intervals for each reactor.

#### Mathematical problem description

The semi-batch reactor model is taken from [54] and used with the parameters given in [55]. The objective is to maximise the throughput $f_i = n_p / t_F$, i.e., amount of product $n_p$ at time $t_F$. The time $t_F$ is a multiple of the time of a discretised interval, i.e., $t_F = N_{\text{intervals}} \Delta t$ with $\Delta t = 3$ h, and it is adapted between iterations such that upon convergence a desired amount of product $n_p(t_F) \geq n_{p, \text{Des}}$ is reached. In Fig. 6 on the left side, the feed profile $A_i x_i(t)$ for a single reactor is shown, which produces the desired amount of product after 63 h. In this example, three reactors, which are scheduled to start at $t = 0$ h, $t = 3$ h, and $t = 6$ h, share a maximum available amount of feed $(\sum_i A_i x_i)_{\text{max}}$. 
Simulation results

In Fig. 6 on the right, one can see how the profiles are adapted in order to not exceed the maximum available amount of feed flow. In Fig. 7, one can see on the left how the batch times (discrete variables) change over the iterations, and on the right, how the prices for the different intervals change. Changes in the batch times influence the prices. Until the first change in the batch times, there is a continuous evolution of the prices, and the same can be observed after a set of batch times converged after iteration 30. While in this non-convex case no statement about global optimality can be made, this allocation results a feasible solution, which is better than a decentralised one, where the feed $u_{\text{max}}$ is evenly distributed between the different reactors.

4.1.4 Conclusions and future work

Two illustrative examples of different complexity were presented. The first considered one shared resource and one discrete decision with convex objective functions, while the second comprised 15 intervals in which the feed flow has to be shared and several discrete decisions. It was shown that for both examples the algorithm converged and can therefore be used for the shared resource allocation in systems with discrete decisions. Future work will include a more theoretical investigation on convergence as well as the analysis of the influence of more shared resources in the load balancing and the influence of the initial values on the result in the semi-batch reactor example.
4.2 Extension to the connection of systems to external markets

In this section two new algorithms are presented, which address the challenge of combining on-site distributed optimisation with the access to external resources [34], [35]. Considering all the networks of such a production process as closed systems is not realistic, since usually at least some of the networks are connected to external sources or sinks, where, on the spot market or according to contracts, limited amounts can be purchased or sold. The motivation is in particular to investigate and to solve the situation in which the site-internal transfer price for shared resources exceeds the external market price. In this situation, the site management should obviously acquire additional quantities of resources e.g. from a pipeline network. This set-up is illustrated in Fig. 8, where three productions plants are connected via two shared resource networks. For each network there exists a connection to a pipeline.

Two different algorithms addressing this problem will be described in the following sections together with simulations studies based on illustrative examples.

4.2.1 Augmented sub-gradient update

The first method is based on the formulation presented in [35].

Problem formulation

Mathematically, the problem to jointly optimise the objectives of $N$ independent systems that are connected via $n_{Networks}$ networks can be written as Eq. (4.11).

\[
\min_{\mathbf{x}, \forall i} \sum_{i=1}^{N} f_i(\mathbf{x}_i), \quad \text{(4.11a)}
\]

s.t.: \[ g_i(\mathbf{x}_i) = 0, \quad \text{(4.11b)} \]

\[ x_{LB,i} \leq x_i \leq x_{UB,i}, \quad \text{(4.11c)} \]

\[ \sum_{i=1}^{N} A_i x_i = 0. \quad \text{(4.11d)} \]

The objective of each of the $N$ individual sub-problems $i$ is given by $f_i$. The sub-problems are constrained by Eq. (4.11b) and the states $\mathbf{x}_i$ are bounded below by $x_{LB,i}$ and above by $x_{UB,i}$ as indicated in Eq. (4.11c). Additionally, the systems interact with the different networks via their contribution $A_i \mathbf{x}_i$. Overall, the network constraint Eq. (4.11d) has to be satisfied for each network $l$.

In this work, the network constraint is extended in order to include access to $M$ different price levels $j$ for each shared resource, where limited amounts $r_{ij}$ can be bought from external resources or sold to outlets at
the predefined prices $p_j$. These levels can be caused by contracts including price ladders or access to different external markets. The resulting augmented problem is given in Eq. (4.12).

\[
\begin{align*}
\min_{x_i} & \sum_{i=1}^{N} f_i(x_i) + \sum_{j=1}^{M} p_j^T r_j, \\
\text{s.t.:} & \quad g_i(x_i) = 0, \\
& \quad x_{LB,i} \leq x_i \leq x_{UB,i}, \\
& \quad \sum_{i=1}^{N} A_i x_i - \sum_{j=1}^{M} r_j = 0, \\
& \quad r_{lb,j} \leq r_j \leq r_{ub,j}
\end{align*}
\] (4.12a)

\[\text{s.t.:} \quad g_i(x_i) = 0, \quad x_{LB,i} \leq x_i \leq x_{UB,i}, \quad \sum_{i=1}^{N} A_i x_i - \sum_{j=1}^{M} r_j = 0, \quad r_{lb,j} \leq r_j \leq r_{ub,j}\] (4.12b)

### Distributed solution

The problem shown in Eq. (4.11) can be solved using dual decomposition, splitting the problem into $N$ independent sub-problems and an overarching coordination problem. This is achieved by relaxation of the network constraints Eq. (4.11d), i.e., adding the local contribution to the network constraint multiplied by the corresponding dual variable $\lambda$ to the objective function of the individual sub-problems. Thus, the specific problem given in Eq. (4.13) can be solved locally for a given value of $\lambda$:

\[
\begin{align*}
A_i x_i^+ &= A_i \arg \min_{x_i} f_i(x_i) + \lambda^T A_i x_i, \\
\text{s.t.:} & \quad g_i(x_i) = 0, \\
& \quad x_{LB,i} \leq x_i \leq x_{UB,i}
\end{align*}
\] (4.13a)

\[\text{s.t.:} \quad g_i(x_i) = 0, \quad x_{LB,i} \leq x_i \leq x_{UB,i}\] (4.13b)

It is assumed that the subsystems are not sharing their objectives or constraints due to confidentiality reasons and thus each subproblem is solved completely independently from the other problems. Only the responses to transfer prices, i.e., $A_i x_i^+$, are communicated.

In Fig. 9a, a one-dimensional example of aggregated supply, which is defined as the sum of all positive contributions to the network constraint, i.e., $\sum \{A_i x_i^+ | A_i x_i^+ \geq 0\}$, and aggregated demand, defined as $\sum \{A_i x_i^+ | A_i x_i^+ < 0\}$, depending on the value of $\lambda$ are shown. Feasibility, i.e., satisfaction of the network constraint Eq. (4.11d),

**Figure 9:** Optimal response of the systems to the transfer prices in the one dimension.

is achieved by adjusting the dual variables in an iterative procedure. Since only the responses to the transfer...
prices are known on the coordinator level, this cannot be done explicitly. Instead the sub-gradient method is used, where transfer prices are adjusted proportionally to the difference between supply and demand

$$\lambda^+ = \lambda + \alpha \left[ \sum_{i=1}^{N} A_i x_i^+ - \sum_{j=1}^{M} r_j \right]$$

(4.14)

Iterating between local optimisation Eq. (4.13) and the update of the transfer prices is done until a stopping criterion is met, e.g., the maximum residuum $$\| \rho^+ \|_\infty = \| \sum_{i=1}^{N} A_i x_i^+ \|_\infty$$ is less than a predefined tolerance $$\epsilon$$. Economically, this can be interpreted as finding the intersection between supply and demand, cf. Fig. 9a. The update of the transfer prices is done using the step size parameter $$\alpha \in (0, 1]$$. Careful selection of the step size parameter is necessary to ensure convergence, when local constraints $$g_i(x_i)$$ are active. If all objectives and constraint functions are convex, the solution found at convergence of the sub-gradient method is the global optimum of the problem.

Also the augmented problem from Eq. (4.12) can be solved using dual decomposition. While the local optimisation can still be done using Eq. (4.13), on the coordination layer, additionally to finding the optimal transfer prices, the cost for interaction with the external markets, $$\sum_{j=1}^{M} p_j^+ r_j$$, has to be minimised. Finding the optimal values for $$r_j$$ can either be done sequentially or in a combined step. In Eqs. (4.15) and (4.16), the new transfer prices $$\lambda^+$$ and the amounts to be bought or sold $$r_j$$ are calculated sequentially using the sub-gradient method:

$$\lambda^+ = \lambda + \alpha \left[ \sum_{i=1}^{N} A_i x_i^+ - \sum_{j=1}^{M} r_j \right]$$

(4.15)

$$r_j^+ = \min \left( \max \left( r_{LB,j}, r_j + \text{diag}(p_j)^{-1} \text{diag}(p_j - \lambda^+) (r_{UB,j} - r_{LB,j}) \right), r_{UB,j} \right)$$

(4.16)

The other approach is to update both at the same time, where the coordinator update is shown in Algorithm 2. In the example from Fig. 9a, this corresponds to evaluating different realisations for $$\lambda^+$$ using Eq. (4.15), based on minimum and maximum values for $$r$$. If the price range is below the price for the external resource $$p$$, then $$r$$ will be at the lower bound, and the maximum value for $$\lambda^+$$ is used for the next iteration, cf. Case 1 in Algorithm 2. If the price range includes $$p$$, Case 2, the difference between supply and demand can be covered with the external resources and therefore $$\lambda^+ = p$$. If the price range is above the price for the external resources, Case 1 will be used and the external resource $$r$$ is at the upper bound. Case 3 is only required, if the step size $$\alpha$$ is too large. The complete algorithm for balancing multiple networks with several external resources and outlets at different fixed prices is shown in Algorithm 2. The residuum vector $$\rho^+ = \sum_{i=1}^{N} A_i x_i^+ - \sum_{j=1}^{M} r_j^+$$ is again used for the evaluation of the stopping criterion in both approaches.

**Example**

In the following, an example is used to demonstrate the proposed extension of the dual decomposition framework. Quadratic objective functions with positive diagonal scaling matrices and affine constraints of the following form are assumed:

$$f_i(x_i) = (x_i - x_{i,Target})^T B_i (x_i - x_{i,Target}),$$

(4.17a)

$$g_i(x_i) = C_i x_i - d_i,$$

(4.17b)

Five subsystems ($$N = 5$$) are connected via three shared resources ($$n_{Networks} = 3$$). For each resource, there are three different sources from which they can be bought ($$M = 3$$). Each subsystem has four independent variables $$x_i \in \mathbb{R}^4$$ and is subject to two local constraints $$g_i : \mathbb{R}^4 \rightarrow \mathbb{R}^2$$. The transfer prices $$\lambda \in \mathbb{R}^3$$ are initialised with 0, the step size is chosen as $$\alpha = 0.03$$, and the tolerance is set to $$\epsilon = 10^{-6}$$. The matrices and vectors $$x_{i,Target}, B_i, C_i, d_i, A_j, p_j,$$ and $$r_{UB,j}$$ are generated from a random seed and, without loss of generality, the lower bound for flows of external resources is $$r_{LB,j} = 0$$.

$$x_{Target,i} = \begin{bmatrix} -2 & -7 & 4 & 6 \\ -3 & 9 & -10 & 5 \\ -7 & 2 & 8 & -7 \\ -8 & 1 & 4 & -9 \\ 1 & 2 & -7 & -6 \end{bmatrix},$$

$$x_{Target,i} = \begin{bmatrix} -2 & -7 & 4 & 6 \\ -3 & 9 & -10 & 5 \\ -7 & 2 & 8 & -7 \\ -8 & 1 & 4 & -9 \\ 1 & 2 & -7 & -6 \end{bmatrix},$$
Deliverable 3.4
First report on methods for distributed optimisation of plants with shared resources

Figure 10: Resulting trajectories of characteristic quantities for the sequential (seq.) and the combined (comb.) approach with the iterations.
Algorithm 2 Coordinator level update step to determine $\lambda^+$ and $r^+_j$

1: $r^+_j \leftarrow r_{LB,j} \forall j$
2: for $\lambda[k]$ in $\lambda$ do
3: \quad $\Pi \leftarrow [-\infty, p_1[k], p_2[k], \ldots, p_M[k], \infty]^T$
4: \quad $\Delta \leftarrow \lambda[k] + \alpha \left( \sum_{i=1}^N A_i[k]x_i - \sum_{j=1}^M r_{LB,j}[k] \right)$
5: \quad for $l$ in $1 : M+1$ do
6: \quad \quad $\bar{\lambda} \leftarrow \lambda[k] + \alpha \left( \sum_{i=1}^N A_i[k]x_i - \sum_{j=1}^l r_{LB,j}[k] - \sum_{j=l+1}^M r_{LB,j}[k] \right)$
7: \quad \quad if $\pi[l] \leq \bar{\lambda} < \pi[l+1]$ then
8: \quad \quad \quad $\lambda^+[k] \leftarrow \bar{\lambda}$
9: \quad \quad \quad $r^+_j[k] \leftarrow r_{UB,j}^+[k]$ \quad \qquad \qquad $\text{Case 1}$
10: \quad \quad \quad Break
11: \quad \quad else if $\bar{\lambda} \leq \pi[l+1] < \Delta$ then
12: \quad \quad \quad if $\sum_{j=1}^{l-1} r_{j,max}[k] \leq \sum_{i=1}^N A_i[k]x_i < \sum_{j=1}^l r_{j,max}[k]$ then
13: \quad \quad \quad \quad $\lambda^+[k] \leftarrow \pi[l+1]$ \quad \qquad \qquad \qquad $\text{Case 2}$
14: \quad \quad \quad \quad $r^+_j[k] \leftarrow \left( \sum_{i=1}^N A_i[k]x_i - \sum_{j=1,j \neq l}^M r^+_j[k] \right)$
15: \quad \quad \quad else
16: \quad \quad \quad \quad $\lambda^+[k] \leftarrow \bar{\lambda}$
17: \quad \quad \quad \quad $r^+_j[k] \leftarrow r_{UB,j}^+[k]$ \quad \qquad \qquad $\text{Case 3}$
18: \quad \quad \quad Break
19: \quad \quad end if
20: \quad \quad end if
21: \quad $\Delta \leftarrow \bar{\lambda}$
22: \quad end for
23: end for

$$B_i = \text{diag} \left( \begin{bmatrix} 3 & 5 & 7 & 8 \end{bmatrix} \right); \text{diag} \left( \begin{bmatrix} 6 & 4 & 8 & 9 \end{bmatrix} \right); \text{diag} \left( \begin{bmatrix} 1 & 3 & 4 & 5 \end{bmatrix} \right);$$

$$\text{diag} \left( \begin{bmatrix} 4 & 7 & 5 & 9 \end{bmatrix} \right); \text{diag} \left( \begin{bmatrix} 3 & 4 & 7 & 7 \end{bmatrix} \right),$$

$$C_i = \begin{bmatrix} 9 & 7 & 9 & 7 \\ 0 & 5 & 4 & 4 \\ 0 & 2 & 1 & 8 \\ 1 & 6 & 2 & 8 \end{bmatrix}; \begin{bmatrix} 4 & 1 & 8 & 5 \\ 0 & 0 & 5 & 4 \\ 5 & 7 & 6 & 6 \\ 9 & 3 & 9 & 9 \end{bmatrix},$$


$$A_i = \begin{bmatrix} -8 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 6 & 0 \end{bmatrix}; \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & -9 & 0 \end{bmatrix}; \begin{bmatrix} -9 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \end{bmatrix};$$

$$\begin{bmatrix} 9 & 0 & 0 & 0 \\ 0 & -5 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix},$$

$$p_j = [5.89, 2.09, 1.52]^T; [6.28, 3.1, 6.95]^T; [7.54, 3.91, 7.5]^T,$$

$$r_{UB,j} = [3.1, 4.5, 3]^T; [2.2, 2.9, 1.4]^T; [4.8, 1.9, 4]^T, $$

In Fig. 10, the evolution of the residuals $\rho$ and of the transfer prices $\lambda$ over iterations can be seen for both, sequential (filled) and combined (empty), update rules. The distributed solutions converge towards the optimum of the centralised solution, which can be seen from the vanishing residuals and the matching of the
transfer prices at convergence with the centralised computed ones. In general, convergence is assumed, when \( \text{abs}(\rho[i]) \leq \rho_{\text{max}} \forall i \). It can be seen that the transfer price at convergence \( \lambda[1] \) (square) is less than, \( \lambda[2] \) (triangle) is equal to, and \( \lambda[3] \) (circle) is larger than the prices at which external resources can be bought (indicated by the thin lines), i.e., this example covers all three scenarios from Fig. 10. Comparing the two different methods, it becomes evident that the sequential method requires more iterations and that it oscillates towards the optimum.

Discussion

The significantly larger number of iterations in the approach with sequential steps is due to \( \lambda[2] \) being equal to the external price. Since the variables are set consecutively, an overshoot is necessary because otherwise the sub-gradient for the usage of the external resource does not change. For the combined approach, after 90 iterations the network constraint of resource 2 is balanced by the external market and hence the residuum is 0. There are methods to initialise the prices more efficiently, e.g., at one of the prices for external resources.

4.2.2 Procurement by the coordinator

This method follows the nomenclature and problem formulation of [34].

Problem description

The topology of an industrial production site that consists of (partially) autonomous subsystems (plants) that are physically connected by networks for shared resources is depicted in Fig. 8. The internal networks are connected to the pipelines via the site hub. The site-hub manages a surplus of demand or supply of a particular resource by buying or selling resources. This resource allocation in the presence of different price regimes can be formulated as an optimisation problem. It is assumed that the site-hub can acquire limited amounts of shared resources at different rates. A possible tariff structure that reflects the amounts and prices that are available from the pipeline is shown in Fig. 11. The optimisation problem of the site can be formulated as

\[
\begin{align}
\max_{\mathbf{x}_i, \mathbf{y}_i} & \quad \sum_{i=1}^{n} \mathbf{p}_{i, \text{prod}}^\top \mathbf{y}_i - \mathbf{1}^{(1 \times m)} (\mathbf{P} \circ \mathbf{R}_p) \mathbf{1}^{(1 \times 1)}, \\
\text{s.t.} & \quad \sum_{i=1}^{n} \mathbf{A}_i \mathbf{x}_i - \mathbf{R}_p \mathbf{1}^{(s \times 1)} = \mathbf{0}, \quad \mathbf{R}_p \in \mathcal{R}_p, \\
& \quad \mathcal{F}_i(x_i) := \{ \mathbf{x}_i | \mathbf{y}_i = \mathbf{g}_i(x_i), \mathbf{x}_i \in \mathcal{X}_i \}, \forall i
\end{align}
\]  

where it is assumed that the site strives for a maximisation of the sum of the individual objectives (here, product sales \( \mathbf{p}_{i, \text{prod}}^\top \mathbf{y}_i \)) under the consideration of the additional costs or benefits that are caused by interfacing the pipelines with the constrained resource amounts given in \( \mathbf{R}_p \). We denote \( \mathbf{A} \circ \mathbf{b} \) as the Hadamard product (element-wise multiplication) and \( \mathbf{1}^{(s \times 1)} \) as a column vector with \( s \) identity entries. The matrix \( \mathbf{P} \in \mathbb{R}^{(m \times s)} \) contains the individual prices for \( m \) resources and \( s \) price regimes in the tariff (cf. Fig. 11) and \( \mathbf{R}_p \in \mathbb{R}^{(m \times s)} \) is the amount of resources that is exchanged with the pipelines for each resource at each price. The overall costs or benefits are then computed as \( \mathbf{P} \circ \mathbf{R}_p \). The constraints are on the one hand the network balance Eq. (4.18b), which should be closed at any time and on the other hand the individual constraint sets \( \mathcal{F}_i \) for each subsystem \( i = 1, 2, \ldots, n \) that result from technical limitations. In the site-wide formulation, the contribution of the subsystem \( i \) to the network balance is given by a vector \( \mathbf{A}_i \mathbf{x}_i \in \mathbb{R}^{(m \times 1)} \) that indicates the demand or supply, which is computed based on a tariff-independent model and where positive numbers indicate demands and negative numbers supplies. If the optimisation problem is decomposed by Lagrangian relaxation to set up a price-based coordination algorithm [56], a single vector of Lagrange multipliers with a single multiplier for each resource is not sufficient, since the site-hub can buy and sell resources at different rates depending on the amount that is taken from or sent to the pipelines. This leads to the following modelling approach.
Virtual splitting of the networks on the subsystem level

A standard Lagrangian relaxation of the problem stated in Eq. (4.18) can be written as the following optimisation problem that needs to be solved by each subsystem $i$

$$
\begin{align*}
\max_{x_i} & \quad p_i^T \prod_i - \lambda^T A_i x_i, \\
\text{s.t.} & \quad F_i(x_i),
\end{align*}
$$

(4.19a)

(4.19b)

where $\lambda$ is the vector of Lagrange multipliers which can be interpreted as the vector of prices that are associated with the demand or supply of the shared resources given by $A_i x_i$ that depends on the degrees of freedom $x_i$. To account for different price regimes in the portfolio contract, the constraints of the individual systems have to be extended. To model the different price regimes in a tariff, the networks can be split into virtual networks, i.e., each of the $m$ resource networks is split into $s$ virtual networks of the same resource. Here, without the loss of generality, we assume that for each resource $s$ price regimes exist. This results in the following formulation ($r_{i,j,l} \in \mathbb{R}$)

$$
A_i x_i = \begin{pmatrix} r_{i,1} \\ \vdots \\ r_{i,m} \end{pmatrix} = \begin{pmatrix} \sum_{l=1}^{s} r_{i,1,l} \\ \vdots \\ \sum_{l=1}^{s} r_{i,m,l} \end{pmatrix} = R_i \mathbb{I}^{(s \times 1)} = \begin{pmatrix} r_{i,1,1} & \cdots & r_{i,1,s} \\ \vdots & \ddots & \vdots \\ r_{i,m,1} & \cdots & r_{i,m,s} \end{pmatrix} \mathbb{I}^{(s \times 1)}.
$$

(4.20)

The optimisation problems for each subsystem are changed such that for each resource $j$ only one virtual network is allowed to be active. This is realised by a constraint adaptation, announced by the site-coordinator. The reasoning behind this is as follows: If the internal price of a shared resource exceeds the next higher external price (pipeline price in the tariff), the coordinator buys an amount of resource from the pipeline and allocates it to the internal shared resource network. From this point in time on, the subsystems that demanded shared resources receive an allocated amount of resource for the particular network. Afterwards, the amount they received is fixed and the cost is stored at the rate of the pipeline price. The next virtual network is activated for the next price coordination round. The reformulation involves the following constraint adaptation, where the resources from the preceding price regimes ($t_j - 1$, with $t$ as a vector storing the index of active price regimes ($1, 2, \ldots, s$) for each resource $j$) are frozen and the succeeding virtual networks, i.e., for $t_j < l$, are set to zero

$$
r_{i,j,l} \in \mathcal{R}^{(t)}_{i,j,l}, \quad \mathcal{R}^{(t)}_{i,j,l} = \begin{cases} \mathbb{R} & t_j = l \\ 0 & l > t_j \\ \hat{r}_{i,j,l} & \text{else} \end{cases},
$$

(4.21)

where $\hat{r}_{i,j,l}$ is the amount of the resource that has been bought during the last price regime by the coordinator and that has been assigned proportionally to the demanded amount of every subsystem. Further, the modelling involves the matrix $F^{(t)}$, $f_{j,l} \in [0, 1]$, which activates the respective virtual networks. The optimisation
problems for the subsystems are changed to
\[
\max_{\mathbf{x}_i, \mathbf{y}_i} \mathbf{p}_{i, \text{prod}}^T \mathbf{y}_i - \mathbf{I}^{(1 \times m)} (\mathbf{A} \circ \mathbf{R}_i) \mathbf{I}^{(s \times 1)}
\]
(4.22a)
\[
\text{s.t.} \quad \begin{align*}
    & \mathbf{r}_i - \mathbf{R}_i \mathbf{F}(t) \mathbf{I}^{(s \times 1)} = \mathbf{0} \\
    & r_{i,j,l} \in \mathcal{R}_{i,j,l}^{(t)}, \forall j, l
\end{align*}
\]
individual constraints, (4.22b)

where the manipulated variables of the subsystem are the vector \( \mathbf{u}_i \) and the matrix \( \mathbf{R}_i \). The vector of Lagrange multipliers \( \mathbf{\lambda} \) is extended to the matrix of prices \( \mathbf{\Lambda} \in \mathbb{R}^{(m \times s)} \).

**Proposed coordination scheme**

The proposed update scheme is formulated for buying resources only. The formulation for selling resources to external resources can be formulated analogously. The proposed coordination scheme receives the planned amounts of consumed and produced shared resources from the subsystems as input (see Algorithm 3). After a convergence check, the network balance is evaluated and it is checked whether the Euclidean norm of the balance is below a threshold \( \varepsilon \). The coordinator evaluates whether one of the internal prices exceeds the next higher pipeline price that is offered to buy additional quantities of resources. If the coordinator decides to buy additional resources at the pipeline price, the subsystems are queried again for the exact pipeline price, since \( \mathbf{\Lambda}_{j,t} \) can be higher than the pipeline price \( \mathbf{P}_{j,t} \), before the bought amount is assigned to the subsystems. The assigned quantities are calculated based on the demands of the receiving plants only, i.e., a producing plant is not assigned any resource. If the internal price is lower than the next higher price in the tariff, a normal sub-gradient update is performed. Whenever the coordinator buys an amount for a specific network, the virtual networks that are linked to the lower-priced tariff regimes are “frozen”, i.e., the Lagrange multipliers in \( \mathbf{\Lambda} \) and the received quantities \( r_{i,j,l} \) are fixed (see constraint in (4.21)).

**Algorithm 3 Price-based coordinator update with constraint adoption**

1. **procedure** \textsc{CoordinatorUpdate}([\mathbf{R}_1, \ldots, \mathbf{R}_n])

2. \( \mathcal{G} \leftarrow \sum_{i=1}^n \mathbf{R}_i - \mathbf{R}_\text{p} \)

3. if \( \|\mathcal{G}\|_2 < \varepsilon \) then
4. return;
5. end if
6. for \( j = 1, 2, \ldots, m \) do
7. if \( \mathbf{A}_{j,t} \geq \mathbf{P}_{j,t} \) then
8. \( \mathbf{\Lambda}_{j,t} \leftarrow \mathbf{P}_{j,t} \)
9. \( \mathcal{G} \leftarrow \sum_{i=1}^n \mathbf{R}_i (\mathbf{A}) - \mathbf{R}_\text{p} \)
10. \( r_{j,t} \leftarrow \min (\mathcal{G}_{j,t}, r_{j,t,p}) \)
11. for \( i = 1, 2, \ldots, n \) do
12. \( r_{i,j,t} \leftarrow (r_{i,j,t} \div \sum_{i=1}^n \max(r_{i,j,t}, 0)) \times r_{j,t} \)
13. end for
14. \( t_j \leftarrow t_j + 1 \)
15. \( \mathbf{F}_{j,t} \leftarrow 1 \)
16. else
17. \( \mathbf{\Lambda}_{j,t} \leftarrow \mathbf{\Lambda}_{j,t} + S\mathcal{G} \)
18. end if
19. end for
20. **end procedure**

**Illustrative example**

The optimisation problem is given in Eq. (4.22). The manipulated variables of each plant are limited to lie between 0 and 1 (lower and upper bound on the load). The numerical values for the tariff are given in Fig. 11.
Figure 12: Transfer prices against the number of iterations. At each diamond symbol the negotiated transfer price exceeds the next higher portfolio tariff at which the coordinator can buy from the external source. At the first two events, the coordinator is not able to buy enough resource to balance the networks (cf. Fig. 13) and thus the transfer price further raises after the action of buying externally. For the resource with the dashed line, at iteration 88, the coordinator buys again externally. However, at this point the acquired amount suffices to balance the network and the optimal transfer price corresponds to the portfolio price.

Figure 13: Network resource balance against the number of iterations. The events indicated with vertical bars correspond to the events shown in Fig. 12. At the first two events the externally available resource is not enough to balance the networks and consequently the network imbalance cannot be reduced to zero. At the third event the coordinator balances the network shown by the dashed line at once. The coordination algorithm then proceeds until the second network is balanced, too.

The model equations for the three plants are:

\[ f_1(x_1) = \begin{bmatrix} 32x_{1,1}^2 + 8x_{1,1} \\ 24x_{1,2}^2 + 3x_{1,2} \end{bmatrix}, \quad g_1(x_1) = \begin{bmatrix} 60x_{1,1} + 20 \\ 40x_{1,2} + 35 \end{bmatrix}, \quad p_{1,\text{prod}} = \begin{bmatrix} 5.4 \\ 2.1 \end{bmatrix}, \quad (4.23a) \]

\[ f_2(x_2) = \begin{bmatrix} 35x_{2,1}^2 + 15x_{2,1} \\ 37x_{2,2}^2 + 10x_{2,2} \end{bmatrix}, \quad g_2(x_2) = \begin{bmatrix} 50x_{2,1} + 10 \\ 45x_{2,2} + 2 \end{bmatrix}, \quad p_{2,\text{prod}} = \begin{bmatrix} 1.5 \\ 2.6 \end{bmatrix}, \quad (4.23b) \]

\[ f_3(x_3) = \begin{bmatrix} 25x_{3,1} + 15 \\ 30x_{3,2}^2 + 8 \end{bmatrix}, \quad g_3(x_3) = \begin{bmatrix} -25x_{3,1} - 15 \\ -20x_{3,2} + 20 \end{bmatrix}, \quad p_{3,\text{prod}} = \begin{bmatrix} 1.5 \\ 3.2 \end{bmatrix}, \quad (4.23c) \]

The results of the simulation study using Algorithm 3 with \( s_{i,j=i} = 0.7, s_{i,j\neq i} = 0, s_i \in S \) are shown in Fig. 12 and Fig. 13. It can be seen that when for some resource network the price-based coordination leads to a price that is higher than the next higher pipeline price (diamond symbols in Fig. 12), the coordinator buys the necessary (less or equal to the available) amount from the pipeline for this price (see vertical bars in Fig. 13). The network imbalance at these events is reduced until finally the network imbalance is below the threshold. Upon convergence of the algorithm the final cost a subsystem has to pay is the summation of the costs from each virtual network, i.e., the cost for a particular resource is determined from the amounts that are bought and received externally at different price regimes and the amount that is traded at the equilibrium price on site.
4.2.3 Conclusion and outlook

The problem of price-based coordination where the coordinator has access to external resources has been studied. In general, it can be said that distributed optimization can also be used for the described setup. Two different algorithms have been proposed: One integrates the access to the external resources into the update of the dual function by using bounds on the transfer prices in order to determine which price range has to be used. The second algorithm employs a coordinator that is aware of prices and quantities that are fixed in a portfolio tariff to handle the import of additional resources to the site. Both methods were demonstrated using illustrative examples. While the first method has better convergence properties for a large number of external markets and various different resources, the strength of the second approach is its transparency with respect to the different price regimes, i.e., each subsystem has consistent information how its overall cost function is computed. In both cases the confidential information of the individual systems remains private, which is one of the decisive advantages of price-based coordination.

Future work includes the investigation of the influence of a dynamic adaptation of the step size parameter $\alpha$, which does not require a priori knowledge about the convergence behaviour of the system by enabling arbitrary starting values for $\alpha$ and at the same time decreases the iterations until convergence. From a practical perspective, the distribution of the gains in the first proposed approach from the usage of external markets needs to be investigated, since in this approach these are only realised at the coordinator level. Another topic of interest is the implementation of other contract types, such as take or pay, minimum required amounts or discounts for large amounts.
5 CoPro applications

Within the project, the extensions of the previous chapter can be used in order to tackle the challenges from the CoPro use cases. In the following, two very different use cases are given as examples, where the aforementioned techniques can be applied. For the Covestro case the outline of the future application is given and for the Lenzing case the possible application of distributed optimisation is motivated in detail. The remaining use case that includes plants with shared resources by INEOS in Köln is methodologically similar to the Covestro use case, with the addition of buffer tanks and discrete decisions in the sub-systems.

5.1 Distributed optimisation and optimal procurement for the Covestro gas networks

Covestro operates networks for basic chemical gases on their site in Dormagen. Here, the networks for hydrogen and carbon monoxide are investigated. Aside from producing hydrogen as a side product in some production units, most of the hydrogen is purchased from external suppliers, where quantities of the gases can be purchased at different conditions. The carbon monoxide demand of Covestro can be fully covered by the external producers, where also different suppliers can be used. The networks with their external suppliers can be seen in Fig. 14.

![Figure 14: Gas networks at the Covestro sites Dormagen and Leverkusen.](image)

The Covestro internal consumers of the gases, i.e. the production plants, currently set the amount of raw material that they need for production. The networks are balanced using the produced hydrogen as well as the external suppliers, where the amounts necessary to bridge the difference between production and consumption are purchased. One idea in the CoPro project is to not only aggregate the flows of the gases, but to use distributed optimisation to coordinate them such that the overall production is improving the energetic and economic efficiency. This can be achieved by optimising across all Covestro internal participants of the gas networks in combination with an optimal procurement on the coordinator level, as introduced in Section 4.2.
5.2 Distributed optimisation in the Lenzing evaporation park

Lenzing has provided the CoPro consortium with a problem that is suitable to be handled via distributed optimisation. It involves the optimisation of two loosely coupled networks: the evaporation plants as a side-process that is devoted to fulfil the needs of the main fibre production process, and the cooling-water distribution network which feeds that evaporation network as well as other systems such as the heat recovery network, energy management or the waste-water treatment.

In the first stage, we focus on the evaporation network and its coupled interaction with the cooling-water distribution. Briefly, the problem is described as follows: The efficiency of an evaporation plant (Fig. 15.b) equipped with a surface condenser (SC) as cooling system (Fig. 15.a) depends on the spin-bath inflow (load), operation set-point (circulating flow through the heat-exchangers line and spin-bath temperature after the HE10 stage) and, importantly, on the cooling system performance, mainly determined by the temperature and flow of cooling water to the surface condensers (apart from the own nominal equipment efficiency by construction).

Hence, the more water is fed to the surface condensers the more efficient the evaporation plants are (less...
specific steam consumption). This saves live steam from boilers, but the water usage is limited due to environmental constraints. As the overall amount of available water is limited, its usage involves also a cost, given by negotiation with the energy and waste-water departments.

Independent optimisation problems have been set up for the load allocation and scheduling of maintenance operations on the one hand [57], [58], and for the cooling water distribution on the surface-condenser evaporators on the other hand, described below.

### 5.2.1 Cooling-water optimisation

There are $E = 20$ evaporators equipped with surface condensers as cooling systems. They are physically grouped in two subnets: one (SN1) composed of $E_{SN1} = 4$ plants and another (SN2) including the other $E_{SN2} = 16$. Given a water limit imposed by negotiation with other departments, the task here is to distribute the available cooling water among each subnet such that it ensures the lowest cost. It should be taken into account that all the SCs are connected in parallel and that SN2 can receive water from SN1 but not the other way around. The optimal distribution strongly depends on the efficiency of each plant (specific-steam consumption $SSC$), determined by the assigned evaporation capacity $EC$ and the fouling state $K_f$.

#### Problem formulation

The problem objective is minimising the trade-off between the cost of live steam and the water usage, which is given by the absolute steam consumpition ($ASC_e$) times its price ($P_{steam}$) plus the water flow ($F_e$) times its price ($P_{water}$). For the two sets of SC evaporators, the problem constraints are simple: the total flow in each subnet has to be lower than the maximum limit ($F_{S1}$), exceeding water can go from SN1 to SN2 but not backwards, upper and lower flow limits defined for each SC (suitable operation range to avoid problems of spin-bath entrainment into the SC) and the outlet water temperature ($T_{out}$) per plant has to be lower than the maximum allowed. Thus, the real-time optimisation problem is:

$$\begin{align*}
\min_{F_e, F_{N12}\in \mathbb{R}^{E+1}} & \quad J = \sum_{e=1}^{E} (ASC_e \cdot P_{steam} + F_e \cdot P_{water}) \\
\text{s. t.:} & \quad \sum_{e=1}^{E_{SN1}} F_e - F_{N12} \leq F_{S1} \\
& \quad \sum_{e=1}^{E_{SN2}} F_e - F_{N12} \leq F_{S2} \\
& \quad F_{N12} \geq 0 \\
& \quad F_e \leq F_e \leq \bar{F}_e \quad \forall e \in E \\
& \quad T_{out_e} \leq T_{max} \quad \forall e \in E
\end{align*}$$

Where $F_{N12}$ states for the water from SN1 to SN2 and $ASC_e = SSC_e \cdot EC_e$. The outlet water temperature $T_{out}$ is obtained with experimental models for each plant, depending on the water flow $F$, inlet temperature $T_{in}$ and fouling state $K_f$:

$$T_{out_e} = f(F_e) + T_{in_e} - K_{f_e}$$

The fouling parameter $K_f$ can be updated in real time using the actual water outlet temperature measurements $\hat{T}_{out}$ as $K_{f_e} = T_{out_e} - \hat{T}_{out_e}$.

Similarly to $T_{out}$, the $SSC$ is computed from experimental models by means of the current cooling power:

$$SSC = g(C_{pow})$$

$$C_{pow_e} = 4,18 \cdot (T_{out_e} - T_{in_e}) \cdot \frac{F_e}{3600}$$
The experimental models \( f(\cdot) \) and \( g(\cdot) \) are \( C^1 \) non-linear functions, usually quadratic polynomials, so the optimisation problem (5.1) is easily solved via NLP software. Note that the evaporation load for each plant \( EC_e \) is treated here as a known fixed value provided by the load-allocation optimisation, summarised next.

### 5.2.2 Load-allocation optimisation

The evaporation park is formed by the above 20 plants equipped with SC plus several new ones in which the SC have been replaced by cooling towers. Each plant has a different nominal capacity and efficiency. These plants need to fulfil the overall evaporation demands from the spinning fibre production process, existing different spinbath cycles (products) so that some plants can work in more than one cycle, but only in a single one at a time. The changeover of a plant from one cycle to another involves a cost and requires cleaning.

Here, the optimisation task is finding the plant-product allocation and load distribution among the plants that fulfils the evaporation demand per product with the lowest SSC. Apart from the nominal efficiencies, several factors affect the plants SSC: the evaporation load, the operating conditions (spin-bath temperature and circulating flow) the cooling system performance and the fouling in the heat exchangers HE1-10 (Fig. 15.b). This is a mixed production-maintenance scheduling problem which has been already formulated as a large mixed-integer linear programming (MILP) problem under some modelling assumptions [57]. Nevertheless, to illustrate the coupling with the water-distribution optimisation we will briefly describe the partial problem of the real-time plant-product assignment and load allocation.

**Problem formulation**

Given a set of \( p \in \mathcal{P} \) products to be processed in \( e \in \mathcal{E} \) evaporation plants, the problem is to allocate plants to products and then distribute the required total demand per product \( SP_p \) in a way that the overall ACS in the network is minimised. Two sets of decision variables are defined for this aim: variables \( X_{ep} \in \{0, 1\} \) which link the product \( p \) to plant \( e \) and real variables \( EC_{ep} \in \mathbb{R} \) defining the evaporation flow to be achieved in plant \( e \) processing the product \( p \), i.e. the evaporator load. Thus, the optimal allocation of products to plants is found by solving the following mixed-integer programming problem:

\[
\begin{align*}
\min_{X_{ep}, EC_{ep}} & \quad J = \sum_{e \in \mathcal{E}} \sum_{p \in \mathcal{P}} ASC_{ep} \\
\text{s. t.} & \quad \sum_{p \in \mathcal{P}} X_{ep} \leq 1 \quad \forall e \in \mathcal{E} \quad (5.4a) \\
& \quad \sum_{e \in \mathcal{E}} EC_{ep} \leq SP_p \quad \forall p \in \mathcal{P} \quad (5.4b) \\
& \quad EC_{ep} \leq EC_e \cdot X_{ep} \quad \forall e \in \mathcal{E}, \quad \forall p \in \mathcal{P} \quad (5.4c) \\
& \quad EC_{ep} \geq EC_e \cdot X_{ep} \quad \forall e \in \mathcal{E}, \quad \forall p \in \mathcal{P} \quad (5.4d) \\
& \quad X_{ep} = 0 \quad (e, p) \notin \mathcal{N} \quad (5.4f)
\end{align*}
\]

where \( EC \) and \( EC_e \) are the maximum and minimum capacities for each plant and the set \( \mathcal{N} \) states for the allowed physical connections between plants and products.

Note that the optimisation (5.4) becomes a MILP, MIQP or MINLP problem depending on the linear/non-linear nature of the surrogate models used to compute the ACS. For instance, if the following quadratic polynomials

\[
ASC_{ep} = \alpha_e \cdot EC_{ep}^2 + [\beta_e \cdot T_{out} + \delta_e \cdot C_{f_e} + \gamma_e] \cdot EC_{ep}
\]

are used (\( \alpha, \beta, \delta, \gamma \) parameters) [58], (5.4) is efficiently solved in a few seconds via MIQP. Note that, here, \( C_f \) is a fouling factor (updated online) which is different to \( K_f \) in previous section, as it describes for the fouling in the spin-bath heat exchangers. Note also that the cooling-water temperature at the SCs outlet \( T_{out} \) is treated here as a known value, provided either by the actual measurements or by the solution of (5.3).
5.2.3 Proposal for a joint distributed optimisation

In the above stated optimisation problems, the interaction between both systems is given by the $EC$ and $T_{out}$ variables. On the one hand, if both problems are solved independently as stated, treating these interactions as hard constraints fixed in advance, the overall optimisation becomes an iterative procedure (i.e., $T_{out}$ found by (5.1), is used for the optimisation (5.4) to compute $EC_e$, which will be used for (5.1) afterwards) with no global optimality guarantees. On the other hand, if both formulations are merged into a centralised optimisation, the problem becomes a MINLP, no matter which type of models are used in (5.5), as (5.1) is already non-linear and (5.4) involves discrete decisions. Moreover, (5.4) is addressing the partial problem of load allocation, but if we include the fouling prediction and formulate the joint production-maintenance scheduling (even using linear models) [57], the problem size increases considerably and the time to get a solution goes from a few seconds to several minutes. Therefore, the complexity of the centralised problem is much higher than the one for the two separate problems above, most likely practically intractable.

This is the motivation to move towards a globally optimal approach but keeping the solution times within acceptable values for real-time execution. Ongoing work is conducted to formulate the overall problem via the Lagrangian decomposition methods presented in the previous sections. In fact, $EC$ and $T_{out}$ are the variables $x$ involved in the shared-resource constraints (3.1c) with $N = 2$ and $r_e = 0$, so that the price-based coordination methods (Section 3), extended to include discrete decisions (Section 4.1), will be adapted and used for this problem.
6 Summary and future work

In this report, distributed optimisation of processing systems with shared resources was covered. The state of the art of techniques for distributed optimisation of plants with shared resources was summarised and two main concepts, Lagrange relaxation and ADMM were explained in detail. First results towards making the concept of distributed optimisation more applicable to the challenges arising from discreteness of decisions as well as how such schemes can be applied in the presence of external, non-participating providers or recipients that act using fixed prices was shown. Furthermore, it was shown how the methodologies and algorithmic developments from WP3 can be applied to the different use cases as defined in WP6. Thus, the work fully meets its originally planned objectives without any deviation.

Integrating the methods and algorithms to these use cases is ongoing work. The next steps include improving the methods and algorithms by investigating the following topics:

- Including more complex contracts in the coordinator update: The developed methods will be extended to handle more complicated constraints on the interaction with external markets. Since in the Covestro case, some of the external suppliers provide both gases at the same time, there are additional constraints on the ratios of the quantities that can be purchased. The resulting coupling between the networks will be investigated.

- Explicit consideration of uncertainties: Further studies address the general topic of uncertainty and how the schemes can be extended to result robust solutions. Deviations between model and plant, availability on external markets, and non-deterministic responses to different prices will be investigated with the aim to ensure convergence to feasible solutions.

- Increased flow of information between subsystems and coordinator: In some of the use cases, privacy concerns are not limiting and information about bounds on resources or active constraints can be exchanged. Hence, methods that integrate more information than the ones covered in Section 3.2 and 3.3 will be analysed and extended to suit the needs of the respective use cases.

Besides these algorithmic challenges, the integration of distributed optimisation for plants with shared resources into real industrial environments will be in the focus in the future.
References


Deliverable 3.4
First report on methods for distributed optimisation of plants with shared resources


