POLITECNICO DI MILANO
Corso di Laurea Magistrale in Ingegneria dell'Automazione
Dipartimento di Elettronica, Informazione e Bioingegneria

ENERGY MANAGEMENT OF A MULTI–BUILDING SYSTEM VIA DISTRIBUTED OPTIMIZATION

Relatore: Prof. Maria Prandini
Correlatori: Prof. Simone Garatti
Dott. Alessandro Falsone

Tesi di Laurea di:
Fabio Belluschi, matricola 801217

Anno Accademico 2014-2015

Ulteriora mirari,
praesentia sequi.

Sommario

L'ottimizzazione energetica degli edifici rappresenta un elemento cruciale nelle reti elettriche di nuova generazione, in cui gli utenti diventano parte attiva della gestione della rete grazie alla disponibilt di dispositivi di misura e attuazione e a reti di comunicazione per la trasmissione di misure e comandi (*smart grid*).

Si stima che quasi il 40% dell'energia elettrica totale consumata negli Stati Uniti e nell'area dell'Unione Europea venga utilizzata dagli edifici residenziali e commerciali, gran parte della quale viene impiegata dai sistemi di ventilazione e condizionamento per la regolazione termica dell'edificio. Le tecniche di ottimizzazione energetica permettono di modulare e spostare nel tempo i carichi energetici per migliore l'accoppiamento fra produzione e richiesta di energia. Ció consente sia di compensare la variabilitá intrinseca della produzione energetica da fonti alternative, la cui integrazione é fondamentale per un futuro energetico sostenibile, sia di ridurre i picchi di consumo, riducendo di conseguenza i costi di installazione e funzionamento degli impianti.

In questo lavoro di tesi viene studiato un approccio distribuito al problema di ottimizzazione energetica per una piccola rete (*microgrid*) di edifici dotati di impianto di condizionamento e che condividono un serbatoio termico. Vengono utilizzati opportuni modelli che rendono trattabile il problema di ottimizzazione perché convesso.

Ogni edificio rappresenta un agente che ottimizza un problema locale appositamente definito in modo da minimizzare il contributo totale al costo dell'energia elettrica del singolo agente, tenendo conto del vincolo di accoppiamento dato dal serbatoio termico grazie alle informazioni scambiate con gli altri agenti, secondo il grafo di comunicazione della rete. Il problema globale viene risolto con un algoritmo distribuito di consenso, e testato per diversi casi di studio, analizzando diverse configurazioni sia degli edifici che del grafo di comunicazione. I risultati vengono confrontati con quelli di una classica soluzione centralizzata. Il vantaggio della soluzione distribuita

consiste nella distribuzione del carico computazionale e nella garanzia di riservatezza delle informazioni relative ai consumi dei singoli edifici.

Abstract

In the perspective of the future evolution of the electrical power grid, leading to the *smart–grid* concept, buildings energy optimization plays a crucial role: almost 40% of the total electrical energy consumption in the United States and in Europe is given by residential and commercial buildings, a large part being consumed by HVAC (heating, ventilation and air conditioning) systems for the thermal conditioning of the building. Energy optimization techniques enable modulation and shift in time of electrical loads so as to improve the energy production–consumption matching. This allows both to compensate the intrinsic variability of the energy production from renewable sources, a key step towards a sustainable energy future, and to reduce the consumption peaks, with great economic impact on plants installation and maintenance costs.

In this thesis, we study a distributed approach to the problem of energy management for a district network (*microgrid*) comprising multiple buildings with their own chiller plant and sharing a thermal storage unit. We adopt convex models for the microgrid components to ease the optimization task.

Each building can be viewed as an agent in a network, optimizing its local problem where it minimizes its own contribution to the microgrid electrical cost while integrating the information exchanged with other agents, so as to account for resource sharing. The global problem is solved through a distributed consensus algorithm and tested for a variety of case studies, encompassing different configurations in terms of buildings and communication network structure. The results are compared with those obtained with a classic centralized solution. The advantage of the distributed solution is that computational load is distributed among the agents and privacy on the agent energy consumption profile is guaranteed.

Contents

Chapter 1

Introduction

1.1 General overview

The problem of efficient energy production, deployment and consumption is becoming a topic of major interest in the scientific community. With the energy demand continuously growing worldwide, the current model for mass production of energy, based primarily on fossil fuels, is rapidly becoming unsustainable [1], [2], due to resources depletion and environmental impact. It is estimated that in the United States energy production through fossil fuels causes almost 40% of human–related emissions of CO2, one of the components of the greenhouse gases most influencing World's climate changes, [3]. To mitigate these problems, scarcity of resources and environmental impact, modern energy policies focus on the reduction of dependence on carbon–based fuels. Increased dependence on alternative, renewable sources, such as solar and wind generated energy, as well as a more efficient energy consumption and the implementation of recovery systems, are fundamental steps for the future developments of the energy grid. The need of a "smarter" grid able to integrate different technologies and sources, while providing reliability, efficiency and security, led to the definition of the new power grid paradigm: the *smart–grid*, [4].

The concept of smart–grid refers to the modernization of the electrical power grid merging traditional systems with more recent computer–based technologies (automatic controls, smart metering and sensors, data communication networks, *etc.*) to enhance the performances of the whole energy production, distribution and consumption chain, to satisfy the energy demand in a "green" and sustainable way.

Although the smart–grid concept is widely accepted as the future goal of the energy grid, its realization poses a number of major challenges, [5]:

i) the integration of energy from renewable sources produced in large scale, *ii*) the integration of distributed generation, local storages and control systems, *iii*) the energy offer–demand balance and load control, while *iv*) providing efficiency, reliability and security (also in terms of cyber–security).

In contrast with the current scenario, based on traditional energy sources and centralized production, the smart–grid poses great a focus on the distributed production of renewable energy. In particular, heavy dependence on unconventional sources, as solar or wind generation, leads to a more variable energy generation due to intrinsic unpredictable nature of the sources. As for the distributed generation, like photovoltaic panels on rooftops, the introduction of a large number of sources may cause reliability problems to grid operations if not properly controlled and coordinated, [3].

Another topic of great interest is the optimization of the energy demand from the final consumer perspective, in particular load–shifting to off–peak periods. The energy supply–demand matching, *i.e.* the capacity to provide the energy requested in real time, is a fundamental requisite of the grid. The demand fluctuation over time causes load variations that affect systems cost and efficiency. In order to satisfy the instantaneous peak demand, plants must be oversized with respect to the baseline average request or, alternatively, reserves must be put in place, with great economical effort. Moreover, the production of the necessary additional energy occurs at a higher marginal cost, which are eventually passed down to the final users, [6]. Load control and off–peak shifting becomes a critical point for increasing systems energetic and economic efficiency.

In this context, buildings energy efficiency and optimal energy scheduling play a huge role. Commercial and residential buildings represent a large share of the primary energy demand, accounting for around 40% of the total demand both in the U.S., [7], [8], and in the EU, [9]. Almost half of this consumption is related to the thermal control of heating, ventilation and air conditioning (HVAC) systems, [1], [10], and this quota, strongly related to climate conditions, is even higher in countries with hot weather, as in the southern Europe region.

The energy request profile to the grid of a building, or a small network of buildings, can be greatly improved by introducing thermal energy storage (TES) systems that offer the possibility to store the thermal energy produced in off–peak periods and provides it during highly intensive work periods, thus reducing absorption peaks, [8], [11]. An analogous result is achieved using pre–cooling strategies that exploit building thermal inertia to shift loads in off–peak periods, [12], [6]. Both techniques results in a flattened energy demand profile with greatly reduced peaks.

Within the microgrid concept of a small, energetically independent network, buildings are no longer passive entities with the only role of consumers, but become active agents able to positively influence the network energy balance,[9], [10].

1.2 Work description

The present work focuses on the building energy management problem described in the general overview. More specifically, we focus on the cooling of a district network of buildings sharing a TES, integrating the results achieved in [10] and [13] in the microgrid context.

We adopt a distributed approach to tackle the overall energy management problem for the network. The traditional control architecture for microgrid energy management considers a central supervisor that performs optimization and dispatches decisions to local, passive agents that implements them. Despite the past success of this approach, it comes with a number of non negligible drawbacks, among which the difficulties to achieve a proper degree of fault tolerance and adaptability, [14]. Moreover, the growing complexity of the models and set–ups causes a growing computational effort, often unsustainable. In contrast, the distributed solution of the problem allows to build robustness of the network system against faults of single components, since decisions are taken locally, and the partition in a number of smaller problems helps to keep the local optimization computationally feasible.

Buildings are considered as active consumers: they are agents able to cooperate and communicate with each other to take part in the optimization process for the entire network. In the adopted set–up we have a single thermal energy storage (TES) unit which is shared among all the agents. The storage has the twofold role of both adding a further degree of freedom in the optimization process and of acting as connection node for the agents: the optimization is performed locally and each agent operates on its own temperature set–point to optimize its energy demand, considering other agents needs in the form of constraints on the available storage resource.

Following [10], we consider a high–level control system that abstracts away from the problem specific details and provides as input the optimized set–points for the building temperature and energy exchanges (charge \ discharge) with the storage. The approach entails the presence of a low–level control system able to promptly and precisely actuate the set–points decided by the higher level controller. This is a sensible approach as long as the temperature set–point changes not too rapidly, which can be accounted for when designing the set–point. The overall control system is drawn schematically in Fig. 1.1, with reference to a single building.

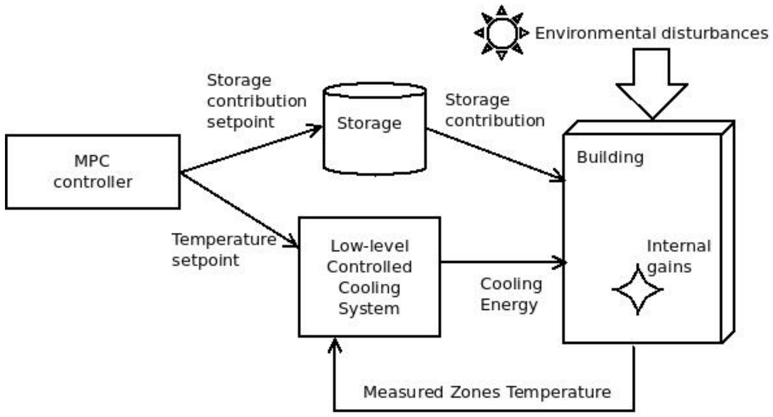

Environmental disturbances
Storage contribution setpoint
Storage
Storage contribution
MPC controller
Building
Temperature setpoint
Low-level Controlled Cooling System
Cooling Energy
Internal gains
Measured Zones Temperature

Fig. 1.1: Proposed high–level control scheme

The high–level control system puts in place an optimal control based on the building model, that describes the cooling energy need for tracking a certain temperature set–point, based on first principle equations.

The overall model, including chiller plant and storage unit, falls within the category of models that are convex in the control input. This feature entails the possibility to solve the control problem through convex optimization algorithms, thus enabling the solution of large problems in a relatively small time, [15].

Privacy protection As final remark, we briefly discuss the privacy issue. The centralized solution of the microgrid energy management introduces the need to share informations about energy consumption profiles. As stated in [16] and [17], these data can expose critical informations about the users and their typical behaviours, which should be protected.

Part of these informations, though, are critical for the correct working of the control system, as well as for other practical reason (*e.g.* billing from the energy supplier), causing contrast between privacy protection and optimal energy management.

The discussion on the future development of microgrids needs to take into account concern for the privacy violations, and strategies to incorporate privacy constraints in energy optimization problems are being studied nowadays, as in [18].

In this work the privacy issue is accounted for by adopting a distributed solution to the optimal energy management of the district network.

1.3 Thesis structure

The rest of the work is structured as follows:

- In Chapter 2 we describe the model of a building equipped with a chiller unit and a thermal energy storage unit, deriving the model equations in terms of energy provided\requested and energy conversion. Subsequently, the model is extended to a district of buildings sharing storage unit as a common resource.

- In Chapter 3 we define the energy management control problem in the form of a convex constrained optimization problem. We first address the single building set–up, then we define the microgrid energy optimization problem, in both a centralized and a distributed set–up.

- In Chapter 4 we describe the program developed to solve the problem. The chapter is divided in two sections: in the first one we describe the approach adopted to implement the model of the district network. In the second section, we provide a detailed description of the algorithm implemented to solve the microgrid energy optimization in a distributed fashion.

- In Chapter 5 we perform some simulations for testing the distributed control strategy on different case studies, accounting for a variety of situations and conditions, showing and commenting the results achieved.

- In Chapter 6 we finally draw some conclusions.

Chapter 2

Cooling of a district network: a convex model

In this chapter we deal with modeling of a multi–building system to the purpose of efficient energy management of the cooling process. In the first part, we consider a single building cooling system with thermal storage, giving a detailed explanation of the model, the characteristics of interest and the ones neglected, outlining its strengths and limitations in our problem framework. In the second part, we extend the model to a district network with multiple buildings sharing resources in a microgrid–like environment. A key feature of the adopted model is that it is convex in the control variables, which eases the control design phase.

2.1 Introduction

The building modelling process is characterized by a trade–off between two contrasting needs: accuracy in the description of the energy dynamics of the building and the energy exchanges involved, and a computational tractability. The latter in particular is of utmost importance, in view of the microgrid perspective of this work: the introduction of several buildings may critically increase the computational load, up to a point where the solution cannot be timely achieved. In this respect, convex programming can help in reliably solving problem up to a large scale with a "moderate" increase of computational time and effort. The above considerations broughts us to adopt the model for building energy dynamics proposed in [10] and [13], which is convex in the control variables.

2.2 Single building set–up

The building is subdivided in *zones*, where a zone is a “spatially defined part of a building, composed by one or more rooms, sharing well stirred and uniform air at the same temperature” ([10, p. 17]). The temperature of each zone is treated as a *control variable* and is subject to constraints in order to guarantee the desired level of comfort for its occupants.

Cooling energy is provided by a Heating, Ventilation and Air Conditioning (HVAC) plant. For this set–up we assume that there are no directly controllable variables for heating, thus we consider only the chiller system.

For improved flexibility, the building has access to a thermal storage unit. The amount of energy stored and the energy exchanges with the building are subject to some constraints on the maximum capacity of the storage and the maximum energy exchange rates.

The key model feature of the proposed set-up is that the desired profiles for zone temperatures are treated as *control inputs* for the system, in contrast with the most common approach of acting directly on the cooling system variables. As a consequence, comfort is enforced directly via constraints on the control variables. We assume that temperature variables perfectly track the imposed temperature profiles. This entails the presence of a lower level control system able to effectively track the set-points. Some additional constraints are enforced on the cooling energy request and thus indirectly on the sets of admissible temperatures and temperatures variation rate, in order to make it a reasonable assumption. Conversely, this assumption leads to a simplified model that neglects all the small details of the cooling system, and in the end is more suited to implement advanced control methods.

2.3 Building cooling system with thermal storage

The general problem of Section 2.2 is here formalized, describing the three main elements of the set–up, namely the building cooling process, the chiller plant and the thermal storage unit. These are modelled over the discrete time horizon, $[0 \cdots M]$, in order to suitably fit the mathematical framework of constrained optimization, thoroughly described in Chapter 3.

2.3.1 The building

In the perspective of optimal energy management, a building thermal model is constructed with reference to the discretized version $[0, 1\ \ldots\ M]$ of the original time horizon $[0,\, t_f]$. The building is composed of n_z thermally conditioned zones. The control input vector is given by $\boldsymbol{T_z} = [T_{z,1} \cdots T_{z,n_z}]^T$, where $T_{z,j}$ is the j-th zone temperature set–point, $j = 1 \dots n_z$.

The cooling energy needed to track T_z is defined as the sum of four energy contributions, *i.e.*:

$$\boldsymbol{E_c} = \begin{bmatrix} E_c(1) \\ \vdots \\ E_c(M) \end{bmatrix} = \boldsymbol{E_w} + \boldsymbol{E_z} + \boldsymbol{E_p} + \boldsymbol{E_i} \ , \tag{2.1}$$

where $E_c(k)$ is the amount of cooling energy needed at time step k. $\boldsymbol{E_w}$ is the wall–zone energy contribution that captures the structure thermal dynamics, $\boldsymbol{E_z}$ is the zone thermal inertia, $\boldsymbol{E_p}$ is due to the people occupancy and $\boldsymbol{E_i}$ is the internal energy contribution, that encompasses all the remaining relevant factors.

The chiller plant gets as input from the building the cooling energy request $\boldsymbol{E_c}$ and uses electrical energy to produce it.

Wall–zone energy contribution

The first step is to adequately capture the dynamics of the building cooling process, in order to correctly evaluate the energy exchange between walls and zone. To this end, we apply a one–dimensional finite elements model to each wall, since the thermal flow can be assumed to be perpendicular to its surface: each wall is divided into “slices”, or layers, along the heat flow direction. Every layer is made of a single material, so that all the properties are constant for the layer, has the same area of the wall, and has uniform temperature.

The inner layers exchange heat only with adjacent layers through the conduction mechanism; inner and outer boundary layers are instead subject to convective and thermal radiation energy exchanges with the (internal or external) environment. Let T_i be the temperature of the i-th layer. Then the i-th energy balance equation is given by:

$$\begin{aligned} \dot{T}_i = \frac{1}{C_i} \Big[& -(k_i^{i-1} + h_i^{i-1} + k_i^{i+1} + h_i^{i+1}) T_i \\ & + (k_i^{i-1} + h_i^{i-1}) T_{i-1} + (k_i^{i+1} + h_i^{i+1}) T_{i+1} \\ & + \alpha_i^{Lu} Q^{Lu} + \alpha_i^{Ld} Q^{Ld} + \alpha_i^{S} Q^{S} - \epsilon_i Q_r(T_i) + Q_{g,i} \Big] \ , \end{aligned} \tag{2.2}$$

where C_i is thermal capacity of the i-th layer; $T_{i\pm1}$ is the temperature of the adjacent layers; $k_i^{i\pm1}$ and $h_i^{i\pm1}$ are, respectively, the conduction and convection coefficient at the interface between the i-th layer and the adjacent layers. Q^S and Q^L are respectively the incoming shortwave and longwave radiation power, with α_i^S and α_i^L the corresponding absorbance coefficients; the longwave radiation can be separated into *upwelling* and *downwelling* radiation, Q^{Lu} and Q^{Ld} respectively. $Q_{g,i}$ is the thermal power generation inside the layer. Assuming the building to be a grey body, the term $Q_r(T_i)$ accounts for the emitted radiation of slice i, with ϵ_i being its emittance coefficient and it is governed by the Stefan–Boltzmann's law, $Q_r(T_i) = \sigma T_i^4$. We linearize it around $\bar{T}_i$, the i-th slice mean operating temperature, so that:

$$Q_r(T_i) = 4\sigma \bar{T}_i^3 T_i - 3\sigma \bar{T}_i^4 \tag{2.3}$$

which is affine in T_i. Note that $h_i^{i-1} \neq 0$ only for the inner boundary slice ($i = 1$) and $h_i^{i+1} \neq 0$ only for outer boundary slice ($i = m$); absorbance and emission coefficients satisfy $\alpha_i^S = \alpha_i^L = \epsilon_i = 0$, $\forall i : 1 < i < m$, given that conduction is the only heat convection mode for internal layers; finally, $k_i^{i-1} = 0$ for $i = 1$ and $k_i^{i+1} = 0$ for $i = m$.

The equations can be written in matrix form: consider a generic wall w composed of m layers, and let $\boldsymbol{T_w} = [T_{w,1} \dots T_{w,m}]^T$, where $T_{w,i}$, $i = 1 \dots m$, is the i-th slice of the wall w. Then, we can describe the overall wall dynamics with:

$$\dot{\boldsymbol{T}}_{\boldsymbol{w}} = \boldsymbol{A_w T_w} + \boldsymbol{B_w T_z} + \boldsymbol{W_w d} \tag{2.4}$$

where vector $\boldsymbol{d} = [T_{out}\ Q^S\ Q^L\ 1]^T$ is the vector of all the exogenous disturbances, with the last constant 1 added to account for the constant term deriving from the linearization in (2.3) and $\boldsymbol{A_w}$, $\boldsymbol{B_w}$ and $\boldsymbol{W_w}$ are suitable matrices that can be easily derived by inspecting the energy balance equations (2.2).

Finally, the wall–zone dynamics can obtained by properly stacking equations (2.4) for each of the n_w walls composing the building structure. Given the building state vector $\boldsymbol{T} = [\boldsymbol{T_1}^T \cdots \boldsymbol{T_{n_w}}^T]$, concatenation of single walls state vectors, we can write building temperatures dynamics equation:

$$\dot{\boldsymbol{T}} = \boldsymbol{AT} + \boldsymbol{BT_z} + \boldsymbol{Wd} \tag{2.5}$$

where $\boldsymbol{A}$ is a block–diagonal dynamic matrix built from $\boldsymbol{A_w}$; matrices $\boldsymbol{B}$ and $\boldsymbol{W}$ are obtained by simply stacking the wall corresponding vectors, *i.e.* $\boldsymbol{B} = [\boldsymbol{B_1}^T \cdots \boldsymbol{B_{n_w}}^T]$ and $\boldsymbol{W} = [\boldsymbol{W_1}^T \cdots \boldsymbol{W_{n_w}}^T]$.

The output transformation of the system is obtained considering the heat transferred from the walls to the zones, so that the thermal power exchanged from wall w, with surface S_w, to zone j is

$$Q_{w \to j} = S_w h_{w,b}^{b'} (T_{w,b} - T_{z,j}) \tag{2.6}$$

where $h_{w,b}^{b'}$ is the convective coefficient of the boundary layer b considered and the pair (b, b') assume values of either $(1, 0)$ or $(m, m+1)$ depending on the fact that the considered is the inner or the outer one. The total power $Q_{b,j}$ of zone j is obtained by summing the contribution of all the walls $\mathcal{W}$ adjacent to zone j:

$$Q_{b,j} = \sum_{w \in \mathcal{W}} Q_{w \to j} \ . \tag{2.7}$$

If we set $\boldsymbol{Q} = [Q_{b,1} \cdots Q_{b,n_z}]$ to be the output vector, then output equation can be expressed in terms of state vector $\boldsymbol{T}$ and input vector $\boldsymbol{T_z}$ as:

$$\boldsymbol{Q} = \boldsymbol{CT} + \boldsymbol{DT_z} \ , \tag{2.8}$$

with $\boldsymbol{C}$, $\boldsymbol{D}$ suitably defined matrices.

We finally obtain the continuous–time linear system for the building model combining equations (2.5) and (2.8):

$$\begin{cases} \dot{\boldsymbol{T}} = \boldsymbol{AT} + \boldsymbol{BT_z} + \boldsymbol{Wd} \\ \boldsymbol{Q} = \boldsymbol{CT} + \boldsymbol{DT_z} \end{cases} \tag{2.9}$$

The next step is the time discretization of the system over horizon $[0,\ t_f]$ in time slots of length dt. If we assume $\boldsymbol{T_z}$ and $\boldsymbol{d}$ to be linearly varying with each time–slot, we can apply a first–order discretization method and obtain (see Ioli [10] for the derivations):

$$\begin{cases} x(k+1) = \widetilde{A}x(k) + \widetilde{B}u(k) + \widetilde{W}w(k) \\ y(k) = \widetilde{C}x(k) + \widetilde{D}u(k) + \widetilde{H}w(k) \end{cases} \tag{2.10}$$

where $x(k)$ is the new discrete time state vector, $u(k) = \boldsymbol{T_{z,k}}$, $w(k) = \boldsymbol{d_k}$ and $y(k) = \boldsymbol{Q_k}$.

Defining $\boldsymbol{u} = [u^T(0) \cdots u^T(M)]^T$, $\boldsymbol{w} = [w^T(0) \cdots w^T(M)]^T$ and $\boldsymbol{y} = [y^T(0) \cdots y^T(M)]^T$ the temporal evolution of the signals in the discrete time horizon $[0,\ M]$, we can unroll the dynamics in (2.10) and obtain:

$$\boldsymbol{y} = F\boldsymbol{x(0)} + G\boldsymbol{u} + H\boldsymbol{w} \ . \tag{2.11}$$

Before continuing with the derivation of the energy exchange model, we make a digression on two important issues for the modellization: the model order reduction and the state estimation.

Model order reduction The application of finite elements method allows to capture the building thermal dynamics with high accuracy, but has the major drawback of generating a very high order model even for the most simple buildings. The consequence is a high computational demand that may affect the problem tractability. The issue is drastically accentuated by the model predictive control, as in general the computational load grows more than linearly with the length of the prediction horizon. Therefore, model order reduction is essential to keep an acceptable computational and the problem tractable. Techniques for model order reduction are a major area of interest in various field of application, and nowadays a variety of dedicated procedures have been developed. For more details on the order reduction algorithm implemented see Section 4.2.2 in Chapter 4.

State estimation The receding horizon policy requires the state to be known at each time–step, as shown by the term $\boldsymbol{x}(0)$ in eq. (2.15). Even if the original state was accessible, a temperature measure of the layers would be not representative of the new states after the order reduction. Therefore we need to resort on some kind of state reconstruction method. A first simple approach, as suggested in [10], can be adopted in two particular situations:

- Under stationary inputs and disturbances, the system reach the steady–state condition, where the state can be easily reconstructed from its dynamic equation:

  $$\bar{x} = (I - A)^{-1}(B\bar{u} + W\bar{d}) \ . \tag{2.12}$$

  However the steady–state condition is hardly achievable, due to the unrealistic assumption of stationary inputs and disturbances.

- Assuming that it's is possible to measure the heat Q_w exchanged between walls and zones and the values of $u(k)$ and $d(k)$, it's possible to retrieve the state from the system output equation. Inspecting the heat contributions equation $Q_w = Q_c - Q_z - Q_p - Q_i$, we notice that the only term that is not (directly or indirectly) measurable is the people generated heat Q_p. Nevertheless, during period with no occupancy, *e.g.* during night hours, the people heat contribution is zero. Moreover, in such periods is safe to assume the internal heat contribution, from machinery and lights, is neglectible. In this situations, all the terms in $Q_w \approx Q_c - Q_z$ and the exogenous signal are somehow quantifiable. Therefore, state can be reconstructed from output transformation as:

  $$x(k) = C^{+}(Q_w - Du(k) - Hd(k)) \ , \tag{2.13}$$

where C^+ is the pseudo–inverse of matrix C.

Another approach would be to adopt more refined estimation tools, like a state observer, although this choice would introduce additional complexity to the system.

We now resume the model derivation. Recalling the relation $y(k) = \boldsymbol{Q_k}$ and assuming that also the heat power exchange $\boldsymbol{Q}(t)$ varies linearly within each time–slot, we can compute the wall energy contribution for the time–slot k as follows:

$$E_w(k) = \frac{dt}{2}(\boldsymbol{Q_k} + \boldsymbol{Q_{k-1}}) = \frac{dt}{2}(\boldsymbol{y_k} + \boldsymbol{y_{k-1}}) \ . \tag{2.14}$$

Lastly, writing the above equation for every time–step in the horizon, $k = 1 \dots M$, and collecting them in matrix form we finally obtain the wall–zone energy contribution matrix:

$$\boldsymbol{E_w} = \begin{bmatrix} E_w(1) \\ \vdots \\ E_w(M) \end{bmatrix}_{[M\times 1]} = \widetilde{F}x(0) + \widetilde{G}\boldsymbol{u} + \widetilde{H}\boldsymbol{w} \ , \tag{2.15}$$

with $E_w(k)$ being the wall energy contribution in the k-th time interval.

Zones energy exchange

The zones energy contribution is related to the thermal inertia of the zone itself: zone temperature variations are indeed caused by energy exchanges, which must be taken into account. For the generic j-th zone, with thermal capacity $C_{z,j}$:

$$E_{z,j}(k) = -C_{z,j}(T_{z,j}(k) - T_{z,j}(k-1)) \ . \tag{2.16}$$

The j-th zone heat capacity $C_{z,j}$ is eventually rescaled to account for the heat capacity of the furniture and other elements occupying the zone volume.

Zone energy exchange matrix over the prediction horizon can be easily expressed as a function of the input profiles $\boldsymbol{u}$ by suitably define matrix Z such that:

$$\boldsymbol{E_z} = \begin{bmatrix} E_z(1) \\ \vdots \\ E_z(M) \end{bmatrix}_{[M\times 1]} = Z\boldsymbol{u} \ . \tag{2.17}$$

People energy contribution

The third contribution account for the heat generated by the presence of people during the occupancy hours; this contribution can become relevant in very crowded buildings (*e.g.* public buildings such as hospitals or offices).

The chosen empirical non–linear model (Butcher [19]) expresses the heat power of zone j as a function of the zone temperature $T_{z,j}$, and people occupancy profile over time, $n_p(t)$:

$$Q_{p,j}(t) = n_p(t)(p_2 T_{z,j}^2(t) + p_1 T_{z,j}(t) + p_0) \ . \tag{2.18}$$

The equation can be linearized around the mean operating temperature $\bar{T}_{z,j}$, discretized and, recalling the assumption of $T_{z,j}$ linearly varying within the time–slot, integrated to compute the people energy contribution in zone j in the k-th time–slot:

$$E_{p,j} = q_{2,k}(n_p)T_{z,j}(k) + q_{1,k}T_{z,j}(k-1) + q_{0,k} \ , \tag{2.19}$$

where coefficients are derived as in [13]. Collecting all the zones contributions into vector $E_p(k) = [E_{p,1}(k) \cdots E_{p,2}(k)]^T$ and evaluating them for the whole prediction horizon, we can finally define the people energy contribution vector:

$$\boldsymbol{E_p} = \begin{bmatrix} E_p(1) \\ \vdots \\ E_p(M) \end{bmatrix}_{[M\times 1]} = N(n_p)\boldsymbol{u} + e(n_p) \ , \tag{2.20}$$

where $N(n_p)$ and $e(n_p)$ are suitably defined matrices and they both depend on the occupancy profiles n_p.

Internal energy contribution

The last energy contribution to evaluate is the so–called internal energy, which is a term that encompass all other relevant heat sources that may significantly influence the building energy balance. Among the possible sources, we consider two in particular:

- The daylight radiation through the windows, which is associated only to the shortwave radiation through some coefficient that accounts for zone absorbance, windows transmittance, incidence angle, sun exposition, etc. Windows are usually shielded against longwave radiation, thus we neglect its influence.

- The heat generated by appliances, electrical equipment, internal lighting, etc. This is in turn due to two different factors: a constant term for the whole daytime and an additive term for the occupancies hours, supposing a more intense usage of equipment during those hours.

Given the above considerations, the internal thermal power for zone j is

$$Q_{int,j}(t) = \alpha_j(t)Q^S(t) + \kappa_j I_{\mathrm{R}^+}(n_p(t)) + \lambda_j \ , \tag{2.21}$$

where λ_j is the constant contribution of the electrical equipment, and $\kappa_j \cdot I_{\mathrm{R}^+}(n_p(t))$ the additional term during occupancy hours, $I_{\mathrm{R}^+}(n_p(t))$ being the indicator function that assumes value 1 when $n_p(t) > 0$ and 0 otherwise. The coefficient α_j is the product of the following factors: σ_j area of the windows, $\alpha_{s,j}$ absorbivity of the space, τ_j transmittance through windows, $K_j(t)$ general (time–dependent) coefficient for sun exposure, radiation incidence angle and windows shading effects.

As in the previous part, we can integrate the power equation to obtain the internal energy contribution $E_{int,j}(k)$ for the j-th zone during the k-th time–slot. Adding up all the zone contributions at time step k, and collecting the quantities for each time step in a single vector, we can finally obtain the internal energy contribution vector:

$$\boldsymbol{E_{int}} = \begin{bmatrix} E_i(1) \\ \vdots \\ E_i(M) \end{bmatrix}_{[M\times 1]} \quad . \tag{2.22}$$

2.3.2 The storage

The *thermal energy storage* (TES) is a system component able to accumulate thermal energy, store it and return it to the system when requested. Its presence introduces an additional degree of freedom in the optimization process, enhancing the control performance.

In this setup, *TES* real dynamics are disregarded because they introduce unnecessary complexity in both modelling and optimization: these dynamics are much faster than the building relevant dynamics. Moreover, only the amount of energy exchanged is of our interest, whilst the modalities used to retrieve energy are not relevant. This holds true as long as the TES system is properly regulated by a low-level layer of control, similarly to the building.

Due to the aforementioned considerations and assumptions, thermal storage can be treated as a blackbox system, with energy requested and released as input and output respectively. The simpliest system consistent with the above description is a first-order ARX model:

$$S(k+1) = aS(k) - s(k) \tag{2.23}$$

where $S(k)$ is the level of energy stored at the k-th time instant, $a \in (0, 1)$ is the loss coefficient introduced to model energy dispersion, $s(k)$ is the energy input to the system, with $s(k) > 0$ if the storage releases cooling energy and $s(k) < 0$ if the storage is recharged with additional thermal energy.

The storage dynamics over the control horizon $[0,\ M]$ can be written as a function of the initial level S_0 and the input profile $\boldsymbol{s} = [s(0) \cdots s(M-1)]^T$, as follows:

$$\boldsymbol{S} = \begin{bmatrix} S(1) \\ \vdots \\ S(M) \end{bmatrix}_{[M\times 1]} = S_a \cdot S_0 - S_b \cdot \boldsymbol{s} \ , \tag{2.24}$$

where S_a, S_b are the matrices of the unrolled dynamics, obtained from (2.23).

2.3.3 The chiller plant

The chiller plant uses electrical energy to satisfy the cooling energy request of the building jointly with the storage. The energy request to the chiller is equal to the cooling energy needed by the building to track the zone temperature profiles, minus the energy provided by the storage:

$$E_{ch}(k) = E_c(k) - s(k) \tag{2.25}$$

Obviously, the energy demand to the chiller is lower if the storage is contributing to the cooling ($s > 0$) and higher if the storage needs to be recharged ($s < 0$).

The chiller characteristic equation is based on the Ng-Gordon chiller model presented in [20]:

$$Q_l = \frac{a_1 T_0 T_{cw} + a_2(T_0 - T_{cw}) + a_4 T_0 Q_c}{T_{cw} - a_3 Q_c} - Q_c \ . \tag{2.26}$$

The model expresses the power absorbed by the chiller as a function of the cooling power request, and other parameters such as outdoor temperature T_0 and the cooling water temperature T_{cw}, a controlled variable set constant by a low–level controller. The model returns an efficiency curve with a maximum in correspondence of the nominal work load ad a rapid efficiency loss, due to overconsumption, moving away from it.

Obtaining an energy expression is not trivial due to the integration of the system non–linearities in the time–slot dt; so, for the sake of simplicity, the cooling power is approximated by its average over dt. This leads to the following expression of electrical energy consumption per time–slot:

$$E_l = \frac{a_1 T_0 T_{cw} dt + a_2(T_0 - T_{cw}) dt + a_4 T_0 E_{ch}}{T_{cw} - a_3 E_{ch}/dt} - E_{ch} \tag{2.27}$$

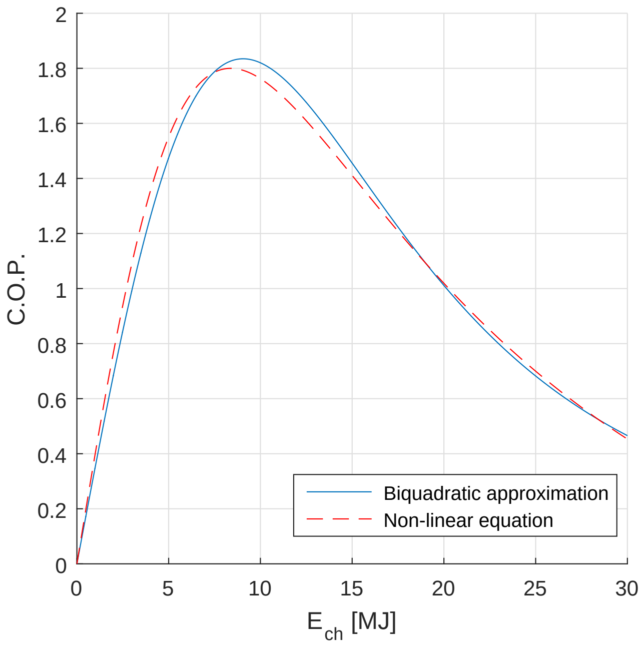


Fig. 2.1: Chiller COP – Gordon model and biquadratic approximation

where a_j, $j = 1 \dots 4$ are coefficients to suitably describe the chiller characteristic curve and T_{cw} is assumed constant. Outdoor temperature T_0 is set to $22\,^\circ C$, coherently with disturbances profiles used.

The convex optimization framework calls for a convex expression of the chiller curve, that adequately approximates the real model; the trade-off between high approximation accuracy, in particular near the efficiency curve maximum, and low computational cost leads to the choice of a biquadratic approximating function, in the form:

$$E_l = c_1(T_0)E_{ch}^4 + c_2(T_0)E_{ch}^2 + c_3(T_0) \tag{2.28}$$

The interpolating coefficient c_1, c_2 and c_3 are determined through a least-square criterion to achieve the best possible fit. Figure 2.1 shows the coefficient of performance (COP), E_{ch}/E_l, obtained via (2.27) and the biquadratic approximation in (2.28).

Consequently, the contribution over the control horizon $[0,\, M]$ of the electrical energy demand is:

$$\boldsymbol{E_l} = \begin{bmatrix} E_l(1) \\ \vdots \\ E_l(M) \end{bmatrix} = c_1 \cdot \begin{bmatrix} E_{ch}^4(1) \\ \vdots \\ E_{ch}^4(M) \end{bmatrix} + c_2 \cdot \begin{bmatrix} E_{ch}^2(1) \\ \vdots \\ E_{ch}^2(M) \end{bmatrix} + \begin{bmatrix} c_3 \\ \vdots \\ c_3 \end{bmatrix} . \tag{2.29}$$

2.4 Multi building set–up

We now consider a district network with multiple building, possibly sharing storage units and chiller plants. The modelling approach used is "modular", meaning that each building in the network can be seen as an instance of the building model developed in Section 2.3. Thus, each building is considered as a separate entity, with its own structure and parameters. A simple example of this type of microgrid–like environment is shown in Figure 2.2, where three buildings, each equipped with its own chiller, share a storage unit.

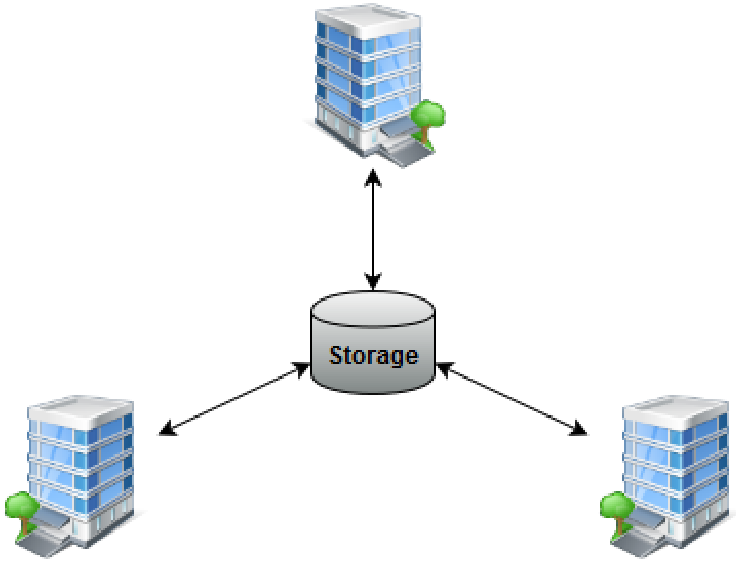


Fig. 2.2: A microgrid set–up example with 3 buildings and a shared storage unit

2.4.1 Discussion on the shared thermal storage unit

Choosing the correct size of a thermal storage unit is a non–trivial issue that must take into account a number of different factors, such as the energy demand profiles, the cooling system, the costs, the capacity required, *etc.*, trading–off over contrasting objectives. The lack of accurate sizing procedures often leads to incorrectly dimensioned storage unit [21]. Under-sizing the storage may affect the efficiency of the temperature control, either leading to a suboptimal energy cost management or impacting the comfort of the occupants. Conversely, over-sizing results in a higher initial investment, along with higher operating and maintenance costs throughout the unit life–cycle, affecting the cost–efficiency of the system.

To this end, the introduction of a single, shared storage can mitigate the impact of the dimensioning process on the overall cost of the TES system.

Sharing the storage unit has the additional advantage that it indirectly allows sharing the buildings chiller plants. Consider a building with its chiller plant. If the chiller is under–sized for the required work–load, then the building can be cooled with additional energy provided by the storage that was previously charged by another building equipped with a larger chiller. In turn, the buildings in the network need to reach an agreement on the usage of the shared thermal storage resource.

This has major impact the optimal energy management problem: the design of the control system must take into account the interaction of the cooling systems, as well as the need for buildings consensus on the storage usage. This leads to the introduction of two approaches to the network problem solution: *centralized* and *distributed* solution. Both approaches have different advantages and drawbacks with respect to computational load, scalability, communication network requirements and reliability in the eventuality of a local failure. These aspects will be discussed in Chapter 3.

2.4.2 Microgrid model definition

We consider the following model for a network of m interconnected buildings:

- Each building i is modelled in terms of cooling energy required as

  $$\boldsymbol{E_{c,i}} = \begin{bmatrix} E_{c,i}(1) \\ \vdots \\ E_{c,i}(M) \end{bmatrix} = \boldsymbol{E_{w,i}} + \boldsymbol{E_{z,i}} + \boldsymbol{E_{p,i}} + \boldsymbol{E_{int,i}} \ , \tag{2.30}$$

  where each of the four energy contribution are defined as in Section 2.3. Each building i has 2 sets of input variables, $\boldsymbol{T_{z,i}}$ and $\boldsymbol{s_i}$. The first represent temperature profiles of the zones of building i, the second is the profile for i-th building–storage energy exchanges.

- Each building is equipped with its own chiller plant and the energy required to the i-th local chiller is equal to the local cooling energy request $\boldsymbol{E_{c,i}}$ minus the local storage energy exchange $\boldsymbol{s_i}$, as stated in expression (2.25). The electric energy requested by the chiller is then given by (2.28).

- The storage unit is shared among all the buildings, and is modelled as a first–order discrete time system. The input vector at each time step is given by the summation over the m buildings of the energy exchange s_i:

  $$S(k+1) = aS(k) - \sum_{i=1}^{m} s_i(k) \ , \tag{2.31}$$

where $S(k)$ is the level of stored energy at the k-th time–step the parameter $a \in (0,\, 1)$ represents the (relative) energy losses of the storage at each time–step. By unrolling equation (2.31) over the control horizon $[0,\, M]$ we obtain an equation similar to (2.24).

Independently of the set–up (single or multi–building), convexity in the input variables is obtained and this will be of utmost importance in addressing the optimal energy management in Chapter 3.

Chapter 3

Optimal energy management control problem

In this chapter we formalize the control problem as a constrained optimization problem. The first part of the chapter tackles the single building case, recalling results of previous works. The second part deals with optimal control problem for a network of buildings, first in the classical form of a centralized problem and then proposing a distributed framework as an alternative, highlighting the differences between the two approaches.

3.1 Introduction

We next show how the control problem can be formulated as a convex constrained optimization problem, giving the possibility to include different objectives in the cost function and constraints on state and input variables.

3.2 Problem formulation for a single building

Consider the optimal energy management problem for a single building: the goal is to obtain the inputs profile that minimize the cooling energy cost over a receding finite time horizon $[0,\, t_f]$, while maintaining the desired level of comfort for the occupants. To this purpose the horizon is discretized in M time intervals of length dt, with

$$M = \frac{t_f - t_i}{dt} \ . \tag{3.1}$$

According to the model in Section 2.3, the control input vector for a building composed of n_z thermally conditioned zones is

$$\boldsymbol{T_z(k)} = \begin{bmatrix} T_{z,1}(k) \\ \vdots \\ T_{z,n_z}(k) \end{bmatrix}_{[n_z \times 1]} \quad , \tag{3.2}$$

where $T_{z,j}(k)$, $j = 1 \dots n_z$, is the j-th zone average temperature set–point at time k. Thus, the goal is to achieve the optimal profile $\boldsymbol{T_z(k)}$, $k = 1, \dots, M$.

The cost function to be minimized is the total electrical energy cost over the prediction horizon, defined as:

$$J = \sum_{k=1}^{M} \Psi(k) E_l(k) \ , \tag{3.3}$$

where $\Psi(k)$ is the unitary cost of energy in the k-th time–slot and $E_l(k)$ is the electric energy required in the same time–slot. The latter is determined through the chiller characteristic curve as a function of the cooling energy $E_{ch}(k)$ requested to the chiller, which is the energy required to cool the building, $E_c(k)$, minus the energy provided by the storage. More precisely:

$$E_l(k) = c_2 E_{ch}^4(k) + c_1 E_{ch}^2(k) + c_0 \ , \tag{3.4}$$

where

$$E_{ch}(k) = E_c(k) - s(k) \ . \tag{3.5}$$

Before defining the optimal control problem, we address our solution to the state estimation problem.

3.2.1 State estimation

The initial state estimation is a key issue in the problem definition: in Section 2.3 we showed how initial state influences the model energy matrices and therefore the optimization problem constraints, as expressed by equation (2.15):

$$\boldsymbol{E_w} = \widetilde{F}\boldsymbol{x(0)} + \widetilde{G}\boldsymbol{u} + \widetilde{H}\boldsymbol{w} \ . \tag{3.6}$$

Moreover, as mentioned in the “Model order reduction” paragraph of Section 2.3.1, the model order reduction procedure returns a (reduced) set of states that lose correspondence with the original (full) states physical meaning, *i.e.* are no longer wall temperatures, significantly impeding the definition of a realistic set of admissible values.

Let us consider now the deterministic case, in which a nominal forecast of the disturbances profiles for the next 24 hours is available. Under the assumption that such disturbances are periodic with a one day period, it is sensible to look for an optimal periodic control strategy. To this end, we enforce additional constraints on the input $\boldsymbol{u}$, on the states $\boldsymbol{x}$ and on the storage level $\boldsymbol{S}$, namely:

$$\boldsymbol{u(M)} = \boldsymbol{u_0} \tag{3.7a}$$

$$\boldsymbol{x(M)} = \boldsymbol{x_0} \tag{3.7b}$$

$$S(M) = S_0 \tag{3.7c}$$

where $\boldsymbol{u_0} = \boldsymbol{u(0)}$, $\boldsymbol{x_0} = \boldsymbol{x(0)}$ and $S_0 = S(0)$ represent, respectively, the input, state and storage level initial values. Analogously, $\boldsymbol{u(M)}$, $\boldsymbol{x(M)}$ and $S(M)$ are the input, state and storage level final values, respectively.

The optimality of the proposed control strategy strongly relies on the periodicity assumption. If the disturbances are characterized as stochastic processes, a notion of stochastic periodicity should be put in place, which is beyond the scope of this work.

3.2.2 The optimization problem

The cooling energy required to track a given temperature set–point is determined through the building model developed in Chapter 2 and expressed in matrix form over the prediction horizon [0, M]. Thus the optimization problem can be formalized as follows:

$$\begin{array}{llrcll}
\mathcal{P}_1 : & \displaystyle\min_{(\boldsymbol{u},\boldsymbol{s},\boldsymbol{x_0},S_0)} & \multicolumn{4}{l}{\boldsymbol{\Psi} \cdot \boldsymbol{E_l}(\boldsymbol{u},\boldsymbol{s})} \\
& \text{s.t.} & \boldsymbol{u_{min}} & \leq & \boldsymbol{u} & \leq \boldsymbol{u_{max}} \\
& & |\boldsymbol{s}| & \leq & s_{max} & \\
& & 0 & \leq & \boldsymbol{S} & \leq S_{max} \\
& & \boldsymbol{E_c} & \geq & 0 & \\
& & \boldsymbol{E_l}(\boldsymbol{u},\boldsymbol{s}) & \leq & E_{max} & \\
& & \boldsymbol{u(M)} & = & \boldsymbol{u_0} & \\
& & \boldsymbol{x(M)} & = & \boldsymbol{x_0} & \\
& & S(M) & = & S_0 &
\end{array} \tag{3.8}$$

The objective is to minimize the cost function J expressed as the scalar product between vectors $\boldsymbol{\Psi}$ and $\boldsymbol{E_l}$, *i.e.* the sum over every k-th time instant of the electrical energy demand $E_l(k)$ times the unitary energy cost $\Psi(k)$ (see (3.3)). The energy cost vector $\boldsymbol{\Psi} = [\Psi(1) \cdots \Psi(M)]$ contains the cost per time–slot over the horizon, and $\boldsymbol{E_l} = [E_l(1) \cdots E_l(M)]^T$ is the collection of electrical energy requests.

The set of optimization variables includes of the zone temperature set–points $\boldsymbol{u} \equiv \boldsymbol{T_z}$, the vector of energy exchanges with the storage, $\boldsymbol{s}$, the system initial state and the storage initial level. Hence, the full set is given by the composition of the following subsets:

$$\boldsymbol{T_z} = \begin{bmatrix} \boldsymbol{T_z}(1) \\ \vdots \\ \boldsymbol{T_z}(M) \end{bmatrix}_{[(M \cdot n_z) \times 1]} \tag{3.9a}$$

$$\boldsymbol{s} = \begin{bmatrix} s(1) \\ \vdots \\ s(M) \end{bmatrix}_{[M \times 1]} \tag{3.9b}$$

$$\boldsymbol{x_0} = \begin{bmatrix} x_1(0) \\ \vdots \\ x_{n_s}(0) \end{bmatrix}_{[n_s \times 1]} \tag{3.9c}$$

$$S_0 = S(0) \tag{3.9d}$$

The enforced constraints have, respectively, the following meaning:

- A set of comfort constraints for each zone and for each time instant, expressed as lower and upper bounds, $\boldsymbol{u_{min}}$ and $\boldsymbol{u_{max}}$, on the zone temperature set–points:

$$\boldsymbol{u_{min}} = \begin{bmatrix} \begin{bmatrix} u_{min,1}(1) & \cdots & u_{min,n_z}(1) \end{bmatrix}^T \\ \vdots \quad \ddots \quad \vdots \\ \begin{bmatrix} u_{min,1}(M) & \cdots & u_{min,n_z}(M) \end{bmatrix}^T \end{bmatrix}_{[(M \cdot n_z) \times 1]} , \tag{3.10a}$$

$$\boldsymbol{u_{max}} = \begin{bmatrix} \begin{bmatrix} u_{max,1}(1) & \cdots & u_{max,n_z}(1) \end{bmatrix}^T \\ \vdots \quad \ddots \quad \vdots \\ \begin{bmatrix} u_{max,1}(M) & \cdots & u_{max,n_z}(M) \end{bmatrix}^T \end{bmatrix}_{[(M \cdot n_z) \times 1]} . \tag{3.10b}$$

- A set of constraints for the storage energy exchange rate per time–slot. s_{max} is the maximum amount of energy that can be exchanged in a single time interval; it originates from technological limitations of the energy exchange process. The absolute value implies that limits are the same both for the energy drawn from the storage (positive limit for $s > 0$) and the energy provided to it (negative limit for $s < 0$); in general, these limits can be different, and the constraint must be modified accordingly.

- A set of constraints on the maximum and minimum level of the storage, where S_{max} is its maximum capacity and $\boldsymbol{S} = [S(1) \cdots S(M)]^T$ is the amount of stored energy on all time intervals. The lower bound is set to 0 since the energy stored cannot be negative; however it can be different from 0 (greater) if, for technological or practical reasons, the storage is required to be non–empty at every time instant.

- In the system under consideration there are no heat sources to control, so that the cooling energy request by the building, $\boldsymbol{E_c}$, must be non–negative at each time instant.

- A set of constraints on the maximum amount of electrical energy that the chiller may request, since its cooling energy production is upper bounded.

- Finally, the sets of equality constraints are the closure constraints on the initial temperature set–point, the initial state and the initial energy stored previously mentioned in Section 3.2.1, introduced to guarantee the problem periodicity assumption.

3.2.3 Discussion and observations

In Chapter 2 we pointed out the importance of the convexity of the model to ease control design. In order to be convex, the optimization problem must satisfy the requirements of *i*) convexity of cost function, *ii*) convexity of inequality constraints and *iii*) affinity of equality constraints. The cost function is an affine transformation of the convex function (3.4), thus is convex. As for the constraints, they are either linear or convex by the expression derived in Section 2.3, thus satisfying properties *ii*) and *iii*). Therefore problem (3.8) is indeed a convex optimization problem.

3.3 Control problem for a multi building network

The second part of this chapter treats the multi–building district case. First we address the problem in a classical centralized fashion, defining a problem comprising the entire district and the interaction between the buildings, treated as a whole. Then we propose a different approach, *i.e.* a *distributed* approach, formalizing a set of local problems which are solved individually by each building, while exchanging informations with the other buildings, so as to reach a consensus. The consensus solution should be identical to the centralized one, making the centralized and distributed problems equivalent.

3.4 Centralized solution

The centralized optimization problem can be expressed in a form equivalent to the single case problem. A proper redefinition of the cost function, optimization variables and constraints sets is required, while the general problem remains analogous to (3.8).

Given a network of interactive buildings as described in Section 2.4, we recall that each of them can be modelled as an instance of the single building model, while the storage is a single shared resource, described by equation (2.31). The storage works as an active element of the network, contributing to the energy cost optimization.

Optimization variables set Consider a network consisting of m buildings, and let j be the generic j-th building, $j \in \{1, 2, \ldots, m\}$. The control goal is to obtain the optimal temperature profiles for each zone of each building, as well as the optimal profiles for their energy exchanges with the TES unit. This means that the optimization variables are given by the collection of all temperatures $\boldsymbol{T_{z,j}}$ and storage variables $\boldsymbol{s}_j$, $\forall j$:

$$\overline{\boldsymbol{T}}_{\boldsymbol{z}} = \begin{bmatrix} \boldsymbol{T}_{\boldsymbol{z},1} \\ \vdots \\ \boldsymbol{T}_{\boldsymbol{z},m} \end{bmatrix} \tag{3.11a}$$

$$\bar{\boldsymbol{s}} = \begin{bmatrix} \boldsymbol{s}_1 \\ \vdots \\ \boldsymbol{s}_m \end{bmatrix} . \tag{3.11b}$$

The input variables vector has dimension $M \cdot \sum_{j=1}^{m} n_{z,j}$, with $n_{z,j}$ being the number of thermal zones for the j-th building. The storage variables vector has dimension $M \cdot m$.

Consistently with the single building problem, the optimization variables include the initial state of each building and the initial level of the storage:

$$\bar{\boldsymbol{x}}_{\mathbf{0}} = \begin{bmatrix} \boldsymbol{x}_{\mathbf{0},1} \\ \vdots \\ \boldsymbol{x}_{\mathbf{0},m} \end{bmatrix} \tag{3.12a}$$

$$S_0 = S(0) \ , \tag{3.12b}$$

where vector $\bar{\boldsymbol{x}}_{\mathbf{0}}$ has dimension $\sum_{j=1}^{m} n_{s,j}$, where $n_{s,j}$ is the number of (reduced) states for building j, and S_0 is a scalar quantity, unvaried from the single building case.

The bar notation is used to identify the global variables and quantities.

Cost function The cost function to minimize is the total energy cost for the microgrid, given by the sum of the local energy cost of each building. Thus, it can be obtained by summing up the local cost functions J_j:

$$J = \sum_{j=1}^{m} J_j = \sum_{j=1}^{m} \boldsymbol{\Psi}_j \cdot \boldsymbol{E}_{l,j}(\boldsymbol{u}_j, \boldsymbol{s}_j) \ , \tag{3.13}$$

where $\boldsymbol{\Psi}_j$ and $\boldsymbol{E}_{l,j}$ are, respectively, the energy cost profile and the electrical energy request for building j over the control horizon $[0,\ M]$.

Typically, the energy cost profiles are identical, $\boldsymbol{\Psi} = \boldsymbol{\Psi}_1 = \ldots = \boldsymbol{\Psi}_m$. So, equation (3.13) can be rewritten as:

$$J = \boldsymbol{\Psi} \cdot \sum_{j=1}^{m} \boldsymbol{E}_{l,j}(\boldsymbol{u}_j, \boldsymbol{s}_j) = \boldsymbol{\Psi} \cdot \bar{\boldsymbol{E}}_{\boldsymbol{l}} \ , \tag{3.14}$$

where the total energy cost is the scalar product of the overall energy request of the network with the energy cost profile. The total energy request vector $\bar{\boldsymbol{E}}_{\boldsymbol{l}}$ is obtained by properly summing up the local energy contribution $\boldsymbol{E}_{l,j}$ components along the prediction horizon:

$$\bar{\boldsymbol{E}}_{\boldsymbol{l}} = \begin{bmatrix} \sum_{j=1}^{m} E_{l,j}(1) \\ \vdots \\ \sum_{j=1}^{m} E_{l,j}(M) \end{bmatrix} \ . \tag{3.15}$$

Energy quantities The electrical energy request is a function of the cooling energy request according to the chiller curve as expressed by equations (3.4) and (3.5). In the general case, each building is equipped with a different chiller with different parameters, thus leading to:

$$\boldsymbol{E_{l}}_{,j} = c_{2,j}\boldsymbol{E}^4_{\boldsymbol{ch},j} + c_{1,j}\boldsymbol{E}^2_{\boldsymbol{ch},j} + c_{0,j} \tag{3.16}$$

$$\boldsymbol{E_{ch}}_{,j} = \boldsymbol{E_{c}}_{,j} - \boldsymbol{s}_j \tag{3.17}$$

$$\boldsymbol{E_{c}}_{,j} = \boldsymbol{E_{w}}_{,j} + \boldsymbol{E_{z}}_{,j} + \boldsymbol{E_{w}}_{,j} + \boldsymbol{E_{i}}_{,j} \tag{3.18}$$

defined individually for each building j, $j \in \{1, 2, \dots, m\}$. The various energy contribution in (3.18) are individually derived from the model for the single building as in Section 2.3.

Thermal storage dynamics The thermal storage for the multi–building case is governed by equation (2.31), here recalled:

$$S(k+1) = aS(k) - s_{sum}(k) \ ,$$

where the sum of energy exchanges for the single time slot has been rewritten as $s_{sum}(k) = \sum_{i=1}^{m} s_i(k)$.

Given an initial condition S_0 on the storage level, it is easy to derive the matrix form for the storage unrolled dynamics over horizon [0, M]:

$$\begin{bmatrix} S(1) \\ S(2) \\ S(3) \\ \vdots \\ S(M) \end{bmatrix} = \begin{bmatrix} a \\ a^2 \\ a^3 \\ \vdots \\ a^M \end{bmatrix} S_0 - \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ a & 1 & 0 & \ddots & 0 \\ a^2 & a & 1 & \ddots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a^{M-1} & a^{M-2} & a^{M-3} & \dots & 1 \end{bmatrix} \begin{bmatrix} s_{sum}(1) \\ s_{sum}(2) \\ s_{sum}(3) \\ \vdots \\ s_{sum}(M) \end{bmatrix} \tag{3.19}$$

which can be rewritten in a more compact form, with an obvious meaning of the symbols, as:

$$\boldsymbol{S} = S_a \cdot S_0 - S_b \cdot \boldsymbol{s_{sum}} \ . \tag{3.20}$$

The difference with equation (2.24) for the single building case lies in the input vector $\boldsymbol{s_{sum}}$ instead of $\boldsymbol{s}$, which of course must account for contributions of the multiple buildings.

3.4.1 Centralized optimization problem formulation

The above discussion naturally leads to the following formulation of the centralized optimization problem:

$$
\begin{array}{llrcll}
\mathcal{P}_2 : & \displaystyle\min_{(\bar{\boldsymbol{u}},\bar{\boldsymbol{s}},\bar{\boldsymbol{x}}_{\boldsymbol{0}},S_0)} & \multicolumn{4}{l}{\boldsymbol{\Psi} \cdot \overline{\boldsymbol{E}}_{\boldsymbol{l}}(\bar{\boldsymbol{u}}, \bar{\boldsymbol{s}})} \\
& \text{s.t.} & \boldsymbol{u}_{\boldsymbol{min},j} & \leq & \boldsymbol{u}_j \ \leq \ \boldsymbol{u}_{\boldsymbol{max},j} & j = 1, \ldots, m \\
& & |\boldsymbol{s}_{\boldsymbol{j}}| & \leq & s_{max} & j = 1, \ldots, m \\
& & 0 & \leq & \boldsymbol{S} \ \leq \ S_{max} & \\
& & \boldsymbol{E}_{\boldsymbol{c},j} & \geq & 0 & j = 1, \ldots, m \\
& & \boldsymbol{E}_{\boldsymbol{l},j}(\boldsymbol{u}_j, \boldsymbol{s}_j) & \leq & E_{max,j} & j = 1, \ldots, m \\
& & \boldsymbol{u}_{\boldsymbol{j}}(\boldsymbol{M}) & = & \boldsymbol{u}_{\boldsymbol{0},\boldsymbol{j}} & j = 1, \ldots, m \\
& & \boldsymbol{x}_{\boldsymbol{j}}(\boldsymbol{M}) & = & \boldsymbol{x}_{\boldsymbol{0},\boldsymbol{j}} & j = 1, \ldots, m \\
& & S(M) & \geq & S_0 &
\end{array}
\tag{3.21}
$$

The centralized control problem $\mathcal{P}_2$ recalls the structure of problem (3.8).

The constraints enforced in problem (3.21) are identical to those in problem (3.8). The difference lies in how these constraints are enforced to account for multiple building and a shared storage:

- Comfort constraints on the temperatures of each zone at every time–step must hold true for each building. Buildings (offices, houses, ...) have possibly different constraints.

- Energy exchanges with the storage are bounded for every building at every time instant. Hence, the set of constraints $|\boldsymbol{s}_{\boldsymbol{j}}| \leq s_{max}$ implies the enforcement of $M \cdot m$ scalar constraints of the kind

  $$|s_j(k)| \leq s_{max} \ , \ \forall j \ , \ \forall k \ .$$

  Once again, the absolute value defines the symmetry of the constraint for positive and negative exchanges.

- Storage level is bounded by the storage unit maximum and minimum capacity. The constraint is the exact replica of the analogous constraint in the single building problem (3.8), although now vector $\boldsymbol{S}$ of the stored energy is determined through equation (3.20).

- Cooling energy request must be non–negative, *i.e.* no heating is allowed, at every time–step. This must hold true for each building.

- Electrical energy drawn from the network is limited by the chiller unit size and maximum capability. The constraint is enforced separately for each j-th building due to the different characteristics of the chillers.
- Lastly, the closure constraints on initial state, initial temperature zone and initial storage value, analogous those in the single building problem, are enforced for each building $j \in \{1, 2, \dots, m\}$.

Coherently with the model assumptions of Section 2.4, comfort constraints and actuation constraints due to chillers can be enforced locally, while the storage constraints are dealt with globally.

Finally, notice that the closure constraint on the initial storage value S_0 in problem (3.21) is defined as an *inequality* constraint, changed from the corresponding *equality* constraint in (3.8). The motivations are related to the algorithm for the solution of the distributed problem and will be further addressed in Section 3.6.2.

3.4.2 Discussion and observation

As discussed in Section 3.2.3, the constrained optimization problem (3.21) must satisfy certain conditions on the cost function and the constraints set in order to be convex. Convexity of inequality constraint and affinity of the equality constraints, are satisfied for the same reasons of problem (3.8): as mentioned above, the types of constraints are unchanged and the same consideration can be made. As for the cost function, it is defined as the sum of convex functions, hence convexity is preserved. Therefore, the optimization problem (3.21) is convex.

A further analysis of the problem gives an important perspective on its structure. Due to modellization of Section 2.4, problem (3.21) result composed of m *almost* independent problems:

- The cost function $J = \sum_{j=1}^{m} f_j(\boldsymbol{u}_j, \boldsymbol{s}_j)$ is the sum of m independent functions $f_j(\cdot)$, each depending explicitly only on the corresponding local variables $\boldsymbol{u}_j$ and $\boldsymbol{s}_j$.
- The majority of constraints can be enforced locally for each building, as made explicit by the j indexing in (3.21). However, the problem are not completely independent due to the coupling introduced on the

storage exchanges variables $\boldsymbol{s}_j$, $\forall j$ by constraint

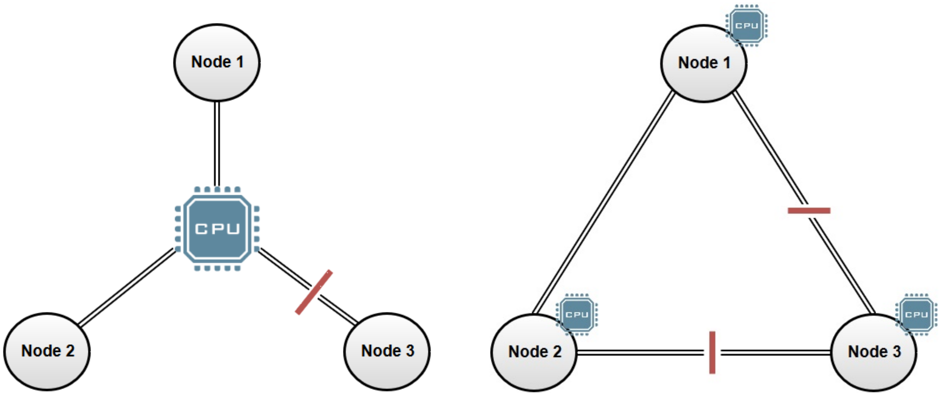
$$0 \;\leq\; \boldsymbol{S} \;\leq\; S_{max} \;,$$

according to equation (3.20).

Each agent can be given a local cost function depending on local variables, which are coupled only by a subset of constraints. Hence, the centralized constrained optimization problem belongs to the class of *almost separable* optimization problems, [22]. This makes a strong case for the solution of the problem in a distributed fashion.

3.5 Distributed control problem

The set–up described in the previous section heavily relies on the computational power of some central unit in order to effectively solve problem (3.21). Considering networks with a large number of nodes, the up–scaling in the dimensionality leads to both huge datasets, that needs to be stored, and extremely high computational effort to compute a solution. This issue can still be manageable for small networks, but it worsen with the number of nodes, making a centralized approach either not applicable or, at least, very inefficient and *de facto* unsustainable.



(a) Centralized set–up **(b)** Distributed set–up

Fig. 3.1: Network failures example for centralized and distributed set–up

Moreover, as remarked in [23], the technological realization of these networks make them unreliable for establishing a centralized computation for the solution. Consider a network subject to the failure of some communication arcs: the resulting graph is made of two non–connected components, as in Figure 3.1, and the information flow between the components is interrupted. In the centralized set–up, *e.g.* as in figure 3.1.a, the isolated node does not receive any inputs from the controller, hence the node remains uncontrolled for the whole duration of the failure, which can lead to serious problems for the system or the users. Viceversa, in the decentralized set–up (figure 3.1.b), the node is equipped with its own computational unit. Therefore, even in the case of failure, the isolated node is able to achieve control by solving the local problem. The solution achieved by the two component are sub–optimal with respect to the global optimal solution, due to the lack of information flow. Nevertheless, the two subsystems are (separately) controlled, granting some degree of robustness.

Therefore, the distributed approach is a suitable alternative to overcome the limitations of computational resources and to grant robustness against uncertainty of the network topology, either because of network failures or because of evolution of the network itself (addition or removal of nodes, changes in the neighbours sets, ...).

3.5.1 Problem overview

Consider the following formalization for a constrained optimization problem, as presented in [24]:

$$\begin{aligned} \min_{x} \quad & \sum_{j=1}^{m} f_j(x) \\ \text{s.t.} \quad & x \in \bigcap_{j=1}^{m} X_j \end{aligned} \tag{3.22}$$

Given a time–varying network composed of m agents, problem (3.22) shows the general framework for constrained convex optimization problem where the m agents cooperates to achieve the optimal solution in a distributed fashion.

For every agent $j = 1, \dots, m$, $f_j(\cdot) : \mathbb{R}^n \to \mathbb{R}$ and $X_j \in \mathbb{R}^n$ denotes the local objective function and the local constraint set of the i-th agent, that constitute the set of private information for agent j. The vector of decision variables $x \in \mathbb{R}^n$, instead, represent the set of shared information.

As discussed at the end of the previous section, the centralized optimization problem is composed by a set of almost separable local problems, coupled by the storage constraints and variables, which suggest the use of a distributed approach for the solution. We will now reformulate problem (3.21) in order to match the problem framework (3.22).

For simplicity of notation, in the following discussion vector $\bar{\boldsymbol{s}}$ is redefined to include all the storage variable, *i.e.* the storage exchanges and the initial storage value. In the same way, we redefine the (local) building variables vector to include both the temperature set–point and the optimal initial storage:

$$\bar{\boldsymbol{s}} = \begin{bmatrix} \boldsymbol{s_1} \\ \vdots \\ \boldsymbol{s_m} \\ S_0 \end{bmatrix}_{[(M \cdot m+1)\times 1]} \tag{3.23a}$$

$$\boldsymbol{u_j} = \begin{bmatrix} \boldsymbol{T}_{\boldsymbol{z},j} \\ \boldsymbol{x_0}_{,j} \end{bmatrix}_{[(M \cdot n_{z,j}+n_{s,j})\times 1]} \qquad j = 1, \ldots, m \tag{3.23b}$$

3.5.2 Problem formulation

Let us consider the problem (3.21), and consider a fixed profile for the storage variables $\bar{\boldsymbol{s}}$, namely $\bar{\boldsymbol{s}}^*$.

The resulting problem is composed of m *decoupled* optimization problems parametric in $\bar{\boldsymbol{s}}^*$. They can be solved separately for every j-th building, so that the global optimal solution is given by the collection of the local ones, and the global cost is the sum of the local cost function:

$$\sum_{j=1}^{m} \min_{\boldsymbol{u_j} \in \, U_j(\bar{\boldsymbol{s}}^*)} J_j(\boldsymbol{u_j}, \bar{\boldsymbol{s}}^*) \tag{3.24}$$

where, for every building $j = 1, \ldots, m$, the set $U_j(\bar{\boldsymbol{s}}^*)$ is the set of local constraints, namely the constraints on the temperature set–points, on the energy requests and on the initial state. The local cost function $J_j(\boldsymbol{u_j}, \bar{\boldsymbol{s}}^*)$ is the scalar product of the energy cost profile $\boldsymbol{\Psi}$ and the local electrical energy request, as expressed in equation (3.16).

If we now let $\bar{\boldsymbol{s}}$ vary, optimizing the storage profiles so as to minimize the district global energy cost, it is possible to rewrite problem (3.21), combining with the results derived in (3.24), in the following forms:

$$\begin{aligned} \min_{\bar{\boldsymbol{s}}} \quad & \sum_{j=1}^{m} \min_{\boldsymbol{u_j} \in U_j(\bar{\boldsymbol{s}})} J_j(\boldsymbol{u_j}, \bar{\boldsymbol{s}}) \\ \text{s.t.} \quad & \bar{\boldsymbol{s}} \in \mathcal{S} \end{aligned} \tag{3.25}$$

and, redefining the local cost function $J_j(\boldsymbol{u_j}, \bar{\boldsymbol{s}})$ as $f_j(\bar{\boldsymbol{s}})$, we finally obtain:

$$\begin{array}{ll} \min\limits_{\bar{\boldsymbol{s}}} & \sum\limits_{j=1}^{m} \quad f_j(\bar{\boldsymbol{s}}) \\ \text{s.t.} & \bar{\boldsymbol{s}} \in \mathcal{S} \end{array} \quad . \tag{3.26}$$

where $\mathcal{S}$ is the set of constraints on the global optimization variables, namely the storage level constraint, the constraints on the energy exchanges with the storage and the initial storage value constraint.

We showed how problem (3.26) is equivalent to the centralized problem (3.21). Moreover, with the proper assignments, namely

$$x = \bar{\boldsymbol{s}} \tag{3.27a}$$

$$f_j(x) = f_j(\bar{\boldsymbol{s}}) \tag{3.27b}$$

$$X_j = \mathcal{S} \qquad , \qquad \forall j = 1, \ldots, m \tag{3.27c}$$

problem in (3.26) is identical to (3.22). Hence, we finally obtained a formulation for the district optimal energy management problem amenable to be solved in a distributed fashion.

3.5.3 Discussion and observation

In literature there are a variety of algorithms for the solution of distributed constrained optimization. A general overview on the operating principles of the distributed algorithms is the following:

- For each agent, a first instance of the problem is generated, based on some appropriate initialization procedure. The optimization problem is solved to achieve the first local optimal solution.
- The solution, or part of it, along with other relevant data are communicated to neighbours agents through the network.
- Agents evaluate the new information received, properly weights them and redefine the local optimization problem accordingly to account for them.
- The updated problem is solved to obtain a refined solution which accounts also for other agents needs.
- The procedure is repeated until some sort of agreement is reached among every agent, which represent the general solution to the problem.

The details of the procedure, such as which informations and data are transmitted through the network, the modalities to weight and incorporate them in the local problem, and the methods to compute the solution, depend on the technical specifications of the algorithm implemented. The algorithm adopted in this work is the *distributed optimal consensus via proximal minimization algorithm* (see [24]).

In the next section, the algorithm and its general working principle will be presented, while the technical implementation details will be discussed in Chapter 4.

3.6 Distributed optimal consensus algorithm

The distributed optimal consensus algorithm, [24], falls into the family of algorithms for distributed convex optimization. It allows to account for time–varying networks, possibly different constraint set for each agents and uncertainty, providing a unified framework for problems usually treated separately. Moreover, under some assumptions for the problem structure and the communication network, the algorithm is guaranteed *i*) to reach convergence, *ii*) to reach consensus among the agents and that *iii*) the consensus solution the agents agreed on is (one of) the problem optimal solution.

For the problem under consideration, is of particular interest the algorithm ability to deal with time–varying networks, granting some degree of robustness to temporary network failures, as in a node (or a set of nodes) no longer connected to the network for a finite time interval. As stated in Section 3.5, adapting to evolutions of the network or to its (partial) failure is one of the major motivations for shifting from a centralized to a distributed set–up, hence the choice of an algorithm robust enough to adapt to these unforeseen events. and different constraints set for the agents,

Hence, this algorithm allows an exhaustive study of the problem under the most general conditions, namely a district of different buildings communicating over a non–fixed graph.

3.6.1 Algorithm description

The algorithm, outlined in Algorithm 1, is an iterative procedure to solve convex optimization problems in the form (3.22).

At generic iteration k, agent i generates a local optimization problem, accounting for the informations known from the previous iteration. This problem is solved to compute the local solution $x_i(k + 1)$. Then, shared

Algorithm 1 Distributed, optimal consensus

```
Initialization
    $k = 0$
    Consider $x_i(0) \in X_i$ , for all $i = 1, \dots, m$.
For $i = 1, \dots, m$ repeat until convergence
    $z_i(k) = \sum_{j=1}^{m} a_j^i(k) x_j(k)$
    $x_i(k+1) = \arg\min_{x_i \in X_i} f_i(x_i) + \frac{1}{2c(k)} \|z_i(k) - x_i\|^2$
    $k \leftarrow k + 1$
```

knowledge is transmitted through the network according to its topology, and solutions from other neighbours agents are received and used to update the value of $z_i(k)$, a local weighted average of all the information received. At the successive iteration, $(k+1)$-th iteration, a new local problems is generated using the previous solution $x_i(k)$ and the updated weighted average $z_i(k)$. The procedure is repeated until convergence is reached for each agent individually: an appropriate convergence condition is thus required.

Weighted average

For each agent i, the weighted average $z_i(k)$ is the term that accounts for information from other agents at the k-th iteration, combining their estimates $x_j(k)$ properly weighted as:

$$z_i(k) = \sum_{j=1}^{m} a_j^i(k) x_j(k) \ , \tag{3.28}$$

where $a_j^i(k) > 0$ is the *positive* weight agent i assign to the information received by agent j, including its own previous solution for $j = i$.

The weight coefficients define the network graphs: each non–zero coefficient $a_j^i(k) > 0$ establish, at iteration k, a directed arc from node j to node i, adding it as a neighbour node, while a weight $a_j^i(k) = 0$ implies the absence of communication between the two nodes.

More formally, the network is represented by a directed graph $(V,\, E_k)$ where

$$V = \{1, \dots, m\} \tag{3.29a}$$

$$E_k = \big\{(j,\, i) \colon a_j^i(k) > 0\big\} \tag{3.29b}$$

are respectively the set of nodes and the set of directed edges. Furthermore, we are interested in the pair of nodes that communicates directly infinitely often, defined by the set of edges:

$$E_\infty = \{(j,\, i)\colon (j,\, i) \in E_k \text{ for infinitely many } k\} \tag{3.30}$$

Corrected cost function

The cost function in each local problem is a modified version of the original local cost function, in the form:

$$J_i(k) = f_i(x_i(k)) + \frac{1}{2c(k)} \|z_i(k) - x_i(k)\|^2 \tag{3.31}$$

At iteration k, the cost function consist of 2 terms: *i*) the local cost function $f_i(x_i)$ over the local variables, and *ii*) a corrective term, penalizing the "distance" between the optimal local solution and the (weighted) neighbours estimates. The latter is the term that drives the different local solutions towards the consensus global solution. The distance is defined as the (squared) euclidean norm of the vector computing the element-wise difference between the optimal solution and the weighted average.

The corrective term is weighted through a coefficient $\frac{1}{2c(k)}$, with $c(k)$ being an iteration–dependant value. The series of coefficient $c(k)$ strongly influences the convergence properties and convergence rate of the algorithm. Hence, it becomes a crucial design parameter, subject to strict conditions on its value to guarantee the convergence of the algorithm (see Assumption 4 in the following section).

Algorithm assumptions

The framework provided by [24] poses some requirements and assumptions on the structure of the problem and on the communication network in order to provide the convergence, consensus and optimality features described above.

We will now report the assumptions, briefly explaining the meaning and impact on the algorithm.

Assumption 1

For each $i = 1, \dots, m$, the function $f_i(\cdot)\colon \mathbb{R}^n \to \mathbb{R}$ is convex, and $X_i \subseteq \mathbb{R}^n$ is a convex set.

Assumption 2

For each $i = 1, \dots, m$, $X_i \subseteq \mathbb{R}^n$ is compact.

Assumption 3

The feasibility region $\bigcap_{i=1}^{m} X_i$ has a non–empty interior, *i.e.* there exists $\bar{x} \in \bigcap_{i=1}^{m} X_i$ and $\rho \in \mathbb{R}_+$ such that $\{x \in \mathbb{R}^n \colon \|x - \bar{x}\| \leq \rho\} \subset \bigcap_{i=1}^{m} X_i$.

Assumption 4

Assume that for all $k \geq 0$, $c(k) \in \mathbb{R}_+$ and $\{c(k)\}_{k \geq 0}$ is a non–increasing sequence, *i.e.* $c(k) \leq c(r)$ for all $k \geq r$. Moreover,

1. $\lim_{k \to \infty} c(k) = 0$,
2. $\sum_{k=0}^{\infty} c(k) = \infty$,
3. $\sum_{k=0}^{\infty} c(k)^2 < \infty$.

Assumption 5

There exists $\eta \in (0,\, 1)$ such that for all $i, j \in \{1, \dots, m\}$ and all $k \geq 0$, $a_j^i(k) \in \mathbb{R}_+$, $a_i^i(k) \geq \eta$ and $a_j^i(k) > 0$ implies that $a_j^i(k) \geq \eta$. Moreover, for all $k \geq 0$,

1. $\sum_{j=1}^{m} a_j^i(k) = 1$ for all $i = 1, \dots, m$,
2. $\sum_{i=1}^{m} a_j^i(k) = 1$ for all $j = 1, \dots, m$.

Assumption 6

The graph $(V,\, E_\infty)$ is strongly connected, *i.e* the for any two nodes there exists a path of directed edges that connects them. Moreover, there exists $T \geq 1$ such that for every $(j,\, i) \in E_\infty$, agent i receives information from a neighbouring agent j at least once every consecutive k iterations.

Assumption 1 define the class of problems for which the algorithm results are derived.

Assumption 2 and Assumption 3 imposes some constraints on the problem structure, in order to guarantee some characteristics (Lipschitz continuity, existence of the solution) needed to demonstrate the convergence and optimality features.

Assumption 4 is necessary to guarantee that the algorithm will reach convergence for $k \to \infty$. As suggested in [24], the generalized harmonic series family satisfies the hypothesis above, and generally $c(k)$ is chosen among that family.

Assumption 5 provide a uniform lower bound for the coefficients of the weight matrix (4.5), implying that informations from neighbours are mixed at a non–decreasing rate. Furthermore, it assure that the weighted average (3.28) is a convex function of the informations received.
Refer to [24] for the complete discussion and proofs.

3.6.2 Algorithm implementation

In the previous section we provided a general picture of Algorithm 1 for the solution of convex optimization problem in the form (3.22), with a description of its assumptions and its working principle. We now further discuss the fitting of the distributed energy management problem into the general algorithm framework.

Recalling the variables assignment (3.27):

$$\begin{aligned} x &= \bar{\boldsymbol{s}} \\ f_i(x) &= f_i(\bar{\boldsymbol{s}}) \\ X_i &= \mathcal{S} \qquad , \qquad \forall i = 1, \dots, m \end{aligned}$$

we already proved that is possible to match the distributed problem (3.26) and the general problem (3.22). In particular, the distributed set–up satisfies Assumptions 1, 2, and 3 on the problem structure: *i*) the local cost functions $f_i(\bar{\boldsymbol{s}})$ are convex (see Section 3.2.3), *ii*) the set of global constraints $\mathcal{S}$ is convex and compact, and *iii*) the set of global constrains $\mathcal{S} = \bigcap_{i=1}^{m} X_i$ has a non–empty interior. The last point leads to an interesting discussion on the storage closure constraint (3.7c), that will be addressed in Section 3.6.2.

On the optimization variables

As stated in Section 3.5.1, the set of optimization variables is split locally into two subsets, which constitute respectively the subsets of local and global variables.

The first, $\boldsymbol{u_i}$, $i = 1, \dots, m$ consists of the zone temperatures set–points, extended with the initial state $\boldsymbol{x_{0,i}}$, which belongs to the set $U_j(\bar{\boldsymbol{s}})$. This set contains all the energy requests, zones temperature comfort and the initial values closure constraints.

The second subset, of global variables $\bar{\boldsymbol{s}}$, consist of all the building–storage exchange profiles with the addition of the initial storage level, S_0. $\bar{\boldsymbol{s}}$ belongs to set of global constraints $\mathcal{S}$, which includes the constraints on maximum exchange rate s_{max}, the constraint on the minimum and maximum storage level and the storage closure constraint.

To reach agreement, the algorithm relies on the penalization of the distance between the solution computed by different agents. The agreement needs to be achieved on the global variables, that refer to a shared resource. The local variables instead can be freely optimized locally, given the restriction on the feasibility set imposed by the consensus, as expressed by the dependence of $U_i(\bar{\boldsymbol{s}})$ on vector $\bar{\boldsymbol{s}}$.

Therefore, the penalty term in the modified cost function (4.2) takes into account only the storage variables:

$$\frac{1}{2c(k)}\left(\|\boldsymbol{z_i}(k) - \bar{\boldsymbol{s}}\|^2\right) , \tag{3.32}$$

where $\boldsymbol{z_i}$ is the weighted average (3.28), computed by the i-th building, based on the information exchanged in the previous iteration.

On the storage closure constraint

Lastly, we will make some remarks on the initial storage constraint.

In Section 3.2.1 we introduced the constraint (3.7c) on the initial storage value S_0, to assure the periodicity, imposing the equality of the initial and the final values. The same constraint also appears in the multi–building problem, both in the centralized and the distributed framework. In particular, in problem (3.26) the constraint belongs to the set $\mathcal{S}$ of the global constraints.

Consider now Assumption 3 of the algorithm: it requires a non–empty interior for $\bigcap_{i=1}^{m} X_i$. Recalling the variables assignment (3.27), it translates in the requirement of a non–empty interior for set $\mathcal{S}$. Hence, the storage closure constraint in the equality form (3.7c) violates Assumption 3.

However, consider the constraint:

$$S(M) \;\geq\; S_0 \; . \tag{3.33}$$

Substituting equality constraint (3.7c) with (3.33), the set $\mathcal{S}$ satisfy the non–empty interior condition, and the problem structure assumptions are all satisfied by the distributed problem (3.26).

Notice that, due to the nature of the energy management problem, this change has no effect on the final solution. The storage initial level can be interpreted as "free" energy given to the system at the beginning of the day. The closure constraint imposes that *at least* the same amount of energy must be returned to the storage at the end of the day. Recharging the storage represents an additional energy cost for the system, and a potential mismatch between S_0 and $S(M)$ would be an unnecessary additional cost. The minimization process drives the inequality constraint to be active, hence constraints (3.7c) and (3.33) are *de facto* the same.

The change of the equality constraint to an inequality one has been made also for the centralized control problem (3.21), although not needed, to keep it consistent with its distributed counterpart (3.26). The observation that constraint (3.33) will always be active, made for the distributed set–up, still holds true for the centralized set–up.

As a final remark, we observe that the other closure constraints of (3.7), namely the equality of initial and final state, and the equality of initial and final temperature set–points, have not been modified.

In fact, in problem (3.26) these constraints belongs to the local set $U_i(\bar{\boldsymbol{s}})$, $i = 1, \dots, m$, which is not subject to the non–empty interior condition. Therefore, for $\boldsymbol{x_0}$ and $\boldsymbol{u_0}$ it is legitimate to impose closure constraints in the equality form.

Chapter 4

Simulator

In this chapter we describe the simulator developed to implement the optimal control problem of Chapter 3. We highlight the main features achieved and their importance in the context of microgrid optimization problem, as well as the limitations of the tool. The first part is dedicated to the implementation details of the model and the solution to the constrained optimization problem. The second part is dedicated to the implementation details of the consensus algorithm for the multi–building district problem.

4.1 Introduction

A fundamental part of this work was the development of a software tool for the implementation and solution of the optimal control problem of Chapter 3. The software has three main requirements: first, it has to construct the building model in the form described in Section 2.3; second, it has to use the model built and the constraints specifications to generate the constrained optimization problem in form of Section 3.2; third, it has to solve the problem to obtain the optimal input profiles.

As for the first point, it becomes a crucial design aspect: from a user perspective, the specifications are given in term of building structure, walls disposition and materials, zones subdivision. Thus is necessary to provide a user–friendly interface for these inputs, easily accessible yet enough detailed, while procedures for the second and third aspects are automatized.

The second part is dedicated to the extension for the multi–building network described in Section 2.4 and the implementation of consensus algorithm, highlighting the choices and the adjustment needed to apply the theory of Section 3.6.2 in a numerical case study.

4.2 Inputs for the building definition

The building energy model is generated from the balance equations (2.2), based on the building structure, the wall disposition and a set of parameters. From a user perspective though, dealing with a large set of equations would result in a tedious and error–prone process, especially when considering the multi–building set–up.

Therefore, the main guideline followed is to develop a toolbox–like program working through a systematic approach. The user works on a "high–level" description, abstracting from the specific details of the model and optimization problem which are automatically tackled by the procedures underneath, allowing for a simple and immediate definition of a wide variety of building types and structures, yet maintaining a high degree of detail for the model and high customization capability.

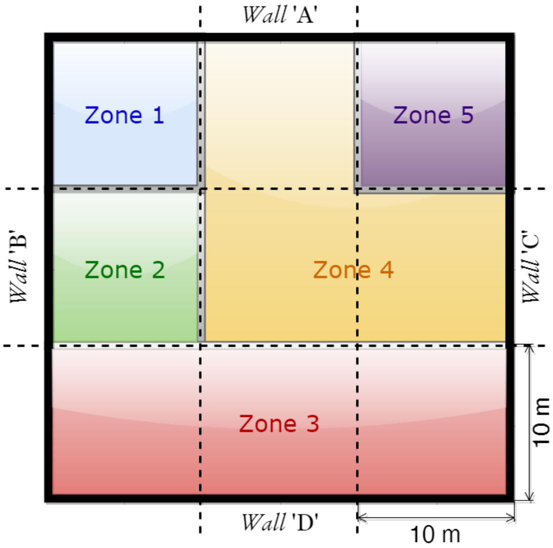


Fig. 4.1: Example building layout

4.2.1 Building structure description

The key feature is a *modular* description of the building, so that it can be represented as the composition of "modules" of equal dimensions, containing all the informations on the adjacent walls. An example of a building composed by 10 $m \times 10$ m modules is shown in figure 4.1.

The composition of these modules serves both to define the building structure and to instantiate the wall properties. In this way the building layout, the zones subdivision, the walls disposition and their properties are fully defined by the correct disposition of modules.

The obtained result is a framework which guarantees some important and useful features:

- The modular approach allows to describe a wide variety of buildings with complex structures (multi–floor, multi–zones, non–rectangular zones, . . .), while keeping the definition simple for the user.

- The problem is highly customizable, in terms of materials and wall typologies, without affecting the code, *i.e.* the simple data structures are available to directly plug in user–defined elements without need to modify the building model procedure.

The informations are collected into data structures that form a database, entailing the possibility to add user–defined entries for materials or wall types and thus allowing a very high level of detailing for different kind of problems.

‘Building’ data structure It is the main interface for the user to define buildings structures. It contains all the building informations: its dimensions, the layout, the disposition and the types of walls, the subdivision and disposition of zones and the windows surface. This data are stored in the following fields:

- The “Walls–Zones Map”, or **WZmap**, is a 3D-matrix representation of the building frame. Figure 4.2 shows the WZmap matrix for the example building layout of figure 4.1, together with a detailed description of its operating principle.

- The windows surface matrix $\sigma_{\%}$, expressed as a percentage over the total surface of the wall. To allow a very detailed description of the building’s windows surface, the matrix is defined per zone and per type of wall, so that element $\sigma_{\%}(i, j)$ is the surface factor for the j-th wall type of the i-th zone (*e.g.* the windows parameter for the north wall of the first zone).

- The lengths L_x, L_y and L_z of each module along the 3D axis, with z the vertical direction.

$$\begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 3 & -1 & 6 & -4 & 6 & -5 & 3 \\ 0 & 6 & 0 & 0 & 0 & 3 & 0 \\ 3 & -2 & 6 & -4 & 0 & -4 & 3 \\ 0 & 6 & 0 & 6 & 0 & 6 & 0 \\ 3 & -3 & 0 & -3 & 0 & -3 & 3 \\ 0 & 2 & 0 & 2 & 0 & 2 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 4 & 0 & 4 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 4 & 0 & 4 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 4 & 0 & 4 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

(a) First level ($z = 1$) **(b)** Second level ($z = 2$)

Fig. 4.2: Example building *'WZmap'*

The WZmap is a 3*D matrix representation of the building. Each non–zero matrix element represent either a wall or a zone:*

- *a 'wall' entry (in blue in figure 4.1) is a positive number identifying the typology of the wall delimiting the adjacent zones, according to the 'WallType' database numeration.*
- *a 'zone' entry (in orange in figure 4.1) is a negative number,* i.e. *the opposite of the zone ID number, that relates the "module" with the corresponding zone; the elements adjacent to a zone entry in every direction, including z-axis, are either the wall entries corresponding to the walls enclosing the zone, or zeros, in case two adjacent modules are not divided by any wall.*

The rest of the matrix is padded with zeros. The WZmap always have even number of z-levels; odd levels are for zones and vertical walls, even levels are reserved for floors and roof.
Consider as example the building with layout of figure 4.1, where 'Wall A' is the wall facing north, 'Wall B' is facing west, 'Wall C' is facing east and 'Wall D' is facing south. The zones are divided by internal walls. Figure 4.2 represent the building resulting WZmap: there are 9 *zone entries, corresponding to the* 9 *modules in figure 4.1, identified by the corresponding negative zone ID. All the adjacent values are wall entries representative of the wall delimiting the module, assigned according to table 4.1.*

'Material' data structure It is the database for materials used in wall definition. The structure contains the list, sequentially enumerated with a 'material ID number', which collects the material specifications: the name of the material, its density (expressed in $[Kg/m^3]$), its specific heat (in $[J/Kg]$), the thermal conductivity (in $[K \cdot W/m]$) and the width (in $[m]$) of the finite layer for the finite element modelling. The pre–defined materials, reported in Appendix A, are *concrete*, *walling*, *insulation* and *substrate*, while other user–defined materials can be easily implemented.

'Wall Type' data structure Analogously to the 'Material' structure, the 'Wall Type' structure serve as database for the different typologies of walls, in terms of their composition and their orientation, since the latter affects the exposition to external disturbances and therefore some of the parameters of the wall itself. Each element of the list is identified by a sequential ID number, the same used in the WZmap for the layout design. A wall type is defined by a vector m with the list of materials composing the wall, modelled from inner to outer layer and from bottom to top for floors and roof, and a vector w containing the number of finite layers for each material in m. Other data collected in the structure are: the internal and external convective coefficients h_i and h_e; the absorbance coefficient α_S for the shortwave radiation; the absorbance coefficients α_L^u and α_L^d for upwelling and downwelling longwave radiation. The absorbance coefficient are computed as the product of the general coefficients ϵ_S and ϵ_L and the specific view factors. The pre–defined wall types are reported in Table 4.1. Full description of wall types implemented can be found in Appendix A.

ID #	Wall Description
1	External wall – facing north
2	External wall – facing south
3	External wall – facing east\west
4	Floor
5	Roof
6	Internal wall

Table 4.1: Standard wall types

4.2.2 Model order reduction

In Section 2.3.1 we remarked the importance of model order reduction techniques in model implementation to maintain a feasible computation load when solving the optimization problem. The method proposed is a balanced order reduction based on the analysis of the system Hankel's eigenvalues, *i.e.* minimizing the Hankel norm:

$$\|G_n - \widehat{G}_r\|_H \ , \tag{4.1}$$

where G_n is the system original transfer function of order n and $\widehat{G}_r$ is the resulting reduced order transfer function, of desired order r.

The states balanced truncation is performed with the *'balred.m'* Matlab function, with the option to maintain the low-frequency (DC) gains: the resulting states are altered to preserve the system energy, although their matching with physical quantities is lost.

The advantages of this procedure are that system properties like stability, time response, observability and controllability, are preserved, as well as gains of the input\output transformation. On the contrary, the new, reduced order state are physically meaningless, aggravating the problem of state estimation.

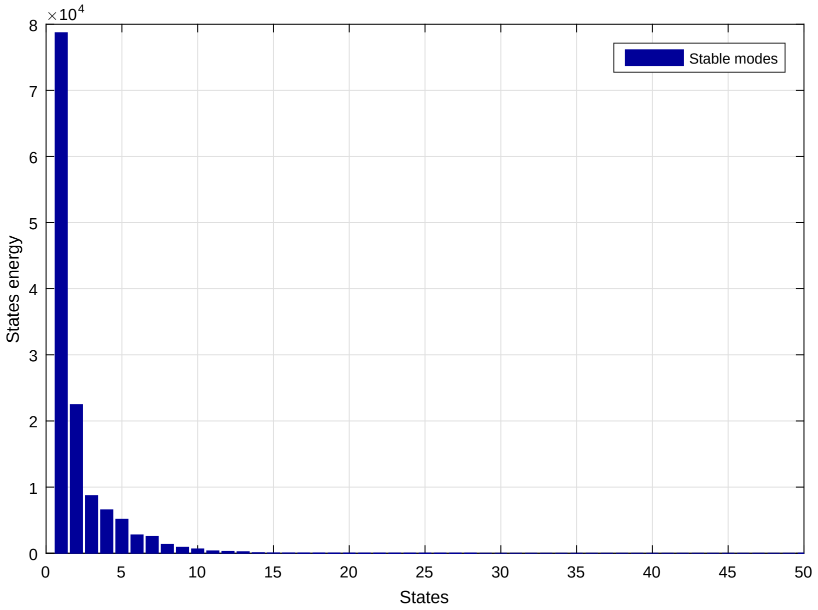


Fig. 4.3: HSVD example (only first 50 states showed)

The choice of an appropriate reduced order r is made by inspecting the system Hankel eigenvalues. The Hankel Singular Values (HSVs) are a measurement of "energy" contribution of each state to the system dynamics, so larger eigenvalues give larger contribution to system dynamics. Properties of stability, time response and frequency are likewise influenced. This suggest that small HSVs do not influence significantly the system properties, and thus can be neglected.

Matlab function *'hsvd.m'* performs the Hankel Singular Values Decomposition. An analysis of the decomposition results gives the picture of the number of significant states.

For example, figure 4.3 shows the HSVs contribution for the the test building of Section 5.2, highlighting the importance of a correct order reduction. The full system's order is in the range of approximately 250 states (only the first 50 are showed in Figure 4.3); however, a quick inspection of the HSVD clearly shows that only the first 10–20 states give noticeable contribution. Hence, the system order can be safely reduced to less than 1/10 of the original order, with substantial consequence on the computational load.

4.2.3 Optimization solver

The optimization problem (3.8) is implemented using the *CVX* toolbox for Matlab, [25], [26]. CVX provide a modelling framework for convex optimization problems, by implementing an intuitive set of rules for problem definition which make use of Matlab syntax, as well as implementing a variety of different solvers for the solution of the problem.

The solver used in the testing of Chapter 5 is *Mosek*, [27]. Mosek solver implement the interior–method point to solve convex optimization problem, and it is optimized towards *sparse large–scale* problems, which best suit our problem set–up.

4.3 Distributed algorithm implementation

In Section 3.5.3 we highlighted the influence of the algorithm choice on the solution of problem (3.26). In Section 3.6, we introduced the distributed optimization Algorithm 1, describing its theoretical requirements and how to fit the specific energy optimization problem into the general algorithm framework.

In this Section, we give a further insight into the specific problems that practical implementation poses, and how they have been tackled.

Initialization

The first step of Algorithm 1 is the initialization, so as to give, for each agent $i = 1, \dots, m$ an initial guess for the optimal solution $x_i(0)$. This guess needs to be a part of the local feasible set, but not necessarily in the global feasible set $\bigcap_{i=1}^{m} X_i$.

For problem (3.26), this means that for all buildings, the initial solution computed must satisfy the following condition:

1. $\bar{\boldsymbol{s}}_{\boldsymbol{i}}(0) \in \mathcal{S}$,

2. $\boldsymbol{u}_{\boldsymbol{i}}(0) \in U_i(\bar{\boldsymbol{s}}_{\boldsymbol{i}}(0))$,

where $\bar{\boldsymbol{s}}_{\boldsymbol{i}}$ is the solution for global variables computed by the i-th agent.

For every agent, we initialize $x_i(0)$ solving the distributed problem as if the buildings were isolated, equivalent to consider a network with no communication arcs for iteration $k = 0$.

Notice that this choice makes each local problem identical to the single–building problem.

Corrected cost function

Consider the algorithm cost function for each local problem as a modified version of the original cost function, in the form:

$$J_i(k) = f_i(\bar{\boldsymbol{s}}_{\boldsymbol{i}}(k)) + \frac{1}{2c(k)} \|\boldsymbol{z}_{\boldsymbol{i}}(k) - \bar{\boldsymbol{s}}_{\boldsymbol{i}}(k)\|^2 \tag{4.2}$$

At iteration k, the cost function consist of 2 terms: *i*) the local cost function $f_i(x_i)$ over the local variables, as in (3.26), and *ii*) a corrective term, penalizing the “distance” between the optimal local solution and the (weighted) neighbours estimates. The latter is the term that drives the different local solutions towards the consensus global solution.

The penalty term is weighted through a coefficient $\frac{1}{2c(k)}$. The coefficient $c(k)$ is a fundamental design parameter that influences the convergence properties and convergence rate of the algorithm. It must satisfy the conditions of Assumption 4. As suggested in [24], the generalized harmonic series family satisfies the hypothesis above, implemented in the form:

$$c(k) = \frac{w}{(k+1)^{\alpha}} \ , \tag{4.3}$$

with $1/2 < \alpha \leq 1$ to satisfy the conditions of Assumption 4. The choice of alpha strongly influence the convergence rate of the algorithm.

The coefficient w is used to make the two terms of the local cost function (4.2) of the same order of magnitude and thus comparable.

Weighted average and network graph

For each agent i, the weighted average $\boldsymbol{z}_{\boldsymbol{i}}(k)$ is the term that accounts for information from other agents at the k-th iteration, combining their estimated optimal solution as:

$$\boldsymbol{z}_{\boldsymbol{i}}(k) = \sum_{j=1}^{m} a_j^i(k) \bar{\boldsymbol{s}}(k) \ . \tag{4.4}$$

The information from neighbouring agents are weighted through the coefficients a^i_j. These coefficients play a twofold role: 1. defining the network graph and 2. granting the correct mixing, at a non–decreasing rate, of the informations. The latter point guaranteed by Assumption 5, so as its conditions ensure that the resulting combination of the various estimates $\bar{\boldsymbol{s}}_j(k)$ is still convex.

As for the first point, we recall that in the network graph (V, E_k), the set of edges at k-th iteration is defined as:

$$E_k = \left\{(j, i) \colon a^i_j(k) > 0\right\}$$

The weight coefficients can be collected in a m-by-m weight matrix, that defines the topology of the network, in the form:

$$A(k) = \begin{bmatrix} a^1_1(k) & a^1_2(k) & \cdots & a^1_m(k) \\ a^2_1(k) & a^2_2(k) & \cdots & a^2_m(k) \\ \vdots & \vdots & \vdots & \vdots \\ a^m_1(k) & a^m_2(k) & \cdots & a^m_m(k) \end{bmatrix} \tag{4.5}$$

Hence, matrix (4.5) is the representation of the network directed graph and provides an immediate picture of the network situation at time instant k. The time–dependence of the network topology is expressed through the coefficient dependence on k.

Convergence condition

In the general formulation of Algorithm 1, the *convergence condition* is not specified, leaved as the generic condition "**For** $i = 1, \dots, m$ **repeat until convergence**".

We consider that the problem has reached convergence at iteration k when, for every agent $i = 1, \dots, m$, the "distance", express in term of infinite norm, between the current value $x_i(k)$ and the previous value $x_i(k-1)$ of the solution is below a suitable threshold δ:

$$\left(\left\| x_i(k) - x_i(k-1) \right\|_\infty \vee \left\| \frac{x_i(k) - x_i(k-1)}{x_i(k-1)} \right\|_\infty \right) \leq \delta \ . \tag{4.6}$$

The condition consider both the *absolute* and the *relative* variation of the variables at two consecutive iteration, considering convergence if either one of the two variation is smaller than the chosen threshold. The condition is checked element–wise for every variable of the vector x_i, and it must hold true for every agent at the same iteration.

Chapter 5

Testing and results

In this chapter we provide the results of the tests to evaluate the performances of the distributed algorithm, analysing the solution obtained for various situations accounting for different building structures and network communication graphs.

5.1 Introduction

The final part of the work is devoted to the presentation of results achieved applying the distributed algorithm to some case studies. We considered a set of tests to analyse 9 situations, in terms of building structures and communication graphs. These case studies provide a general picture of the algorithm performances, advantages and drawbacks, under different conditions.

The chapter is divided in two sections:

- the study of a single building;
- the study of a multi–building district.

The results achieved with the distributed approach will be analysed and discussed.

5.2 Single building results

We now define a certain building structure, that will also be employed in the multi–building case. We adopt the zones set–up and wall characteristics in [10].

The building is a medium–size commercial building. It is a square–based, 3 storeys building with dimensions $20\,m \times 20\,m \times 9\,m$, each floor being $3\,m$

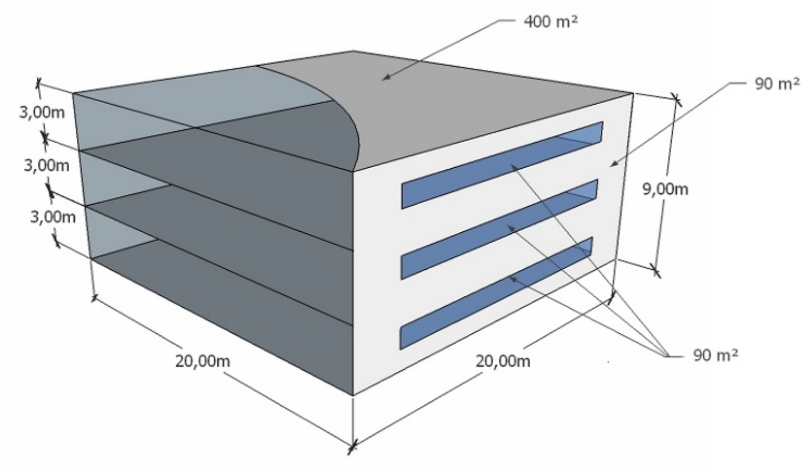


Fig. 5.1: Building layout

tall. Each facade is $180\,m^2$, half being windows and the other half consisting of solid walls. The detailed description of the walls composition is reported in Table A.2. The orientation of each facade influences the corresponding sun exposure and therefore the amount of incident solar radiation. This informations are collected in the specific absorbance coefficient (see Table A.3 for the detailed specifics). Figure 5.1 shows the layout of the building.

Figure 5.2 shows the building subdivision in zones, where each floor is treated as a separate thermally regulated zone.

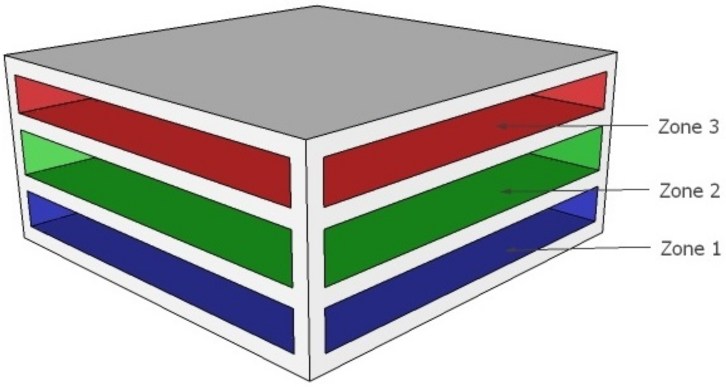


Fig. 5.2: Zones subdivision for building in Fig. 5.1

This set–up recalls the “Multi–zone” case studied in [10] and [13] which, combined with the presence of the storage unit, provides the most general and flexible set–up. Different set–ups (storage versus no–storage, single–

zone versus multi–zone) addressed in [10] and [13], with discussion of their the advantages\drawbacks, are not of interest for the purpose of this work.

Simulation parameters The optimal energy management problem is solved and observed along a one day finite horizon, discretized in time–slot of length dt. We used the following parameters:

- The time–step for horizon discretization has been set to the value of $dt = 10$ minutes.
- The simulation is performed over an horizon of 1 day; with the time–step dt, the number of time–slots in the horizon is $M = 144$.

The storage has minimum and maximum capacity of $S_{min} = 0$ MJ and $S_{max} = 180$ MJ, a maximum energy exchange rate of $s_{max} = 15$ MJ and a energy losses coefficient of $a = 0.995$.

The building is equipped with a 'Medium' size chiller (see Table 5.2), with coefficients $c_2 = 3.79 \cdot 10^{-5}$, $c_1 = 2.77 \cdot 10^{-2}$ and $c_0 = 2.46$. Its maximum electrical consumption is $E_{l,max} = 30$ MJ.

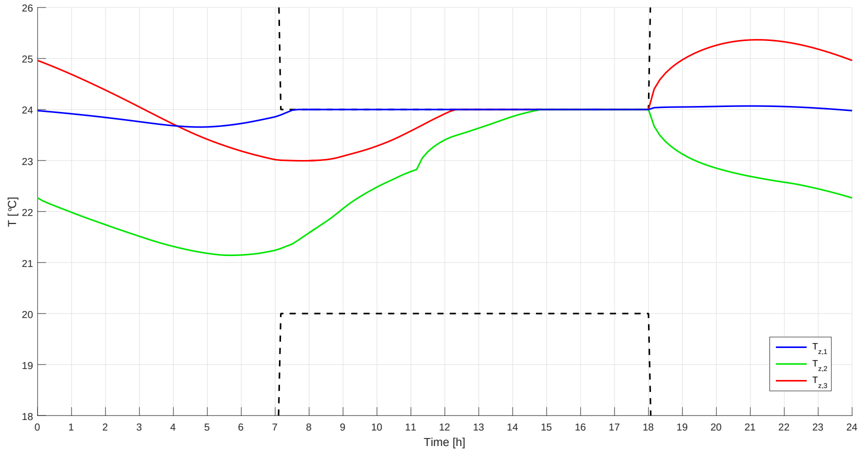
Other data relative to building structure, disturbances profiles and energy cost profile are reported in Appendix A.

We next apply the distributed Algorithm 1 for optimal energy management of the building via optimization of the zones temperature profiles and storage usage. In this case (one single building) the distributed algorithm is identical to the centralized solution.

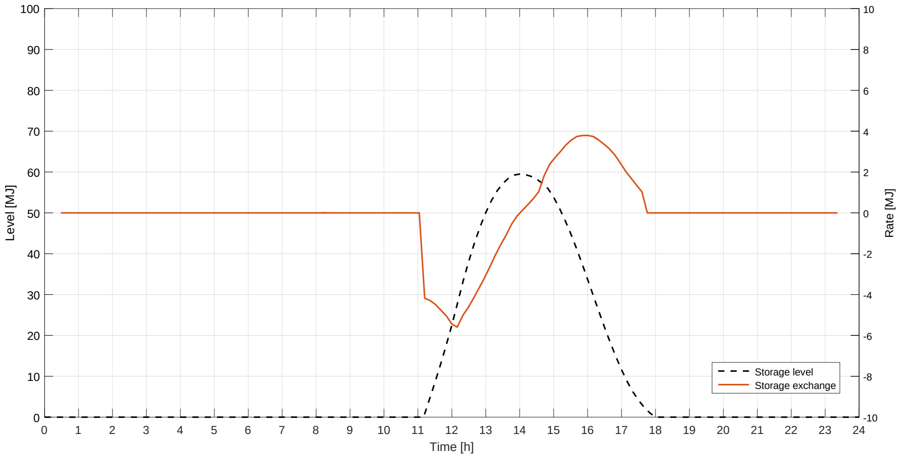
Results analysis

Figure 5.3 shows the solution. The optimal temperature set–points in Figure 5.3.a shows, as main trend, a strong pre–cooling of the building, before the "working hours" (8 AM – 5 PM period). Inspecting zone by zone:

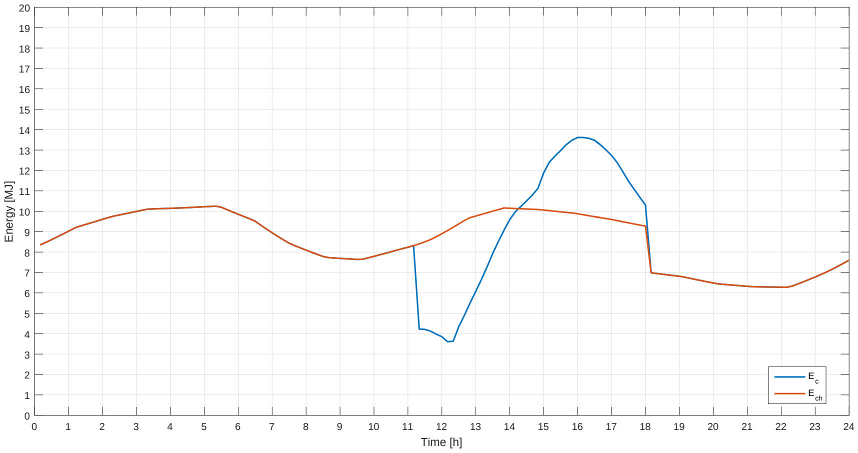
- The ground floor, coinciding with zone 1 (blue line), is weakly affected by external disturbances, and its temperature can be easily kept almost constant.
- The middle floor, *i.e.* zone 2 (green line), is used as an additional thermal storage. Its temperature is strongly lowered in the $0 - 6$ and $18 - 24$ periods, and is let free to slowly rise during the occupancy period. Due to its position, its dynamics are strongly interlaced with both the first and third zone, absorbing heat through both the floors.



(a) Optimal temperature profiles



(b) Optimal storage profile



(c) Cooling, Chiller and Electrical energy requests

Fig. 5.3: Initialization optimal solution

Therefore, if it subject to a pre–cooling during non–occupancy hours, it can store "cooling" energy that is released during the day to other zones.

- The top floor, *i.e.* zone 3 (red line), opposite to zone 1, is strongly subject to external disturbances due to the extended roof surface absorbing solar radiation; optimal control impose the pre–cooling during early hours to reduce the energy request peak in the middle of the day, using the structure thermal inertia to shift the energy load.

The storage optimal profile in Figure 5.3.b shows a (negative) peak in the storage exchange, while the storage is being charged, followed by a peak of usage of previously stored energy.

Consider Figure 5.3.c, where E_c is the cooling energy request, E_{ch} is the chiller energy request and E_l is the electrical energy consumption. By inspecting the cooling energy request, it can be noticed a decrease in the request, followed by its quick rise to a peak, coinciding with the peaks in the storage usage. On the contrary, the energy request to the chiller remains approximately constant in the same time period: the storage unit allows to take advantage of the demand drop to produce energy at an optimal efficiency, using it to reduce the energy needed during high–load periods. Figure 5.4 shows this results on the chiller coefficient of performance, which is kept approximately constant as well.

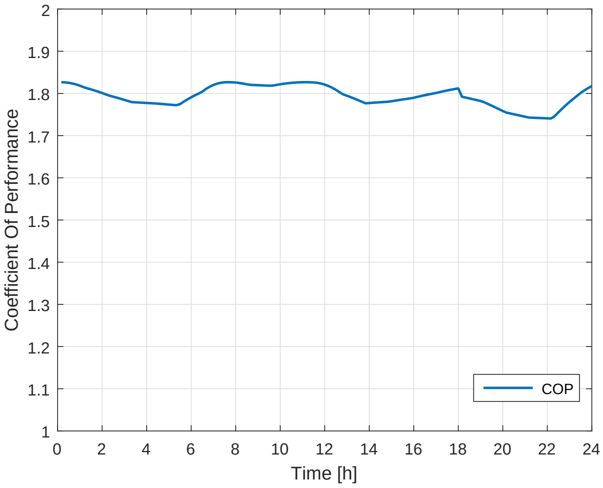


Fig. 5.4: Coefficient of performance (COP)

5.3 Testing for a building district: general overview

In the case studies, we analyse a network of 3 agents communicating with each other through a fixed or time–varying network. Each agent computes a tentative solution and exchanges information with the neighbouring agents, then repeat iteratively the procedure until consensus among agents is reached. The testing will be carried out on 3 different districts, with the goal of analysing the algorithm performance in different settings:

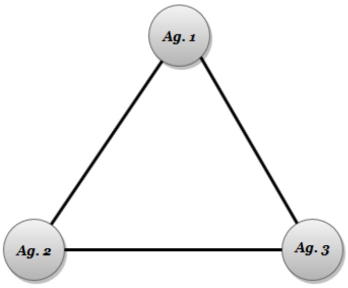
1. In the first district, *District* 1, we consider identical buildings, equipped with the same chiller, meaning that each agent has the same constraint set X_i and the same objective function $f_i(x_i)$. We expect a high degree of symmetry in the local solutions.
2. In the second district, *District* 2, we consider buildings all having the same structure, but equipped with different the chillers, implying that the constraint sets are still the same but the local cost functions are different.
3. In the third district, *District* 3, the buildings in the network are different both in the structure and for the chiller characteristics, so that local optimization problems are different in both the objective function $f_i(x_i)$ and constraint set X_i.

For the initialization iteration we solve the problems as in Section 5.2.

For each group of tests, we implement 3 types of communication networks to analyse their influence on the algorithm performances, such as the number of iteration required, the average time per iteration and the overall time needed to reach convergence.

The devised communication structures are the following:

1. The first network, shown in Figure 5.5.a, is a complete graph, meaning that each agent is connected to every other through a direct arc, and the information are passed directly from every agent to every other one. The corresponding weight matrix is shown in Figure 5.5.b. Every element of the matrix A is non–zero, *i.e.* $a_{i,j} \neq 0$, $\forall i, j = 1 \dots m$, due to the fact that the graph is complete. The weight of each arc instead is an arbitrary design parameter, subject to Assumption 5. We opted to equally weight each arc. The network is fixed in time.
2. The second network is represented as a connected graph, *i.e.* a path exists between each pair of nodes in the graph, but not a complete



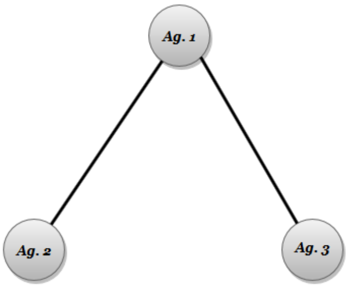
(a)

$$A_a = \begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{bmatrix}$$

(b)

Fig. 5.5: Complete network design and weight matrix

graph, so that not every pair of agents is connected by a direct arc. Thus, every agent receives informations from every other agent either directly, if there is an arc connecting them, or indirectly through intermediate nodes along the path. The weights of the arcs connecting different nodes are all set identical and equal to 1/3. The other coefficient are set accordingly to satisfy Assumption 5. Figures 5.6.a and 5.6.b shows the network and its corresponding weight matrix A_b . As for the previous case, also the second network is time–invariant.



(a)

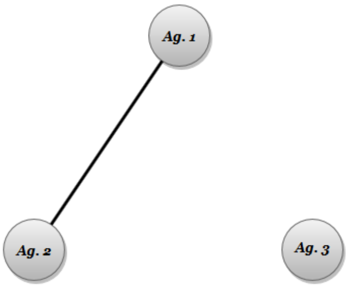
$$A_b = \begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 1/3 & 2/3 & 0 \\ 1/3 & 0 & 2/3 \end{bmatrix}$$

(b)

Fig. 5.6: Connected network design and weight matrix

3. The third network is a time–varying network, cycling through 3 different graphs; these three graphs are not complete nor connected, but at some point in the cycle each agent becomes connected to at least part of the network, receiving the informations needed to update its local solution and send it to other agents. Figure 5.7 shows the first 3 instances of the network, that are repeated cyclically until convergence is reached, with the corresponding weight matrices.

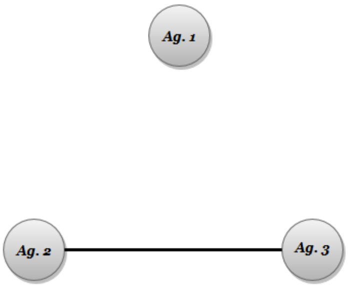
Notice that, even if each k-th instance of the network is not connected, the resulting graph (V, E_∞), with E_∞ defined as in Section 3.6, is strongly connected and satisfies the algorithm Assumption 6 with $T = 3$.



(a) Network for $k = 1$

$$A_{c,1} = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 1/2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

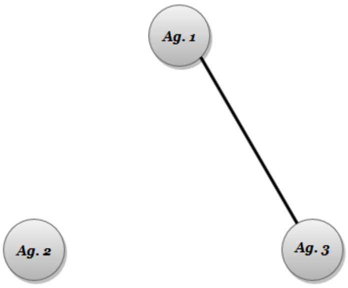
(b) Weight matrix for $k = 1$



(c) Network for $k = 2$

$$A_{c,2} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1/2 & 1/2 \\ 0 & 1/2 & 1/2 \end{bmatrix}$$

(d) Weight matrix for $k = 2$



(e) Network for $k = 3$

$$A_{c,3} = \begin{bmatrix} 1/2 & 0 & 1/2 \\ 0 & 1 & 0 \\ 1/2 & 0 & 1/2 \end{bmatrix}$$

(f) Weight matrix for $k = 3$

Fig. 5.7: Time–varying network design and weight matrix

Finally, recall the convergence condition defined in Section 4.3. The quantity used to check convergence can either be the optimal value of the cost function $J_i(k)$, or the optimal solution $x_i(k)$. In the problem under consideration, given the magnitude of the quantities involved, using $x_i(k)$ provides a more accurate condition on convergence.

The threshold value is chosen considering a trade–off between the solution precision and the algorithm speed of convergence. The number of iterations needed to reach smaller values of tolerance increase more than linearly when approaching the CVX solver precision tolerance. After testing different values of the threshold, the empirical choice is setting the threshold value at $\delta = 10^{-3}$.

In all the following simulations we use a thermal storage unit with minimum capacity of $S_{min} = 0$ MJ and maximum capacity of $S_{max} = 540$, three times the capacity of the storage used in Section 5.2 for the single building study. The maximum charge\discharge rate is $s_{max} = 15$ MJ, and the losses coefficient is $a = 0.995$.

5.3.1 District 1

The first district consists of $m = 3$ buildings identical in structure and parameters, modelled as the single case building of Section 5.2. Each building is equipped with the same chiller, selected from the list in Table 5.2. We opted for a 'Medium' size chiller, with coefficients $c_2 = 3.79 \cdot 10^{-5}$, $c_1 = 2.77 \cdot 10^{-2}$ and $c_0 = 2.46$ and maximum electrical consumption $E_{l,max} = 30$ MJ.

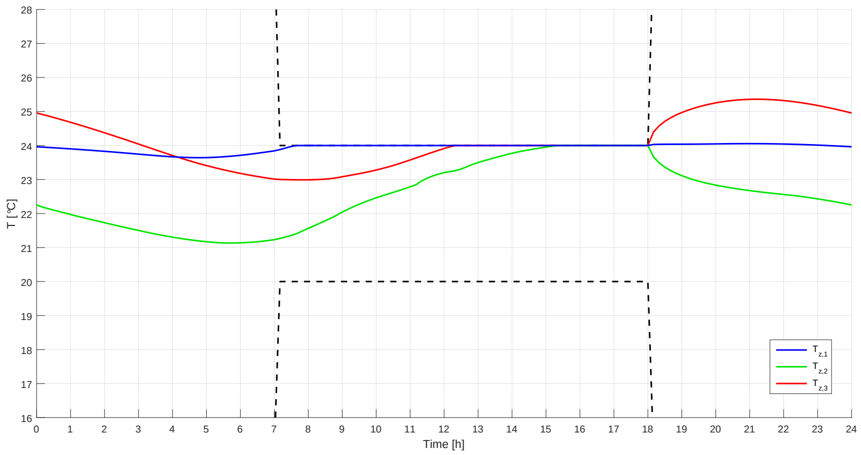
Identical structures and identical chillers implies that the local constraints sets U_i and the objective functions f_i, $i = 1, 2, 3$, are also identical and therefore each agent solve the same local problem. For this reason, we expect a high degree of symmetry in the solutions achieved.

Simulation results

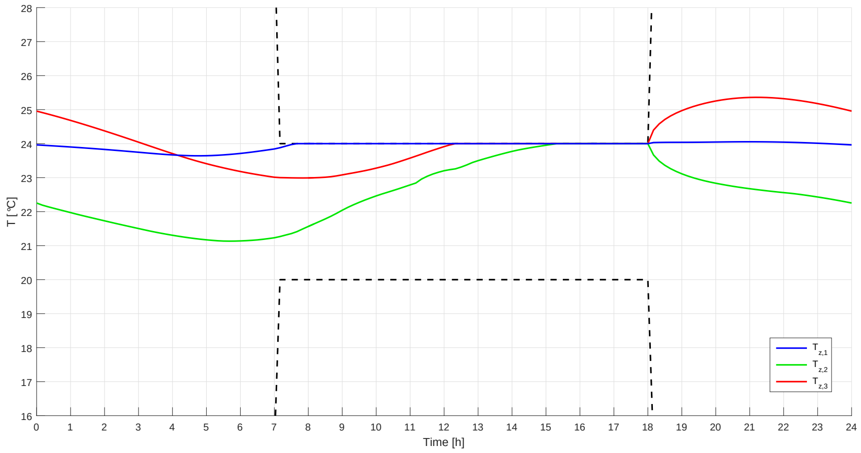
Figure 5.8 shows the optimal temperature profile for each agent.

The results shows a perfect match of the temperature set–points computed by each agent. This is expected, as each agent solve the same local problem due to the symmetry of the setting.

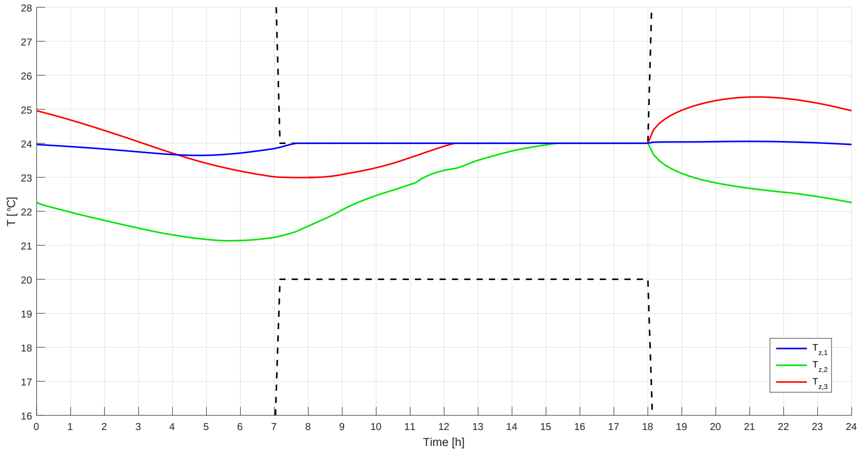
The optimal profiles show the same general trends of the single problem. The building is pre–cooled outside of the working hours to shift in time the energy request peak, and the middle zone, with a stronger pre–cooling,



(a) Building 1

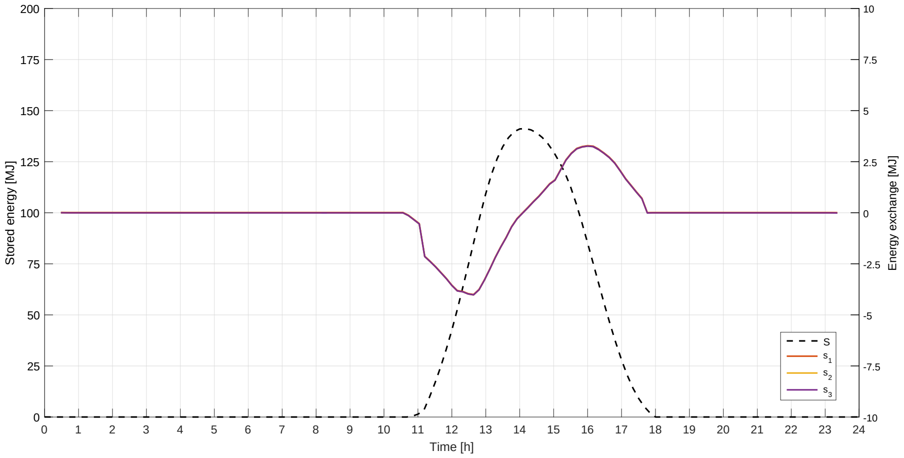


(b) Building 2

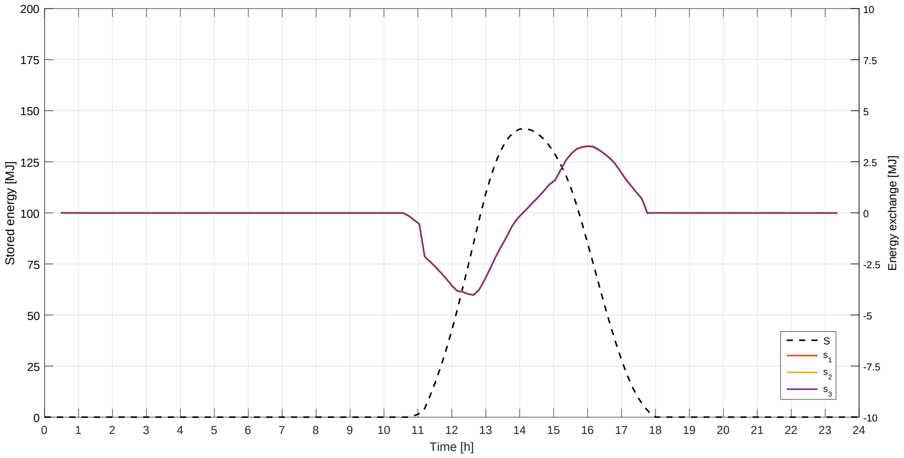


(c) Building 3

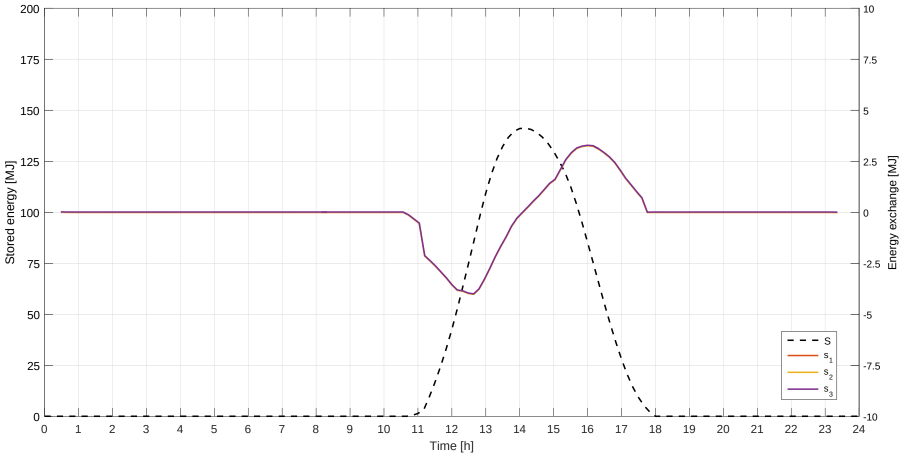
Fig. 5.8: District 1 – Optimal temperature profiles



(a) Building 1



(b) Building 2



(c) Building 3

Fig. 5.9: District 1 – Optimal storage profiles

is used as an additional thermal storage that provides additional cooling energy to the other zones.

Figure 5.9 shows the optimal profiles of the energy exchanges with the storage for the three buildings, after convergence of the distributed algorithm.

The optimal solutions $\bar{\boldsymbol{s}}^{\boldsymbol{i}}$, $i = \{1, 2, 3\}$, coincide: the agents reached consensus on an identical usage of the shared resource.

The symmetry of the solution is coherent with the symmetry of the setting and of the local problems.

Consider the results for the single building case of Section 5.2. Since the storage usage is well below the storage limits, we expect the overall usage of the storage, i.e. $\boldsymbol{s_1} + \boldsymbol{s_2} + \boldsymbol{s_3}$, to be 3 times that of the single building case.

The solution does not meet exactly this expectation, as the storage usage is slightly below the expected value. However, by inspecting the value of the local objective function for each building, $J_1 = J_2 = J_3 = 15.429$ and comparing it with the single building cost, $J = 15.436$, we found an absolute and relative error on the figure of merit, respectively:

$$\epsilon_{abs} = 0.007$$

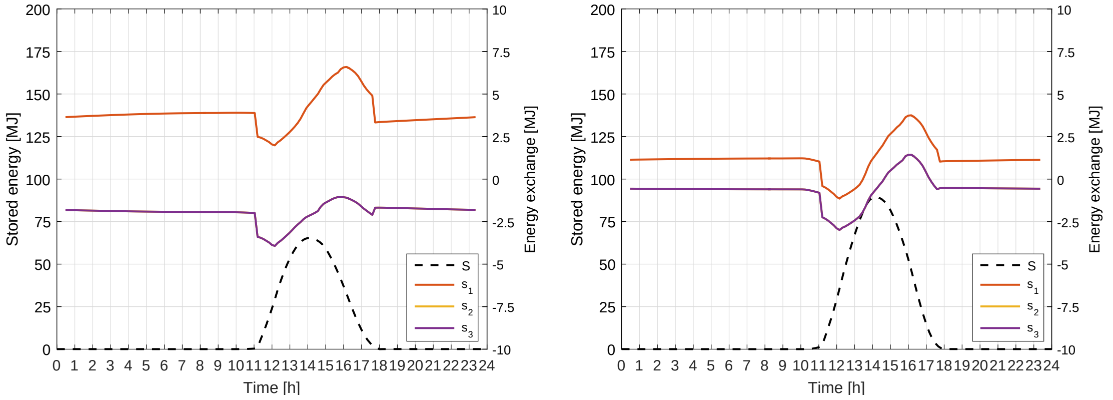
$$\epsilon_{rel} = 0.045\%$$

which is ascribable to numerical precision.

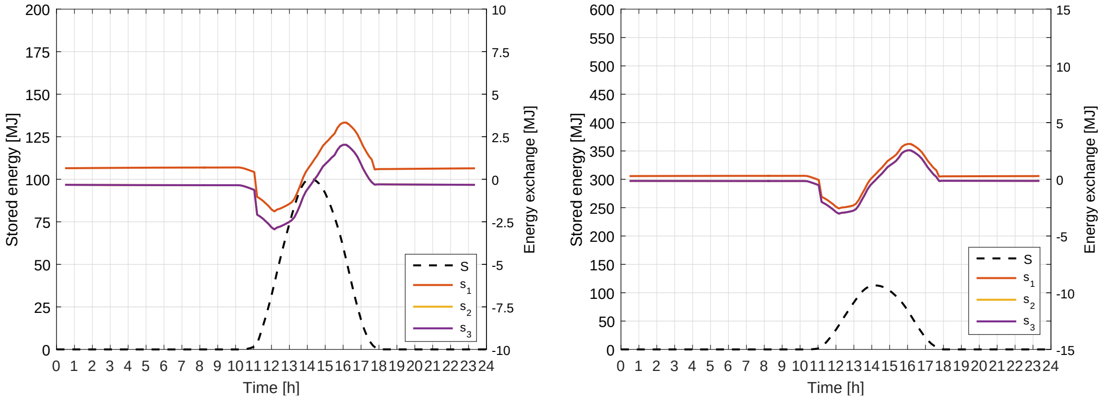
The inspection of the tentative solution computed as a function of the iteration number k gives an interesting insight on the algorithm mechanism. It will be analysed for agent 1 in the complete communication graph case. Analogous considerations can be made for the solutions computed by agents 2 and 3 and for the other network structures.

Consider the evolution of the solution as computed by agent 1, shown in Figure 5.10, where $\boldsymbol{s_1}$ (orange line) is the storage exchange profile of building 1, $\boldsymbol{s_2}$ and $\boldsymbol{s_3}$ are the storage exchanges for building 2 (yellow line) and building 3 (purple line) respectively, and $\boldsymbol{S}$ (black dashed line) is the stored energy level. The profiles for $\boldsymbol{s_2}$ and $\boldsymbol{s_3}$ are perfectly overlapped.

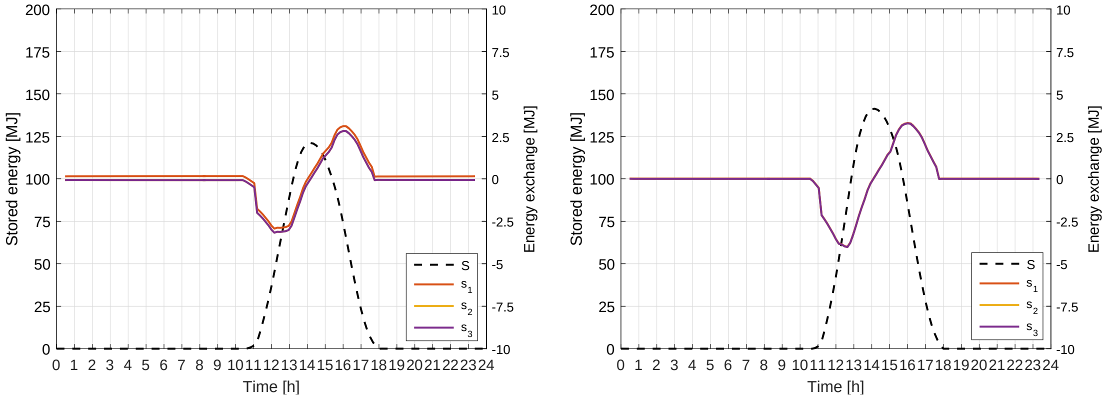
At the first iteration, due to the low weight of the penalization term, the agent is completely selfish, minimizing its own local cost and disregarding the needs and costs of other buildings. Indeed, the coupling provided by the storage implies the coupling of the buildings chiller plants, and agent 1 takes advantage of this “shared” cooling system to optimize its own local cost function: building 1 is purely a consumer, while buildings 2 and 3 provides the energy and recharge the storage (Figure 5.10.a).



(a) $k = 1$ **(b)** $k = 10$



(c) $k = 20$ **(d)** $k = 50$



(e) $k = 100$ **(f)** Consensus solution

Fig. 5.10: Evolution of storage usage by the 3 agents and stored energy, computed by Agent 1

As the number of iterations increases, more informations are exchanged, and these are given a higher importance ($\frac{1}{2c(k)}$ grows linearly with k). Hence, the agent progressively reduces its selfish behaviour (Figure 5.10.b for $k = 10$, Figure 5.10.c for $k = 20$, Figure 5.10.d for $k = 50$) by allowing the other agents to use the storage more.

The process continues until the storage usage is comparable for each building (Figure 5.10.e for $k = 100$). Finally, the solution is refined to reach consensus up to the convergence threshold δ (Figure 5.10.f).

Even if the buildings are not explicitly aware of the other buildings constraints and needs, eventually each one accounts for them in its solution. The process takes place indirectly through the necessity of agree on a shared resource, without exchanging sensitive information directly.

	Complete Network	**Connected Network**	**Time–varying Network**
# of iterations	194	354	878
Average time			
Building 1	2.743s	2.747s	2.693s
Building 2	2.753s	2.753s	2.687s
Building 3	2.738s	2.742s	2.686s
Overall time			
Building 1	532.160s	972.692s	2407.32s
Building 2	534.164s	968.333s	2399.91s
Building 3	531.219s	970.668s	2398.38s

Table 5.1: Simulation data for District 1

Finally, we provide some statistics on the algorithm performances under different conditions (see Table 5.1).

As expected, the best performance is obtained with a complete communication network, which corresponds to the smallest number of iterations and, consequently, the smallest overall time.

On the contrary, the time-varying graph leads to the slowest convergence speed due to the non–continuous flow of information.

The overall time needed for each agent depends only on the communication structure and on the number of iterations, while the average time remains constant for each building and for each network, as it depends only on the local optimization problem dimension.

5.3.2 District 2

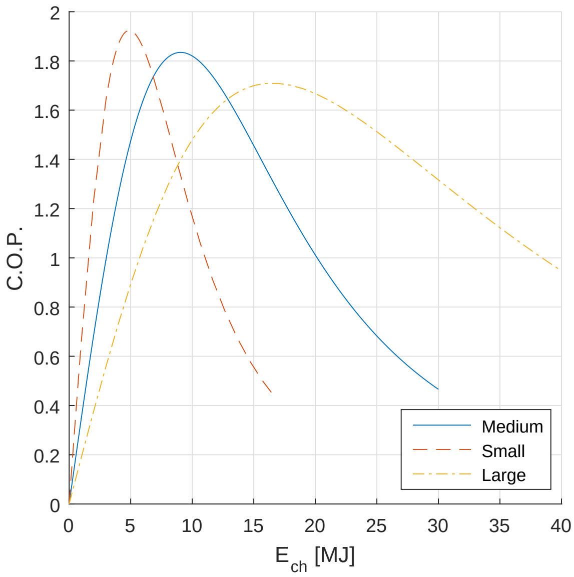


Fig. 5.11: C.O.P. curves for chillers of different size

The second district consists of $m = 3$ identical buildings. As for *District* 1, each building is modelled after the single case building, and the local constraint sets U_i are still identical. Differently from District 1, each building is now equipped with a different chiller: Specifically, the first building is equipped with a 'medium' size chiller, the second building is equipped with a 'small' size chiller and the third building is equipped with a 'large' size chiller. The Coefficient of Performance curves of the three chillers are reported in Figure 5.11, where $COP = E_{ch}/E_l$, and the coefficients for their biquadratic approximation are listed in Table 5.2.

Having different chillers implies that each agent solves a local problem with a different objective function.

Chiller size	c_2	c_1	c_0	$E_{l,max}$ [MJ]
Small	$2.49 \cdot 10^{-4}$	$4.98 \cdot 10^{-2}$	1.26	18
Medium	$3.79 \cdot 10^{-5}$	$2.77 \cdot 10^{-2}$	2.46	30
Large	$3.56 \cdot 10^{-6}$	$1.58 \cdot 10^{-2}$	5.11	40

Table 5.2: Different chillers with parameters

Simulation results

Figure 5.12 shows the optimal temperature profiles of each building. The solutions are very similar and this is expected, given that the 3 buildings have the same structure. Small differences can be observed in the pre–cooling phase of zone 2, which is more pronounced for building 3, equipped with the large chiller, and less for building 2, equipped with the small one. The results are also similar to those of District 1 (Figure 5.8), therefore all the comments about the shape of the profiles still apply.

The optimal storage profiles are shown in Figure 5.13, where $\boldsymbol{s_1}$ (orange line), $\boldsymbol{s_2}$ (yellow line) and $\boldsymbol{s_3}$ (purple line) are the energy exchanges of respectively building 1, building 2 and building 3 with the storage.

In this case, it is possible to identify some specific behaviours; for example building 2 draws energy from the storage for the entire day, whereas building 3 constantly provides the energy to charge the storage. This is a noticeable fact in that it shows how the buildings can cooperate by sharing resources.

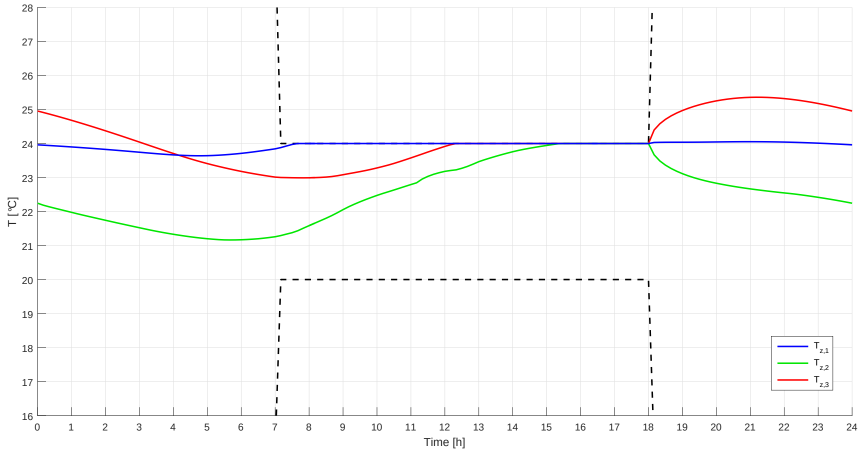
Consider now $k = 0$, where each building is unaware of other agents choices. Given the cooling energy request, both the small and the large chiller operate in suboptimal conditions with reference to the COP curves in Figure 5.11: the small chiller is overloaded, working to the right of its optimal COP, while the large chiller is underloaded, working to the left of its maximum COP. The resulting COP profiles (blue line for the medium chiller, orange dashed line for the small one and yellow dotted line for the large one), in Figure 5.14.a, show that both the chillers are producing energy with a low performance coefficient.

After reaching consensus, we can see from Figure 5.14.b that all chillers work at their best efficiency. This optimal condition can be achieved thanks to the shared thermal storage through which building 3 can supply part of the cooling energy needed by building 2.

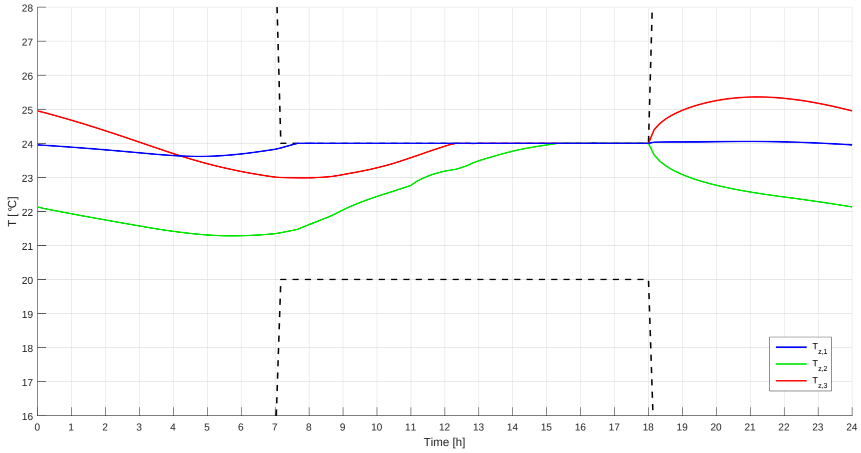
Figure 5.15 shows the evolution of the optimal storage profiles as a function of number of iteration k. $\boldsymbol{s_1}$ (orange line), $\boldsymbol{s_2}$ (yellow line) and $\boldsymbol{s_3}$ (purple line) correspond to the optimal profile of the storage energy exchange with building 1, building 2 and building 3 respectively, while $\boldsymbol{S}$ (black dashed line) is the optimal profile of the amount of stored energy.

At the first iteration, the agent is selfish, as the profile of $\boldsymbol{s_1}$ is set to minimize its own cost, regardless of the other buildings (Figure 5.15.a).

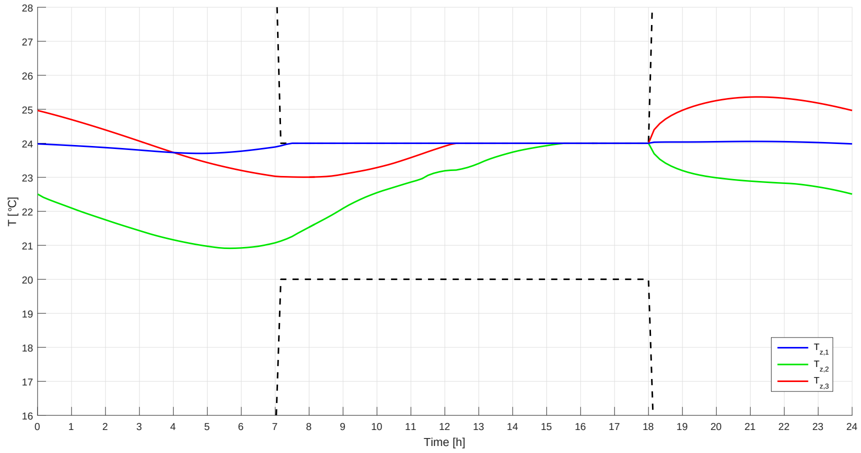
As iterations progress, the gap between $\boldsymbol{s_1}$ and $\boldsymbol{s_2}$ is promptly closed



(a) Building 1

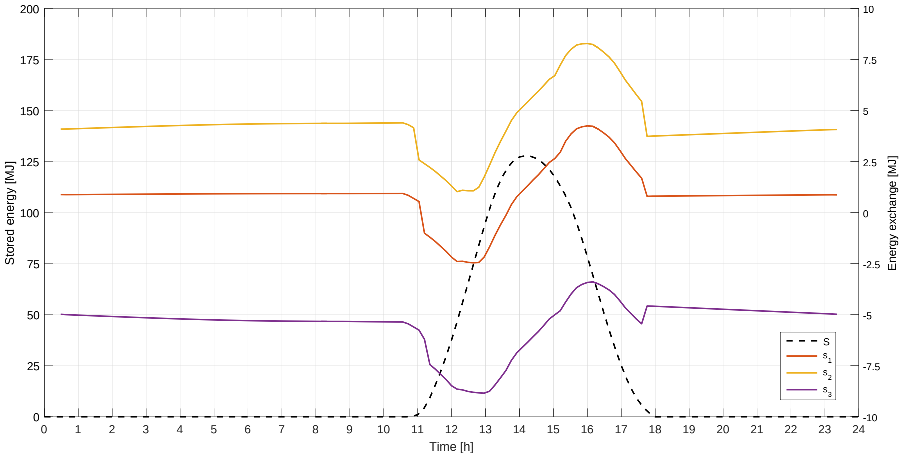


(b) Building 2

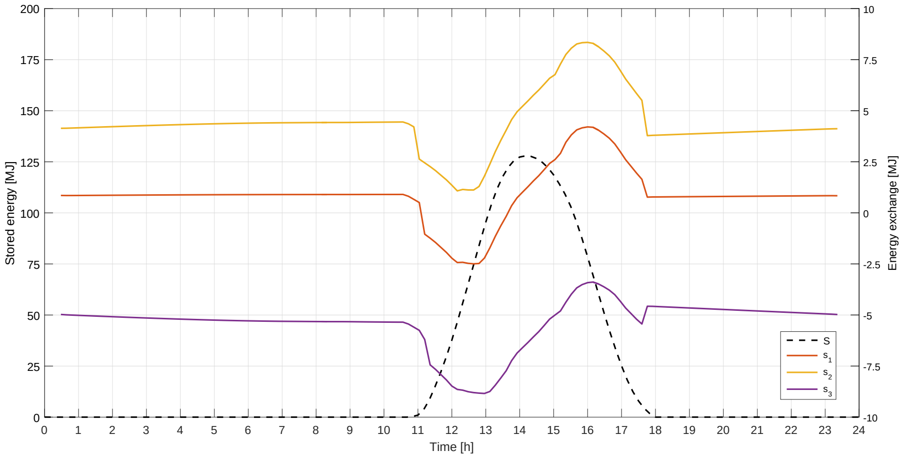


(c) Building 3

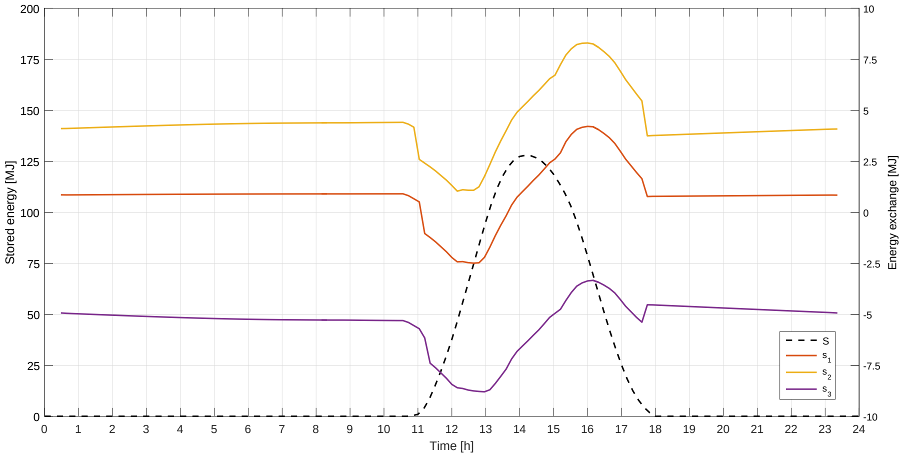
Fig. 5.12: District 2 – Optimal temperature profiles



(a) Building 1

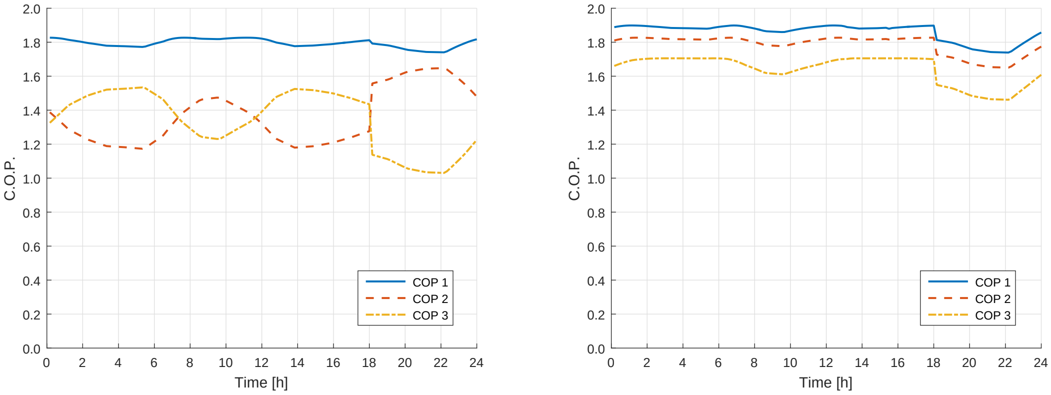


(b) Building 2



(c) Building 3

Fig. 5.13: District 2 – Optimal storage profiles



(a) Initialization COP **(b)** Optimal COP

Fig. 5.14: District 2 – COP comparison

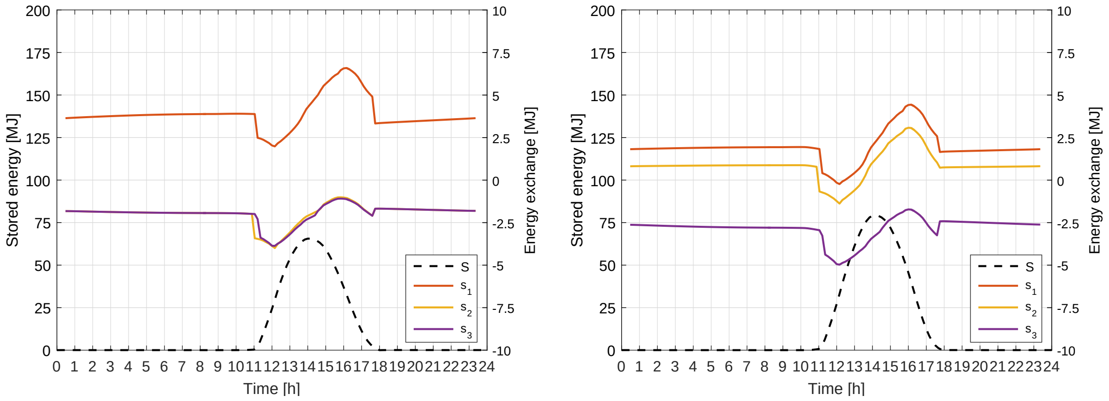
(Figure 5.15.b), and in very few iterations the two profiles are swapped (Figure 5.15.c), meaning that the needs of building 2 are quickly taken into consideration. For the remaining iterations, the solution is refined to reach consensus up to the convergence threshold accuracy (Figure 5.15.f).

Figure 5.15 refers to the solution computed by agent 1 in the complete communication graph. Analogous consideration can be made for other agents and communication structures.

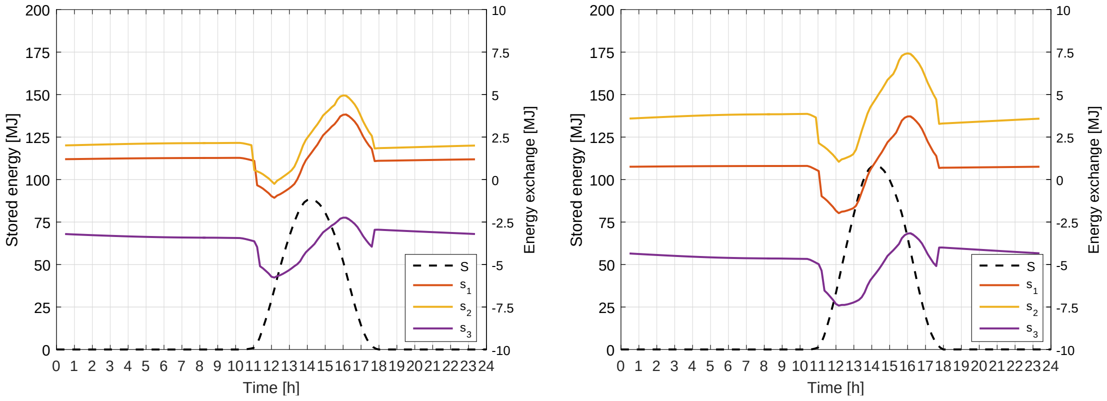
	Complete Network	**Connected Network**	**Time–varying Network**
# of iterations	223	278	1032
Average time			
Building 1	2.635*s*	2.579*s*	2.571*s*
Building 2	3.367*s*	3.267*s*	3.191*s*
Building 3	2.966*s*	2.860*s*	2.901*s*
Overall time			
Building 1	587.667*s*	717.083*s*	2653.27*s*
Building 2	750.889*s*	908.134*s*	3293.11*s*
Building 3	661.382*s*	795.150*s*	2993.83*s*

Table 5.3: Simulation data for District 2

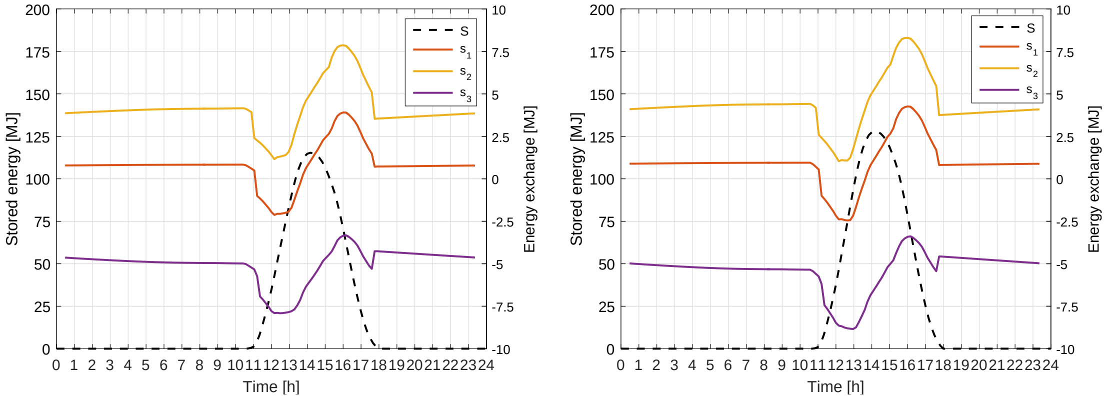
In Table 5.3, we provide some statistics of the algorithm performance in the District 2 case. We can make observations similar to District 1.



(a) $k = 1$ **(b)** $k = 5$



(c) $k = 10$ **(d)** $k = 50$



(e) $k = 100$ **(f)** Consensus solution

Fig. 5.15: Evolution of storage usage by the 3 agents and stored energy, computed by Agent 1

The agents reached consensus independently of the communication graph, which influence only the speed of the algorithm, and therefore the number of iteration and the overall time needed to reach convergence. Variations in the average time for each building is due to the difference in their local problems problem.

5.3.3 District 3

The last district consists of 3 different buildings. In contrast to the previous cases, each building now has a different structure:

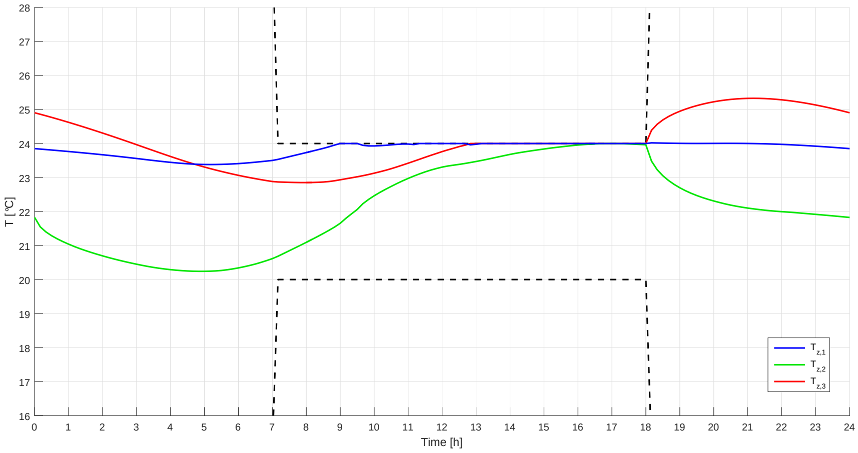
- Building 1 is the same building of the single building case, with the identical structure, zones subdivision and parameters (full details in Section 5.2).
- Building 2 is similar to building 1 in term of structure and parameters, but it has a different zone subdivision. Specifically, *i*) the ground floor is divided into two zones, namely zone 1 and zone 2, by an internal wall, along the North–South direction; *ii*) the first floor is similarly divided into two zones, namely zone 3 and zone 4, along the East–West direction; *iii*) the top storey is a single zone, namely zone 5, identical to the previous cases.
- Building 3 is a $30\,m \times 60\,m \times 5\,m$ rectangular–based, single storey building, representative of a large warehouse or industrial facility. The facades, half made of solid walls and half consisting of windows, have different absorption coefficients due to different orientation, analogously to the other buildings. The building is divided into 3 thermal zones of $30\,m \times 20\,m$ by internal walls.

Similarly to District 2, the buildings have different chillers plants, described by the coefficients in Table 5.2. The buildings are assigned, respectively, with the ‘medium’, ‘small’ and ‘large’ chiller.

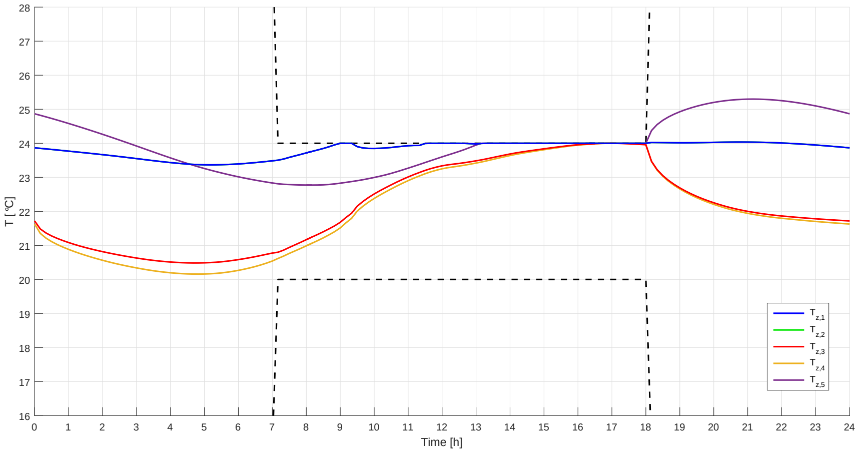
As a consequence of the set–up, the local problem solved by each agent has different objective functions f_i and constraints sets U_i.

Simulation results

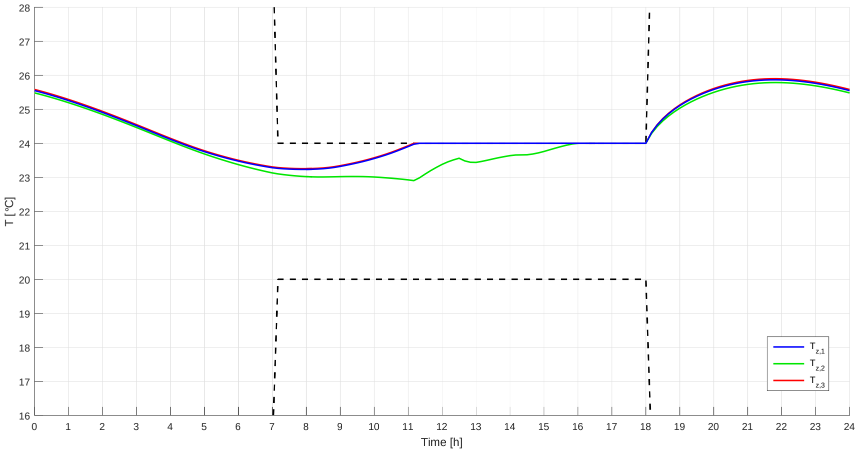
Figure 5.16 shows the optimal temperature set–points for the third district.



(a) Building 1

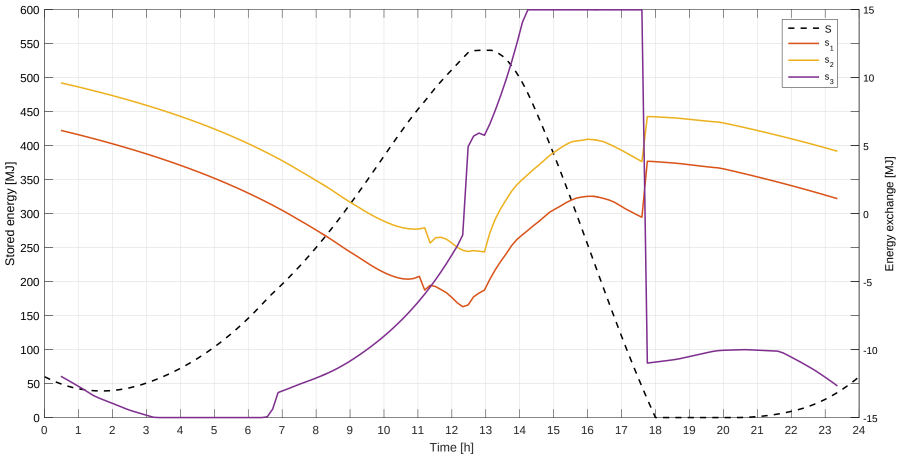


(b) Building 2

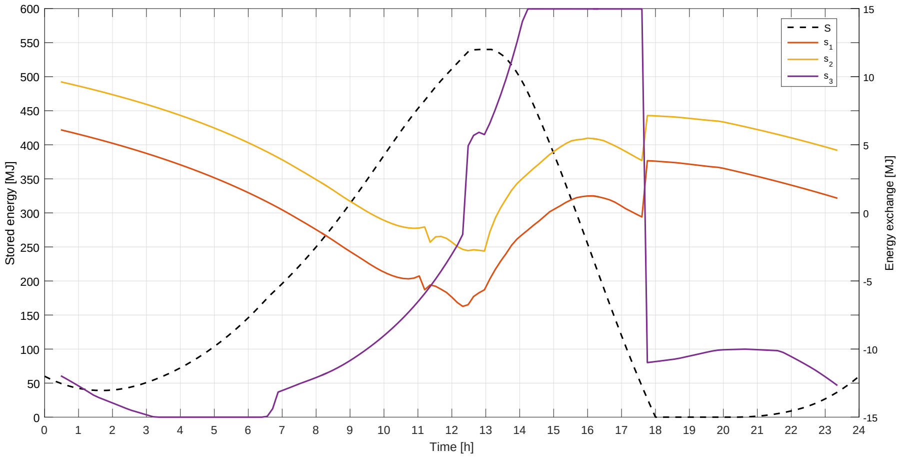


(c) Building 3

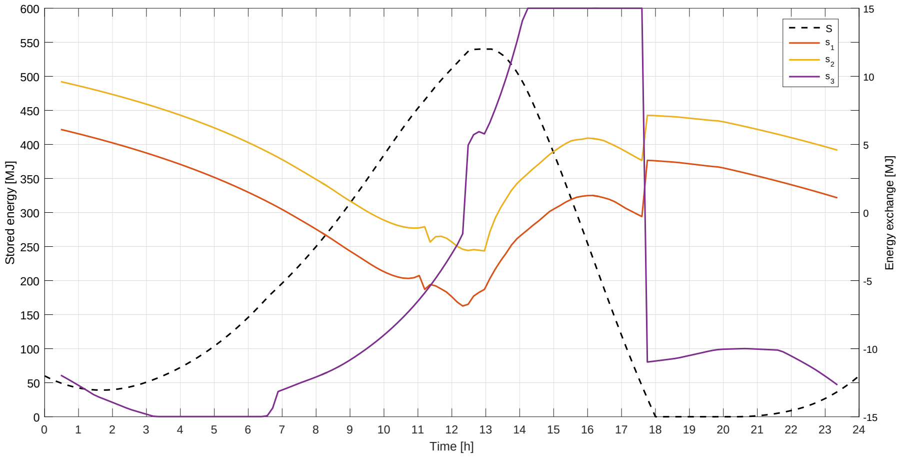
Fig. 5.16: District 3 – Optimal temperature profiles



(a) Building 1



(b) Building 2



(c) Building 3

Fig. 5.17: District 3 – Optimal storage profiles

- Building 1 (Figure 5.16.a) has the same structure of the buildings considered in the previous districts. Since the results are quite similar, we shall not discuss them further.

- For building 2 (Figure 5.16.b), we observe a certain degree of similarity with the behaviours of building 1. In particular, *i*) zone 1 and zone 2 (blue and green lines, which are overlapped), composing the ground floor of building 2, exhibit a behaviour which is similar to the ground floor of building 1; *ii*) zone 3 and zone 4 (red and yellow lines), which compose the first floor of building 2, behave like the middle floor of building 1, with small differences related to different view factors of the external walls; *iii*) finally, zone 5 (purple line) presents the same optimal profile of the top floor of building 1. As a general trend, the middle floor, due to its position, is exploited to store additional cooling energy by performing pre–cooling of the building.

- For building 3 (Figure 5.16.b), the optimal set–points $\boldsymbol{T_{z,1}}$ and $\boldsymbol{T_{z,3}}$ for zone 1 and zone 3 (blue line and red line, respectively) coincide, due to the symmetry of the building structure. The behavior of zone 2, which is slightly more cooled down, is not new and seems to be characteristic of those zones which are in between other zones. We observe the absence of pre–cooling outside the working hours. Considering the dimensions of the building (with a volume of $3000m^3$ per zone), each zone has a very large heat capacity and therefore a very large thermal inertia. For this reason, pre–cooling the zones requires a huge energetic effort by the chiller and therefore it might not be the optimal choice.

Figure 5.17 shows the optimal exchanges with the storage upon convergence. As in the previous cases, the agents reach consensus on the optimal solution.

Due to high energetic request of building 3, the storage is fully exploited, in fact the storage level (black dashed line in Figure 5.17) reaches the maximum capacity, $S_{max} = 540$ MJ around $12 : 30$. Also the storage exchange rate for building 3 has been saturated at 15 MJ (for both positive and negative values) for almost four hours.

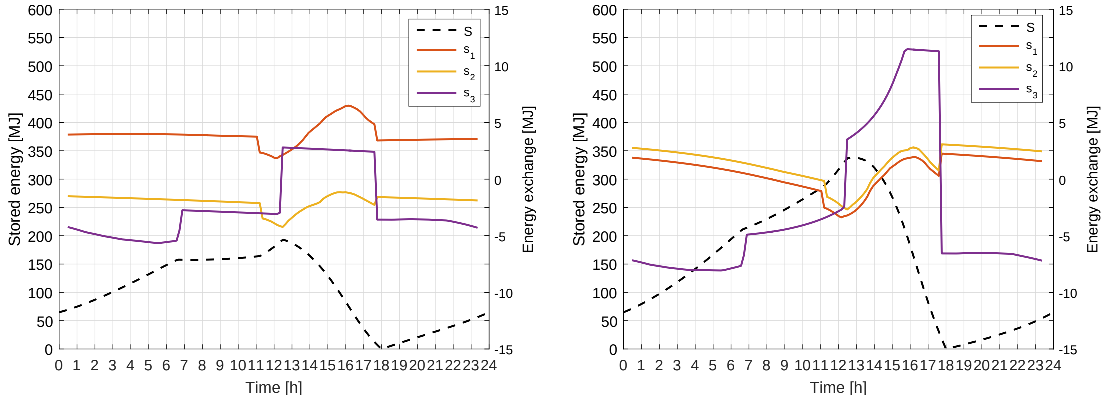
For the same reason, both the other buildings charge the storage before the peak period, when all the energy previously charged is used by building 3 to keep the temperatures set–points of its zones inside the comfort bounds.

Figure 5.18 shows the evolution of the optimal storage profiles as a function of the iteration number k. We show the solution computed by agent 1, with the complete communication graph. The analysis of the tentative solutions shows a behaviour similar to the previous cases.

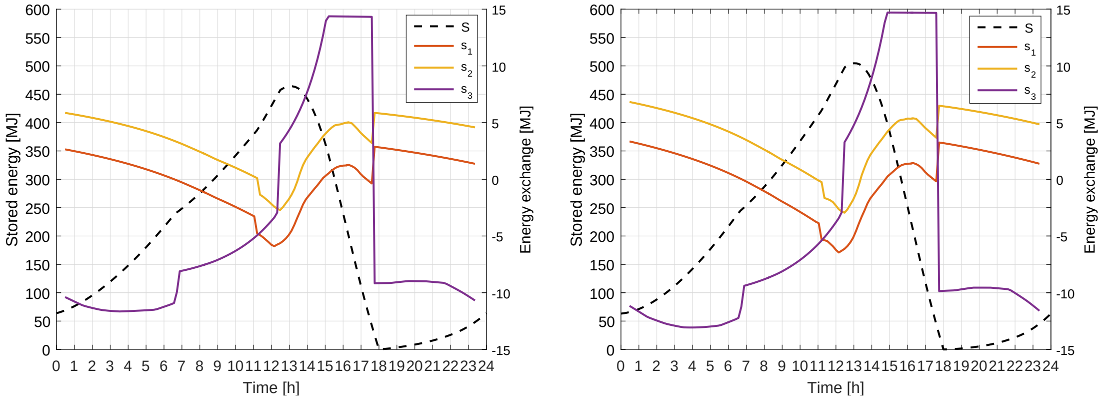
	Complete Network	**Connected Network**	**Time–varying Network**
# of iterations	691	901	1678
Average time			
Building 1	$2.682s$	$2.733s$	$2.646s$
Building 2	$6.083s$	$5.904s$	$6.016s$
Building 3	$2.901s$	$3.004s$	$2.901s$
Overall time			
Building 1	$1853.262s$	$2462.433s$	$4439,98s$
Building 2	$4203,353s$	$5319.504s$	$10094,85s$
Building 3	$2004,591s$	$2706.604s$	$4867,88s$

Table 5.4: Simulation data for District 3

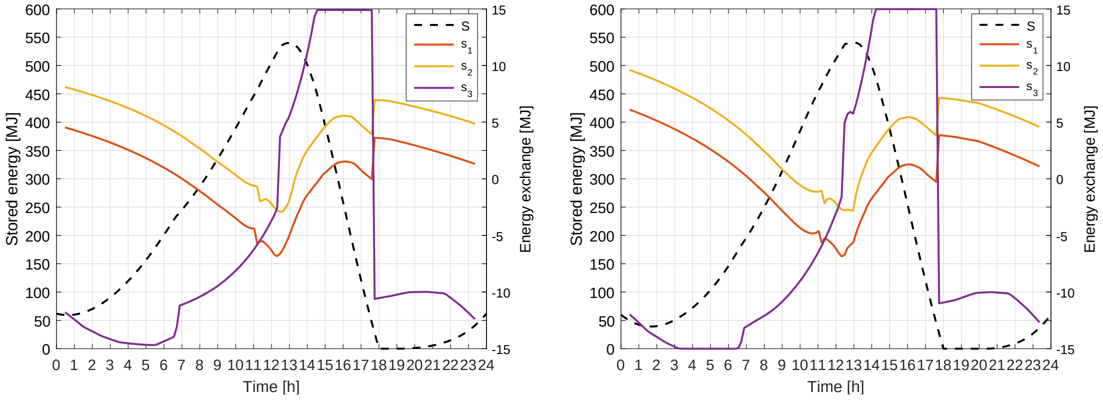
Table 5.4 shows the algorithm statistics for the District 3 case. We observe a major difference of average time per buildings: due to the higher number of zones, building 2 has a larger the set of optimization variables, and consequently agent 2 requires more time to solve the local problem. The number of iterations required is generally higher than the previous cases. As expected, the best performances are achieved with the complete graph, while the time–varying graph leads to the slowest convergence rate.



(a) $k = 1$ **(b)** $k = 10$



(c) $k = 50$ **(d)** $k = 100$



(e) $k = 300$ **(f)** Consensus solution

Fig. 5.18: Evolution of storage usage by the 3 agents and stored energy, computed by Agent 1

5.4 Final remarks

The buildings reach consensus in all the performed test: the specific local optimization problems and the communication graph do not influence the solution obtained after convergence, as expected given the theoretical guarantees of Algorithm 1.

The communication structure affects the flow of information and hence the speed of convergence, as shown by the number of iteration and overall computing times.

The complete graph allows the best information flow, as every building receives information directly from all the others and hence provides the best performance. Conversely, the time–varying network leads to the slowest convergence rate.

The average time required to each agent to solve a single iteration strongly depends on the dimension of the local optimization problem, and consequently on the building structure: a large number of zones implies a large set of optimization variables, and thus a higher computational effort and longer times to reach convergence.

Chapter 6

Conclusions

We presented a distributed algorithm for the solution of the optimal energy management for the cooling of a district network of buildings, each one with its own chiller plant but all of them sharing a thermal storage unit. Each building can set its own set temperature profile and has to agree with the others on the storage usage.

The approach exploits a unified framework that allows to tackle:

- problems with different local constraint sets for each building,
- problems with different local cost functions,
- problems with time–varying communication networks,

allowing to address the problem for districts composed by different buildings equipped with different chiller plants, as well as dealing with communication structures subject to temporary failures.

As shown in Chapter 5 on the numerical results, the distributed approach provides as advantages:

- the computational load is reduced by dividing the effort on multiple computational units, and the obtained reduction grows with the size of the network, given that the computational load grows more than linearly with the number of nodes in the network;
- a certain degree of robustness is guaranteed against network failures.

Finally, in the perspective of privacy protection, the solution is achieved by exchanging information only on the shared resource, while the sensitive information on the energy consumption profiles is kept private.

The focus in this work was purely on a deterministic setting, with nominal disturbances, and the case of stochastic disturbances has not been addressed, as it is not easy to address and goes beyond the scope of this thesis.

Nevertheless, the introduction of stochastic disturbances is surely a topic that deserves an in-depth investigation. In [10] the stochastic problem was addresses for a single building using a *scenario* based approach, where a solution robust with respect to a finite number of possible disturbance realizations is computed while providing probabilistic guarantees on the unseen realizations. [10] actually showed the potentiality of such an approach for the optimal energy management under uncertain disturbances.

A future development of this work should be to investigate the stochastic problem for a district network, combining the results achieved in this work with the potential of the scenario approach to tackle stochastic optimization problems in a distributed setting.

Bibliography

[1] L. Pérez-Lombard, J. Ortiz, and C. Pout, “A review on buildings energy consumption information,” *Energy and Buildings*, vol. 40, no. 3, pp. 394 – 398, 2008.

[2] R. Baños, F. Manzano-Agugliaro, F. Montoya, C. Gil, A. Alcayde, and J. Gómez, “Optimization methods applied to renewable and sustainable energy: A review,” *Renewable and Sustainable Energy Reviews*, vol. 15, no. 4, pp. 1753 – 1766, 2011.

[3] G. W. Arnold, “Challenges and opportunities in smart grid: A position article,” *Proceedings of the IEEE*, vol. 99, no. 6, pp. 922–927, June 2011.

[4] K. Moslehi and R. Kumar, “A reliability perspective of the smart grid,” *IEEE Transactions on Smart Grid*, vol. 1, no. 1, pp. 57–64, June 2010.

[5] National Institute of Standards and Technology, “Workshop on technology, measurement, and standards challenges for the smartgrid,” March 2013. [Online]. Available: http://www.nist.gov/smartgrid/upload/Final-Version-22-Mar-2013-Smart-Grid-Workshop-Summary-Report.pdf

[6] W. Turner, I. Walker, and J. Roux, “Peak load reductions: Electric load shifting with mechanical pre-cooling of residential buildings with low thermal mass,” *Energy*, vol. 82, pp. 1057 – 1067, 2015.

[7] (2012, February) Building energy data book of DOE. [Online]. Available: http://buildingsdatabook.eren.doe.gov/

[8] Y. Ma, F. Borrelli, B. Hencey, A. Packard, and S. Bortoff, “Model predictive control of thermal energy storage in building cooling systems,” pp. 392–397, Dec 2009.

[9] M. B. Bulut, M. Odlare, P. Stigson, F. Wallin, and I. Vassileva, “Active buildings in smart gridsexploring the views of the swedish energy and buildings sectors,” *Energy and Buildings*, vol. 117, pp. 185 – 198, 2016.

[10] D. Ioli, “Optimal energy management of a building cooling system with thermal storage: modeling and control,” Master’s thesis, Politecnico di Milano, 2014.

[11] K. M. Powell, W. J. Cole, U. F. Ekarika, and T. F. Edgar, “Dynamic optimization of a campus cooling system with thermal storage,” pp. 4077–4082, July 2013.

[12] C. Balaras, “The role of thermal mass on the cooling load of buildings. an overview of computational methods,” *Energy and Buildings*, vol. 24, no. 1, pp. 1 – 10, 1996.

[13] D. Ioli, A. Falsone, and M. Prandini, “Optimal energy management of a building cooling system with thermal storage: A convex formulation,” in *Proceedings of the 9th International Symposium on Advanced Control of Chemical Processes. (ADCHEM 2015), Whistler, British Columbia, Canada*, 2015.

[14] M. R. B. Khan, R. Jidin, and J. Pasupuleti, “Multi-agent based distributed control architecture for microgrid energy management and optimization,” *Energy Conversion and Management*, vol. 112, pp. 288 – 307, 2016.

[15] S. Boyd and L. Vandenberghe, *Convex optimization.* Cambridge University Press, 2004.

[16] P. McDaniel and S. McLaughlin, “Security and privacy challenges in the smart grid,” *Security Privacy, IEEE*, vol. 7, no. 3, pp. 75–77, 2009.

[17] F. Marmol, C. Sorge, O. Ugus, and G. Perez, “Do not snoop my habits: preserving privacy in the smart grid,” *Communications Magazine, IEEE*, vol. 50, no. 5, pp. 166–172, 2012.

[18] Z. Wang, K. Yang, and X. Wang, “Privacy-preserving energy scheduling in microgrid systems,” *Smart Grid, IEEE Transactions on*, vol. 4, no. 4, pp. 1810–1820, 2013.

[19] K. Butcher, *CIBSE Guide A: Environmental Design*, 2006.

[20] J. Gordon, K. Ng, and H. Chua, “Optimizing chiller operation based on finite-time thermodynamics: Universal modeling and experimental confirmation,” *International Journal of Refrigeration*, 1997.

[21] I. Dincer, “On thermal energy storage systems and applications in buildings,” *Energy and Buildings*, vol. 34, no. 4, pp. 377 – 388, 2002.

[22] M. Bürger, G. Notarstefano, and F. Allgöwer, “A polyhedral approximation framework for convex and robust distributed optimization,” *CoRR*, vol. abs/1303.6092, 2013. [Online]. Available: http://arxiv.org/abs/1303.6092

[23] D. Mosk-Aoyama, T. Roughgarden, and D. Shah, “Fully distributed algorithms for convex optimization problems,” *SIAM Journal on Optimization*, vol. 20, 2010.

[24] K. Margellos, A. Falsone, S. Garatti, and M. Prandini, “Distributed constrained optimization and consensus in uncertain networks via proximal minimization,” 2015, submitted.

[25] M. Grant and S. Boyd, “CVX: Matlab software for disciplined convex programming, version 2.1,” http://cvxr.com/cvx, mar 2014.

[26] M. Grant and S. Boyd , “Graph implementations for nonsmooth convex programs,” in *Recent Advances in Learning and Control*, ser. Lecture Notes in Control and Information Sciences, V. Blondel, S. Boyd, and H. Kimura, Eds. Springer-Verlag Limited, 2008, pp. 95–110, http://stanford.edu/~boyd/graph_dcp.html.

[27] MOSEK ApS, *The MOSEK optimization toolbox for MATLAB manual. Version 7.1 (Revision 28).*, 2015. [Online]. Available: http://docs.mosek.com/7.1/toolbox/index.html

[28] L. Magni and R. Scattolini, *Complementi di controlli automatici.* Pitagora, 2006.

[29] J. Michaels, “Commercial buildings energy consumption survey (CBECS),” U.S. Energy Information Administration, 2012. [Online]. Available: http://www.eia.gov/todayinenergy/detail.cfm?id=21152

[30] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *Foundations and Trends in Machine Learning*, vol. 3, no. 1, Jan. 2011.

[31] U.S. Department of Energy Microgrid Exchange Group, “DOE microgrid workshop report,” August 2011. [Online]. Available: http://energy.gov/sites/prod/files/Microgrid%20Workshop%20Report%20August%202011.pdf

[32] M. Kintner-Meyer and A. Emery, “Optimal control of an HVAC system using cold storage and building thermal capacitance,” *Energy and Buildings*, vol. 23, no. 1, pp. 19 – 31, 1995.

[33] J. Renegar, *A Mathematical View of Interior-point Methods in Convex Optimization*, ser. MPS-SIAM Series on Optimization. Society for Industrial and Applied Mathematics, 2001.

[34] D. Kim and J. E. Braun, “Reduced-order building modeling for application to model-based predictive control,” in *SimBuild 2012 Conference in. Madison, Wisconsin.*, vol. 1, 2012, p. 2.

Appendices

Appendix A

Model implementation – complete data report

The data used for the problem implementation of Chapter 4 and simulations of Chapter 5 are here fully reported and commented. This data include model parameters, such as materials used and wall types implemented, simulation parameters and disturbances profile.

A.1 Wall materials parameters

The basic materials defined in the database are concrete, walling, insulation and substrate; these are the most common materials used in the buildings, and can be used to describe a wide variety of buildings. They are defined by the parameters summarized in table A.1: name, density, specific heat, thermal conductivity coefficient and the layer width for the finite elements method.

ID #	**Material**	**Density** $[Kg/m^3]$	**Specific heat** $[J/Kg]$	**Conductivity** $[W/(mK)]$	**FEM width** $[m]$
1	Concrete	2400	880	1.6	0.025
2	Insulation	30	2100	0.026	0.010
3	Substrate	1000	880	0.84	0.010
4	Walling	1200	840	0.36	0.025
5	Floor Concrete[1]	2400	880	1.6	0.020

Table A.1: List of wall materials

---

[1] The material is the same as ‘concrete’, but with different finite element width.

A.2 Wall types parameters

The wall type data structure contains all the parameters that uniquely identify a wall, in terms of both wall composition and thermal exchange parameters. Thus, walls are differentiated not only based on the materials they are made of and how these are combined to form the wall layering, but also on the orientation of the wall, which affects the shading factors and the absorption coefficients.

For the composition part, the wall type structure is defined by two arrays:

- The first array is list of materials composing the wall, ordered from the most internal to the most external, and from bottom to top in case of floors and roof.

- The second array contains the number of finite layer for each material in the first array. Obviously, the number of elements of the two arrays must be the same.

Figure A.1 sketch an example on how to correctly model the wall composition using the arrays, while the description of the implemented wall types is in table A.2.

ID #	Description	Materials array	# of layers array
1	External wall, facing north	[4 1 2]	[2 10 8]
2	External wall, facing south	[4 1 2]	[2 10 8]
3	External wall, facing east\west	[4 1 2]	[2 10 8]
4	Roof	[1 2 4]	[5 10 2]
5	Floor	[3 5]	[7 10]
6	Internal wall (symmetrical)	[3]	[5]

Table A.2: List of wall materials

The thermal exchange parameters needed for the wall description are the internal and external convective coefficients and the absorption coefficient for shortwave and longwave radiation, the latter being separated for the upwelling and downwelling radiation. The specific absorption coefficients are expressed as the product of *general* absorption coefficients, ϵ_S and ϵ_L respectively, and the specific view factors, function of the wall orientation and exposure. Table A.3 lists all the coefficient implemented.

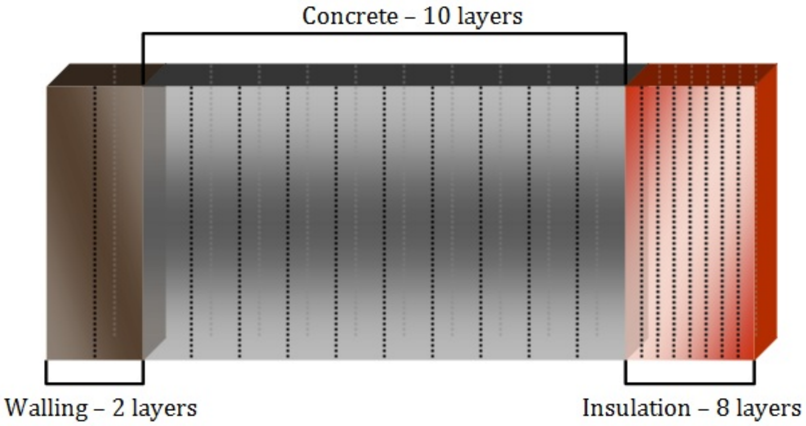


Fig. A.1: Wall modelling example

Consider a wall composed of 3 material: walling, concrete and insulation, order from the inner to the outer one. According to the identification number in table A.1, the materials array will be

$$[4\ 1\ 2]$$

. Now consider a possible finite element subdivision, as in figure: the walling layer is divided in 2 slices, the concrete core in 10 slices and the insulation layer in 8 slices. Thus, the number of finite elements array is, in order from inside to outside,

$$[2\ 10\ 8]$$

.

ID #	h_i	h_e	α^S	α^{Lu}	α^{Ld}
1	30	15.35	$0.3 \cdot \epsilon^S$	$0.6 \cdot \epsilon^L$	$0.5 \cdot \epsilon^L$
2	30	15.35	$0.8 \cdot \epsilon^S$	$0.6 \cdot \epsilon^L$	$0.5 \cdot \epsilon^L$
3	30	15.35	$0.5 \cdot \epsilon^S$	$0.6 \cdot \epsilon^L$	$0.5 \cdot \epsilon^L$
4	30	15.35	$0.8 \cdot \epsilon^S$	$0.8 \cdot \epsilon^L$	0.0
5	0	30	0	0	0
6	0	18	0	0	0
General shortwave coefficient ϵ^S					0.6
General longwave coefficient ϵ^L					0.9

Table A.3: List of wall thermal coefficients

A.3 Disturbances

Lastly, we deal with the external data used for the definition of disturbances. The model requires some deterministic profiles for the disturbances affecting the model. Recalling the disturbance vector from Chapter 2:

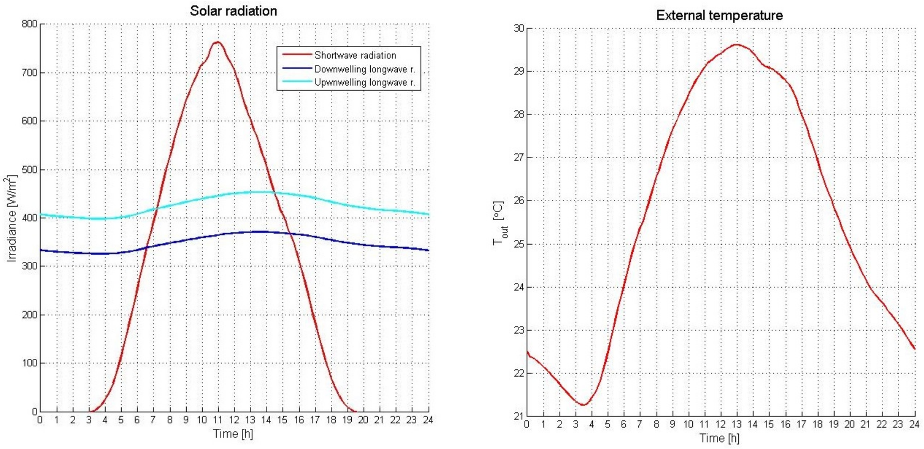
$$\boldsymbol{d} = [T_{out}\ Q^S\ Q^{Ld}\ Q^{Lu}\ 1]^T \tag{A.1}$$

where the longwave incident radiation has been already split in upwelling and downwelling, we need to define the profile over the 2 days horizon for each of the quantity of vector (A.1). The data necessary for the profiles definition has been taken from [10],[13] based on the data collected by the Council of National Research of Bologna. As explained in [10], the raw data has been treated with appropriate data conditioning before the implementation. The disturbances profiles (over the single day horizon) are shown in figure A.2. The profiles are identically replicated for each day of the horizon.

In addition to disturbances, we need to account also for the occupancy profile for each zone, as well as the unitary energy price profile for the optimization cost function, shown in figure A.3. The occupancy profile in figure A.3.a shows the presence of people for the whole building, as in the single–zone test set–up. Conversely, for multi–zone tests, the profile has been equally divided among all the zone, making the people distribution in each k-th zone equal to $N_{p,k} = N_p/N_z$ (rounded to the nearest integer), with N_z being the number of zones and N_p the number of people for the whole building. Finally, for the occupancy heat equation

$$E_{p,j} = q_{2,k}(n_p)T_{z,j}(k) + q_{1,k}T_{z,j}(k-1) + q_{0,k}\ , \tag{A.2}$$

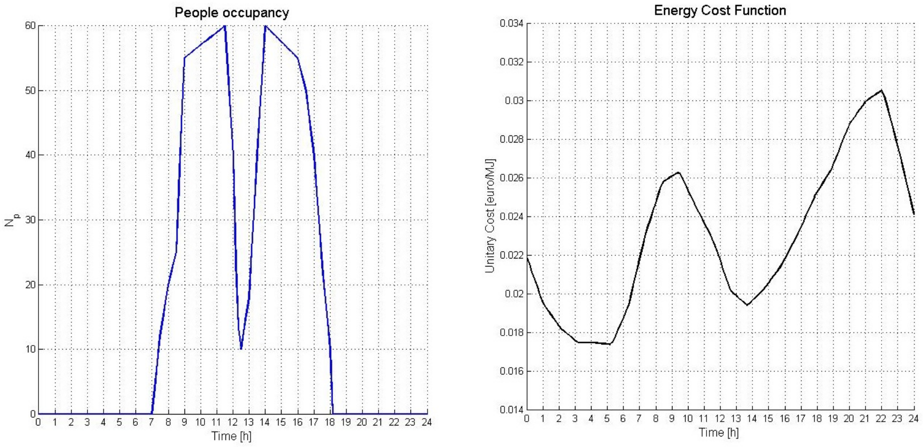
the empirical coefficient has been set to $p_2 = -0.2199$, $p_1 = 125.125$ and



(a) Solar incident radiation **(b)** Outside temperature

Fig. A.2: Model exogenous disturbances profiles over single day horizon

$p_0 = -1.7685 \cdot 10^4$.



(a) Occupancy profile

(b) Energy unitary cost profile

Fig. A.3: People occupancy and energy unitary cost profiles

List of Figures

List of Tables