Percorso dottorale sviluppato con il sostegno finanziario di NextGenerationEU:
Missione 4, Componente 2, Investimento 3.3, CUP B63D22000960008
Borsa MUR exDM352/2022, cofinanziata da Barilla G. e R. Fratelli SpA, ciclo 38°

Dipartimento di Ingegneria dell'Informazione e Scienze Matematiche

Dottorato in Ingegneria e Scienza Dell'Informazione

XXVIII° Ciclo

Coordinatore: Prof. Mauro Barni

Optimization Models and Algorithms for Decision Making in Agriculture

Settore scientifico disciplinare: MAT/09 - Operations Research

Candidato
Luca Nerozzi
Università degli Studi di Siena

Firma digitale del/della candidato/a

Supervisore
Paolo Detti
Università degli Studi di Siena

Firmato digitalmente da
LUCA NEROZZI
03/03/2026 17:40:28
con la Firma Digitale cheFIRMA!

Anno accademico di conseguimento del titolo di Dottore di ricerca
2024/25

Università degli Studi di Siena
Dottorato in Ingegneria e Scienza Dell'Informazione
XXXVIII° Ciclo

Data dell'esame finale
19 / 03 / 2026

Commissione giudicatrice
Alessandro Agnetis – Professore Ordinario, Università di Siena
Andrea Pacifici – Professore Ordinario, Università di Roma "Tor Vergata"
Maria Grazia Scutellà – Professoressa Ordinaria, Università di Pisa

Supplenti
Gaia Nicosia – Professoressa Ordinaria, Università di Roma Tre
Marco Pranzo – Professore Associato, Università di Siena

Contents

List of Figures

List of Tables

Introduction

Growth and development are at the basis of our society. In every domain, progress is shaped by the ability to allocate limited resources, coordinate activities, and continuously improve performance while complying with technical, economic, and institutional constraints. As systems expand and become more interconnected, decision-making becomes harder: choices that were once simple and local turn into decisions with multiple objectives, long-term consequences, and non-trivial interactions. In this landscape, relying exclusively on intuition or experience is rarely sufficient. Effective decisions increasingly require analytical support, rigorous reasoning, and tools that can translate data and constraints into actionable plans.

Agriculture exemplifies this challenge in a particularly clear way. It is a strategic sector for food security and economic stability, yet it operates under strong resource limitations and increasing pressure to meet sustainability goals. Farmers and stakeholders have to face with uncertainty driven by climate variability, biological dynamics, and market volatility, while respecting agronomic rules, operational constraints, and regulations. At the same time, the transition toward sustainable practices calls for decisions that explicitly account for environmental impacts, long-term soil health, biodiversity preservation, and responsible use of inputs (e.g. water, fertilizers). These requirements make many agricultural choices inherently multi-criteria and tightly constrained, often spanning multiple seasons and spatially distributed operations.

The set of quantitative disciplines broadly referred to as Decision Science offers a coherent perspective to address such complexity by placing decisions, trade-offs, and measurable outcomes at the center of the analysis. Within this framework, Operations Research provides a methodological backbone for building decision-support systems grounded in mathematical modeling. By representing a decision problem through variables, constraints, and objective functions, Operations Research enables the systematic evaluation of alternative strategies and the identification of solutions that improve performance under realistic limitations.

This work mainly adopts mathematical optimization methods, with a strong emphasis on combinatorial optimization. Many relevant agricultural decisions are discrete in nature—for instance selecting crops, assigning land to activities, scheduling operations, or designing service routes—and lead to large-scale optimization models where the number

of feasible alternatives grows rapidly. In these settings, straightforward enumeration is computationally impractical, and effective decision support hinges on the design of appropriate formulations and efficient solution algorithms, including exact approaches and tailored heuristics able to deliver high-quality solutions within practical time limits.

The core of this dissertation consists of two main chapters, each addressing an optimization problem in the agricultural domain. The first chapter investigates a crop planning problem in which crop rotations and sustainability-oriented constraints play a central role. The objective is to maximize the farmer's profit over a multi-period planning horizon while ensuring agronomic feasibility and explicitly incorporating sustainability requirements arising from both public regulations and private schemes. The second chapter addresses a vehicle routing problem arising in fertilization operations in viticulture. In this context, fertilizer demand varies across vineyard plots, and routing decisions must account for operational constraints such as vehicle capacity limits, binary refilling at the depot, and the possibility of split deliveries. The objective is to minimize total travel distance while simultaneously constructing an operational schedule that supports efficient fertilization activities and an effective organization of working hours.

Overall, the research developed in this thesis follows a common workflow: understanding the decision context and the available data, formulating mathematical models that capture the key constraints and objectives, and designing and implementing algorithms that can effectively solve the resulting problems. Most of the material presented in the remainder of this dissertation is grounded in research conducted during the Ph.D. program and has been published, accepted for publication, or is currently under review and/or in preparation for submission.

The list of related publications and submitted manuscripts is reported below.

- Benini M., Detti P., Nerozzi L., Optimization models and algorithms for sustainable crop planning and rotation: An arc flow formulation and a column generation approach, *Omega*, Volume 135, 2025, DOI: 10.1016/j.omega.2025.103320.
- Cappella M.T., Nerozzi L., Benini M., Detti P., Blasi E., Evaluating the economic and land use consequences of sustainability-driven CAP reforms: Insights from an optimization approach in Italian cereal farms, *Land Use Policy*, Volume 158, 2025, DOI: 10.1016/j.landusepol.2025.107720.
- Nerozzi L., Benini M., Detti P., Blasi E., Cappella M., A Heuristic Approach for Sustainable Crop Rotation Planning, *Shaping a Sustainable Future in the Era of Big Data - AIRO Springer Series*, (forthcoming).
- Nerozzi L., Girardot R., Valloo Y., Gras J.P., Benini M., Detti P., Tisseyre B., Combining Site Specific Nitrogen Application and Optimal Routing in Viticulture: A Low Cost Approach, (in preparation).
- Benini M., Detti P., Nerozzi L., Split Delivery Vehicle Routing Problem with Binary Refilling, (in preparation).

Chapter 1

Crop Rotation Problem

The importance of sustainable agricultural practices has grown significantly in recent years, particularly due to the escalating impacts of climate change. Moreover, the United Nations' 2030 agenda explicitly identifies sustainable agriculture as a key instrument for achieving multiple Sustainable Development Goals (SDGs). In particular, the improvement of agricultural production mandated by Goal 2, Zero Hunger, must be achieved through a more conscientious management of natural resources [1].

Crop rotation is a sustainability practice that can address the issues arising from monoculture schemes commonly used in intensive agriculture [2, 3]. It involves alternating different crop species on the same plot of land over consecutive seeding periods to preserve soil fertility and maintain agricultural productivity. Indeed, long-term experimental data from various production contexts have revealed several issues associated with monoculture and intensive agricultural practices, including the loss of soil fertility and increased costs due to more complex phytosanitary management of crops. This often leads to the excessive use of pesticides and industrial fertilizers. Concurrently, the agronomic literature has shown that crop rotation can significantly reduce parasite infestations and enhance soil fertility [4].

Crop rotation has therefore been introduced as a new commitment into the Good Agricultural Environmental Conditions (GAEC) list published in the Common Agricultural Policy (CAP) [5] proposal of the European Union (EU), and has repeatedly been included as a key element in sustainable production schemes promoted by both public and private initiatives [6]. In particular, the latest CAP reform for the period 2023–2027 represents a turning point, introducing mechanisms such as *Eco-schemes*, stricter GAEC standards, and conditional subsidies. These measures aim to align agricultural support with environmental and climate objectives, promoting biodiversity, soil health, and the reduction of chemical inputs [7, 8].

At the same time, the increasing complexity and stringency of the CAP regulations, especially regarding crop allocation and land use, has raised concerns among farmers about economic sustainability. This tension culminated in widespread farmer protests across Europe in 2023 and early 2024, highlighting fears that new environmental rules could compromise farm profitability, particularly for small and medium-sized enterprises [9]. In response, the EU has adopted several derogations aimed at granting more flexibility to pre-existing practices and alleviating compliance burdens.

Within this evolving scenario, this chapter presents a study of the CAP reforms (CAP 2014–2020, CAP 2021–2023, CAP 2023–2027) and how they have influenced, and may further affect, the profitability of arable farms and reshape production strategies. The analysis focuses on Italian cereal farms operating in highly intensive agricultural contexts,

assessing how enhanced sustainability constraints have affected production practices, particularly in terms of crop rotation, land use, and resource allocation.

To promote sustainable practices more effectively, the CAP not only includes regulatory mechanisms but also offers economic incentives to farmers who comply with specific production and crop planning constraints [10]. Complementing these public incentives, private initiatives such as the "Carta del Mulino" (CdM), introduced by the Barilla Group, reward farmers who adopt crop diversification and biodiversity-enhancing strategies with additional financial compensation. These private schemes work in synergy with CAP regulations, offering both economic and contractual advantages to compliant farmers.

On the other hand, farmers' main objective is to plan crop seeding on their land in an economically sustainable and profitable manner. In this complex context, farmers face increasingly challenging decisions in order to optimize their land use, making it difficult to evaluate the impact of joining sustainability initiatives such as the CAP and the "Carta del Mulino" (CdM). As a result, the attention is increasing for decision-support tools able to help farmers to maximize their revenues, while respecting sustainability principles and regulations [11].

This chapter presents decision-support models and algorithms designed to help farmers to optimally plan their crop production over specified time horizons, considering sustainable crop rotation practices and constraints imposed by both public (CAP) and private (CdM) frameworks. The models are capable of simulating farm behavior under different policy scenarios allowing for a comparative analysis of the economic and operational impact of each policy configuration. Although the proposed approaches are general and can be easily adapted to different scenarios, the focus of the experimental analysis are on the Mediterranean pedo-climatic context. The proposed decision-support methods prescribe the most profitable crop rotation plans that comply with all sustainability constraints. They also provide valuable information to farmers, enabling them to evaluate the agricultural and economic impact of these sustainability regulations through farm-level quantitative economic indicators. Additionally, regulatory bodies can use these methods to assess the impact of their sustainability initiatives and the suitability of rules and incentives for promoting farm sustainability transition.

1.1 Crop Rotation

Crop rotation is one of the most important aspects in agriculture and it is well studied in the agricultural literature since it prevents problems caused by monoculture practices. In agriculture, crop rotation refers to the succession of different crops on the same piece of land over consecutive seeding periods. In fact, monoculture schemes, in which the same crop is assigned to the entire farmland over the whole planning horizon, lead to a series of issues (such as loss of yield and soil fertility, increase of weed and pest diseases) that can be mitigated by applying suitable rotation of crops. The problems caused by intensive agriculture and monoculture practices have been the subject of numerous research over the years, both technical-agronomic and socio-economic [3, 12]. To overcome these problems, movements of scholars, technicians and civil society proposed more sustainable arable land use patterns, inspired to agroecology principles [13]. The principles of the

International Federation of Organic Agriculture Movements (IFOAM) were used worldwide to define specific regulatory framework for organic agriculture. In Europe, since the publication of the first regulation on organic farming in 1992, the importance of crop rotation has been translated into rotation schemes that can be adopted by farmers. In fact, it was highlighted the importance of the inclusion of leguminous plants or crops whose management had improving effects on the structure and, more generally, on the natural fertility of the soil [14, 15].

For instance, in Italy, the viable crop rotation schemes for organic agriculture are defined by ministerial decree [16]. In general, the *best crop rotation practices* involve assigning a sequence of consecutive crops to the same plot in such a way that the loss of yield is minimized and the soil fertility is preserved. In this context, fallow can be considered a crop itself. Any deviation from these best practices typically results in increased production costs for the crops to achieve the same yield, due to the additional labor required on the soil, such as the use of fertilizers and pesticides. Consequently, the production cost and the profit derived from cultivating a crop on a particular plot depend on the crops that were previously grown on that plot. Specifically, in the Mediterranean pedo-climatic context (e.g., as prescribed by Italian regulations on organic agriculture [17]), the best rotation practices involve sequences of *three crops*, and crops are classified into three types: *renewal, impoverishing,* and *improver*.
The characteristics of renewal, impoverishing and improver crops are described below:

- *Renewal crops*: these crops (e.g., corn, sugar beet, potato, tomato, sunflower, etc.) require particular care consisting in excellent soil preparation and balanced organic fertilizations, which has a positive effect on the structure of the soil. However, in some specific contexts, also a fallow period could be considered as renewal soil quality practice.

- *Impoverishing crops*: they exploit the nutritional elements present in the soil and deplete it. Crops in this class are wheat, oats, barley, rye, rice, corn, sorghum and generally all grain cereals.

- *Improver crops*: they increase the fertility of the soil, enriching it with nutrients. Improver crops mainly are legumes, such as alfalfa or clover, which are able to fix atmospheric nitrogen.

As explained above, proper crop rotation schemes may bring many advantages to the farm, both of an agronomic and economic-managerial perspective [18]. On the other hand, when crop rotation does not follow the best agronomic practice, in order to keep their yield stable over time, farmers will have to compensate for the loss of soil fertility and the increase in weeds and pests' risks by employing technical inputs, such as herbicide fertilizers and pesticides. The use of technical inputs leads to an increase in production costs, directly related on the specific crop succession performed in the rotation scheme. Summarizing, in general, all crop successions are feasible, but specific (additional) production costs may arise when crop patterns do not follow best crop rotation practices [14].
In the Mediterranean pedo-climatic context, it is generally assumed that the costs and profits of a crop directly depend on the *two* crops previously assigned to the same plot.

The best rotation schemes are those that involve sequences of the three crop types that do not result in any increase of the production costs. For example, the three-crop-type rotation of *renewal—impoverishing—improver*, in this order, is recognized as a classical best practice in crop rotation [17].

As an example, Figure 1.1 reports 3-crop rotations following and not following best practices.

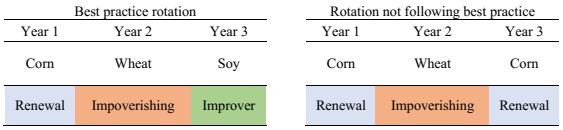

Figure 1.1: 3-crop rotations following and not following best practices.

Another critical factor in crop rotation is the *maximum crop replanting*, which refers to the number of times a given crop family can be consecutively replanted on the same plot of land within a certain time interval. Although not adhering to best rotation practices can lead to increased costs, crops cannot be replanted indefinitely on the same land without incurring prohibitively high production costs, even if technically feasible. This is because continuous replanting of the same crop without alternating with different crops disrupts soil health and fertility.

1.2 The problem of planning crop rotations

In this chapter, the problem of planning the allocation of crops to arable lands, taking into account crop rotations principles and diversification strategies promoted by sustainable agriculture, is addressed. More specifically, the problem consists in deciding, for each plot of a farmland, the sequence of crops to plant over a finite number of seeding periods (i.e., the planning horizon). This decision-making process is influenced by two key factors:

1. the profit and yield of a crop directly depend on the sequence of crops immediately preceding it on the same plot;

2. the incentives and requirements defined by sustainability regulations, such as the CAP regulations of the EU [5] and the "Carta del Mulino" (CdM) rules of the Barilla Group [19].

The addressed problem was first introduced in [10], where an Integer Linear Programming (ILP) model was proposed and tested on real-world instances, showing the potentialities of a decision-support tool to help farmers optimally plan their cultivation strategies and assess the benefits of participating in sustainability initiatives. Given the critical role of sustainability in agriculture, it is essential to enable as many farmers and agricultural businesses as possible to adopt sustainability practices and possibly join initiatives like the CAP and the "Carta del Mulino" (CdM). A decision-support tool, capable of aiding farmers in making complex decisions about crop planning and sustainability policy evaluation, can accelerate the adoption of such practices. However, such a tool must be

flexible and scalable to accommodate diverse agricultural realities, while the strongly NP-hardness of the problem (see Section 1.4) poses significant computational challenges in practice. Although promising in many scenarios, the ILP model in [10] has limitations in scalability and adaptability. Specifically, the model's performance is strongly influenced by the farm's size, particularly by the dimension of the arable land area, making it challenging to scale. Moreover, the model is tailored to manage crop rotations of three crops, where the profit of a crop depends on the preceding two crops, limiting its flexibility in different contexts and scenarios. To address these challenges, new optimization models and algorithms that can serve as the core of decision-support systems for crop planning are proposed. Specifically, a novel arc flow formulation is devised, where variables are associated with graph arcs, each representing the assignment of a sub-sequence of crops to specific seeding periods. In this way, all crop rotation and sequencing issues are enclosed in the graph, and do not affect the arc flow formulation, ensuring flexibility for various contexts and constraints. In fact, the arc flow formulation easily allows the modeling of rotation schemes with any number of crops, overcoming the rigidity of the model in [10] (based on three crop rotations). Furthermore, the flow on an arc indicates the number of plots following the crop rotation defined by that sub-sequence, thereby overcoming the scalability limitations related to farm size. The arc-flow formulation is proposed to model various scenarios introduced by the CAP regulations and the "Carta del Mulino" (CdM) initiative. To address these scenarios, both exact and heuristic solution methods are developed. While the exact approach is tractable for small- to medium-scale instances, larger and more complex agricultural cases present significant computational challenges. To overcome these limitations, a dynamic programming-based algorithm and a column generation-based matheuristic are proposed to efficiently obtain high-quality solutions. Furthermore, a Lagrangian relaxation approach is developed to efficiently compute tight bounds for the problem. The models and algorithms presented are designed to form the core of a decision-support system capable of conducting various analyses to improve crop planning and land management. More precisely, they prescribe the best crop rotation for each plot, while adhering to agricultural, management and sustainability constraints. The proposed methods support what-if analyses, enabling farmers to make informed decisions regarding crop mix selection, arable land subdivision, and participation in sustainability initiatives.

Extensive computational experiments have been conducted to assess both the effectiveness of the proposed arc-flow models and algorithms. These experiments include real-world instances derived from real farm data, as well as newly generated instances designed to test the limits of the approach under diverse conditions. Results show that the proposed methods significantly outperform the baseline ILP model in [10] in terms of computation time and solution quality, especially when the size of the farmland increases or when more complex regulatory and agronomic constraints are introduced. Moreover, the experiments confirm that the proposed approach can flexibly handle a wide range of sustainability scenarios (e.g., CAP compliance, participation in private schemes like CdM), making it a robust decision-support tool for modern agriculture. Experiments show that the dynamic programming–based algorithm and the column generation–based matheuristic produce high-quality feasible solutions, and that the bounds obtained from

the Lagrangian relaxation are tight.
Public authorities and private organizations can use the proposed methods to evaluate the impact of sustainability policies. Specifically, a fine-tuning of the economic incentives can be performed to achieve the desired level of farmer participation. Moreover, the methods aid in predicting the effect of sustainability policies on farmers' decisions regarding land and crop usage and, so, the impact on crop diversity and food security.

1.3 Literature review

In this section, a review of strategies and methodologies employed in the literature for crop planning management is provided. In the literature, crop planning and rotation problems have received considerable attention from the operations management and agricultural economics communities. Numerous analytical and computational models have been developed in the literature to address crop planning and rotation problems, with Operations Research playing a prominent role in these studies (for comprehensive reviews see [20, 21, 22, 23]). The primary contributions in the field of crop rotation can be broadly classified into two main research streams, where the principal objective is either minimizing the cultivated land or maximizing farmers' profits.

Land management

Many works in the literature propose rotation models where the objective is the minimization of the cultivated land, in a sustainable development context.

Alfandari *et al.* [24] address a crop planning problem within the framework of sustainable forest development in Madagascar. The objective is to minimize the cultivation area while meeting seasonal demands, aligning with a sustainability campaign against deforestation. In the problem, a plot can be fractionated and assigned to multiple crops in each period. The authors propose a Mixed Integer Linear Programming (MIP) model for the problem, where crop rotation is represented through precedence constraints among crops and enforced fallow periods after a limited number of cultivation periods. The proposed model employs the crop rotation framework developed by Dogliotti *et al.* in [25]. The yield of a crop on a plot is influenced by the number of fallow and non-fallow periods preceding the crop. In [26], a problem similar to the one addressed in [24] is considered. The objective is to plan crop cultivations and fallow periods on land plots of different size, over a given time horizon, to minimize the total land surface used while meeting crop demands in each period. A single crop can be cultivated on each plot at each period. The problem is modeled using the concept of a *state* of a plot, defined by the current crop and the number of preceding fallow and non-fallow periods. In this formulation, fallow is enforced after a limited number of periods, and not all state transitions are allowed, meaning that certain crop rotations are infeasible. An ILP formulation is derived using a Dantzig–Wolfe decomposition approach, and the authors propose a branch-and-price-and-cut algorithm along with a matheuristic to solve the problem.

Santos *et al.* [27] consider a crop rotation problem with the objective of minimizing the total planting area, where crop rotation consists in regularly inserting fallow and green

manure periods in the crop schedule, and forbidding certain crop successions. The land is composed of indivisible standard plots with the same characteristics and size (multiple of a base area). A branch-and-price-and-cut algorithm is proposed for the problem. Santos *et al.* [28] propose a heuristic approach based on column generation for the crop planning and rotation problem with the objective of maximizing plot occupation in the planning horizon. The same rotation constraints of [27] are considered, and planting constraints for adjacent plots are introduced. Forrester and Rodríguez [29] developed a mixed-integer programming model for crop rotation planning in an organic farm, aiming to minimize the deviation from actual crop demand while satisfying agronomic constraints on soil fertility, pest control, and irrigation. Crop rotation rules were modeled through binary restrictions preventing incompatible crop successions. The model produced feasible multi-year plans that met demand and improved compliance with rotation and land management practices.

In this chapter, we address a crop rotation planning problem with the objective of maximizing the farmer's profit, thereby aligning with the profit maximization research stream (see Section 1.3). Differently from the studies mentioned above, we explicitly account for how the profit of a crop is influenced by the succession of its preceding crops (i.e., not limited to fallow and green manure). In our problem, the farmland is divided into indivisible plots with varying sizes and characteristics. Among the referenced works, such a feature has only been considered in [26]. Furthermore, we incorporate several agronomic and management requirements, along with constraints imposed by sustainable regulations, as detailed in Sections 1.4.3 and 1.5.

Profit maximization

Many authors address crop rotation problems with the aim of maximizing the farmer's profit. Nordblom and Thomson [30] developed a LP model integrating crop rotations and livestock production, with the objective of maximizing total farm profit under land, labor, and resource constraints. The approach relied on static crop rotation schemes and did not consider the full set of feasible crop sequences or policy-related sustainability constraints. Detlefsen and Jensen [31] addressed the problem of finding an optimal crop rotation, in terms of gross return, for a given selection of crops on a given piece of land, where profits and costs depend on sequences of at most three crops. The problem is modeled as a minimum cost network flow problem, and no additional constraints are incorporated in their study. Salassi *et al.* [32] employed network flow models to address crop rotation decision problems, focusing on maximizing net returns. Specifically, the proposed models consider a two-year rotation involving three different crops, where each network node represents a specific crop planted in a particular year. The network flow models proposed in the above studies are simpler than the one presented in this work, as they do not consider additional sustainability requirements, exclude maximum crop replanting, and only consider rotation sequences of limited length. Galán-Martín *et al.* in [33] propose an LP model for solving a crop planning problem, where land can be assigned to different crops in each period, and crop rotation is simply performed by imposing that the same crop cannot be grown in the same area in the next season. CAP 2014–2020 constraints are also considered, and the objective is of maximizing the total revenue. Santos *et al.* [34], addressed a crop

planning and rotation problem arising in the Brasilian context, similar to that considered in [26], where the problem is to determine the division of the available heterogeneous arable areas in plots and, for each plot, find a crop rotation schedule. The objective is to maximize the total production return. In the problem, crop rotation is modeled through the interdiction of certain crop successions, and the regular insertion of fallows and green manures. Column generation was applied to a Linear Programming model where the continuous variables represent the surface area assigned to a given rotation. In [35], a crop rotation problem with the same rotation constraints of [27], with identical plots and adjacency constraints, limiting the cultivation of crops from the same botanical family in neighboring plots, is addressed. The objective is of maximizing the profit over the planning horizon. An improved mathematical model is proposed and five different relaxation approaches are presented. In Fikry *et al.* [36], an ILP model is presented for a crop rotation problem, where the objective is to maximize the net return, taking into account constraints on the maximum allowable frequency for planting each crop during the rotation cycle in each plot. The proposed model also incorporates a novel timing preference constraint, along with the agronomic, water availability and seasonal demand restrictions.

Haneveld and Stegeman [37] propose a Linear Programming model where crop rotation constraints are given in the form of inadmissible sequences of crops, i.e., not all possible crop sequences are considered. In the model, the continuous decision variables represent the areas where a certain admissible sequence of crops is cultivated. No other sustainability requirements are considered in the model.

Boyabatli *et al.* [2] proposed a stochastic dynamic programming model for crop planning in sustainable agriculture, where the crop rotation benefits across growing seasons are considered. The presence of revenue uncertainty is also considered, where revenue is stochastically larger and farming cost is lower when a crop is grown on rotated farmland. However, crop rotations only include the succession between two crops (i.e., corn and soybeans). Yang *et al.* [38] formulated an ILP model to maximize the total profit of crop cultivation under risk scenarios. The approach optimized annual crop allocation considering yield variability, production costs, and market conditions, but did not explicitly model crop rotation sequences or agronomic dependencies among crops.

Albornoz *et al.* [39] present a zone delineation and crop planning problem that involves selecting the crops to plant in different management zones satisfying a given demand requirement. The optimal crop rotation, chosen from a set of possible rotations, must be assigned to each zone. In [40], the fuzzy goal programming technique is applied to land use planning, where the levels of various production of crops as well as the total expected profit from the farm are fuzzily described. Filippi *et al.* [41] addressed the problem of crop planning with the objective of maximizing farmer profit while incorporating risk management strategies. The presented models account for crop rotation, embedding it as a measure to manage yield uncertainty.

In Li *et al.* [42], an operational model is proposed, that considers a crop rotation scheduling problem for an investor that offers contracts to many smallholder farmers. A heuristic algorithm is developed to identify the optimal rotation scheduling that would achieve both objectives of maximizing prices and minimizing the profit differences between

smallholder farmers. Capitanescu *et al.* [43] presents a novel multi-stage optimization model for farm management during a certain planning time horizon. The proposed model produces an optimal crop rotation plan which maximizes farmers' profit while satisfying specified environmental constraints and crop rotation schemes of two set of crops. Cortignani and Dono [44] propose a study on the possible effects of the crop diversification imposed by CAP regulations on arable farms in Italy. At this aim, an aggregate mathematical programming model is used on data of about 2800 farms of the Farm Accountancy Data Network.

The problem addressed in this paper was first considered in [10], where an ILP model was proposed. In the problem, differently from the above studies, all possible rotation schemes are allowed for a given general crop set, and the yield and profit of a crop on a plot is linked to the $k-1$ crops immediately preceding it on that plot (where k is a parameter depending on the pedo-climatic context, with $k = 3$ in the Mediterranean area). This feature, along with the sustainable requirements coming from sustainable regulations, introduce an additional layer of complexity to the addressed crop rotation planning problem. With respect to [10], in this paper we propose new, more efficient, models and solution approaches, able to solve large instances with a smaller computational effort. In particular, a graph-based formulation is proposed and compared with the ILP model presented in [10], showing improved computational performance (see Section 1.9). The formulation is then applied across all CAP-defined scenarios. Moreover, its flexibility with respect to existing approaches in the literature is emphasized, as the method can be adapted to different pedo-climatic contexts and policy frameworks.

1.4 Notation and problem definition

In this section, notation is introduced and the addressed crop rotation problem is formally defined. In practice, the problem consists in assigning crops to plots of a farmland over a set of time periods, one crop for each plot in each period, in such a way that management, agronomical and sustainability constraints are fulfilled, and the total profit is maximized.

In Section 1.4.1, the three main elements of the problem are described, i.e., the farmland, the crops and the planning horizon. Then, the profit function is formally defined (Section 1.4.2), and the management and agronomic (Section 1.4.3) constraints are introduced. The problem definition is given in Section 1.4.4. The sustainability constraints are introduced in Sections 1.5.1 and 1.5.2.

1.4.1 Farmland, crops and planning horizon

Let H be the set of plots on a farmland to be cultivated. Two plots are considered *homogeneous* if they share the same size and characteristics, such as soil fertility, irrigation availability and lay of the land, which influence the cultivation of crops in terms of profit, yield, and cost. The plots in H are partitioned into classes, say $H_1, H_2, \ldots, H_m$, of *homogeneous* plots with the same size and characteristics. We denote as s_w the size (i.e., the surface) of each plot in class H_w, with $w \in \{1, \ldots, m\}$, and as $S_{\text{tot}} = \sum_{w=1}^{m} s_w |H_w|$ the

total farmland surface, where $|H_w|$ is the number of plots in class H_w.

Let $T = \{1, \ldots, p\}$ be the planning horizon composed of p seeding periods. Since, without loss of generality, we assume that the planning horizon begins at period 1, in the following p is used to denote both the last period and the length of the planning horizon, i.e. $p = |T|$. As we are considering the European context, production periods in T correspond to semesters of consecutive years: either "first" semesters, starting in October, or "second" semesters, starting in April. Let the periods with odd indices in T be first semesters, while the periods with even indices be second semesters. We denote by T_o the set of odd periods and by T_e the set of even periods, with $T = T_e \cup T_o$, and assume that the last period p of T is an even period.

The set of crops that can be cultivated on the farmland is denoted as $C = \{c_1, c_2, \ldots, c_n\}$. The crops in C are partitioned into three disjoints sets, denoted as C_A, C_F and C_S, containing the *annual*, *first semester* and *second semester* crops, respectively. Each annual crop in C_A can only be planted on a plot in odd periods $t \in T_o$ (first semesters) and occupies the plot for the whole year. Furthermore, annual crops cannot be harvested before a minimum number of years. We denote by d_c the *duration* of an annual crop c, indicating the number of *periods* after which it can be harvested. Hence, if an annual crop c is planted in period t, it cannot be harvested before period $t + d_c - 1$. For example, if an annual crop c is planted in $t \in T_o$ and has a duration of $d_c = 4$ periods, it will be harvested at $t + 3 \in T_e$, i.e., at the end of the second semester of the next year. Furthermore, we allow a crop c of duration d_c to be planted in a period t, such that $t + d_c - 1$ is bigger than the last period of the planning horizon p, i.e., we assume that the crop can be harvested after the end of the planning horizon, in such a way that its duration is respected.

The first semester crops from C_F must be planted at the beginning of an odd period $t \in T_o$ and harvested at the end of that same period. The second semester crops in C_S can only be sown during a second semester period $t \in T_e$ and exclusively on plots where a crop of C_F was planted in the first semester $t - 1 \in T_o$ of the same year. They are harvested at the end of the sowing period $t \in T_e$.

To summarize, *compatible* periods for planting a crop $c \in C_F \cup C_A$ are all the odd periods $t \in T_o$, while *compatible* periods for the crops in C_S are all the even periods $t \in T_e$.

In the problem, it is assumed that a crop from $C_F \cup C_A$ must be assigned to each plot in every odd period $t \in T_o$, ensuring that no plot remains uncultivated during these periods. In fact, the uncultivated land, i.e., fallow, is modeled by introducing a *dummy fallow crop* denoted as c_{fallow} and belonging to the crop set C_A. Second semester crops from C_S may or may not be assigned to even periods $t \in T_e$, and only if a first semester crop was assigned to the preceding odd period $t - 1$.

As reported in Section 1.1, crops are also classified as *impoverishing*, *renewal* and *improver* crops, according to their effect on the soil. In what follows, we refer to *impoverishing*, *renewal* and *improver* as crop *types*, and we denote as C_D, C_R and C_M the sets of impoverishing, renewal and improver crops, respectively, with $C = C_D \cup C_R \cup C_M$. As stated before, the dummy fallow crop c_{fallow} belongs to the set of *renewal* crops C_R.

1.4.2 Profit of a crop and of a crop sequence

The profit obtained by assigning a crop to a plot depends on the crop itself and on the $k-1$ crops preceding it on the same plot, with $k \geq 1$ fixed. As mentioned, this is due to the fact that the yield is directly affected by the rotation and, to maintain the desired yield, additional costs occur (e.g., for phytosanitary treatments and fertilizers) when cultivating a crop in a poor rotation scheme, resulting in lower profit. More precisely, the profit associated with a crop is computed as the revenue generated by its average yield minus the production costs. These costs account for the rotation defined by the $k-1$ preceding crops, since unfavorable rotations generally require additional inputs and consequently reduce profitability. In a general definition of the crop rotation problem, k is the *rotation length*, which may vary from one geographical area to another. In the Mediterranean pedo-climatic context, $k = 3$ is commonly used, implying that the yield and the profit of a crop are *affected* by the two crops preceding it. More precisely, in a k-crop rotation scheme, the profits are computed by considering the classification of the crops in *impoverishing*, *renewal* or *improver*, and the highest profits are attained when best practice rotation schemes are performed.

Formally, given a crop sequence $\sigma = \langle c_1, c_2, \ldots, c_h \rangle$, for each crop $c \in \sigma$, let $\sigma_k(c)$ be the subsequence of σ containing c and the $k-1$ crops immediately preceding it in σ, if any, and let $t_\sigma(c)$ be the period in which c is sown. If less than $k-1$ crops precede c in σ, then $\sigma_k(c)$ will contain c and all the preceding crops. Note that, the profit of a crop sequence on a plot depends on the characteristics and size of the plot and, so, on the class H_w the plot belongs to, with $w \in \{1, \ldots, m\}$

We denote by $p_w(\sigma_k(c), t_\sigma(c))$ the profit earned by growing crop c, in period $t_\sigma(c)$, on a plot of H_w, when the (at most) $k-1$ crops immediately preceding c on the plot are those given in $\sigma_k(c)$. The total profit $\pi_w(\sigma)$ earned by cultivating the crop sequence σ on a plot of H_w is then given by

$$\pi_w(\sigma) = \sum_{c \in \sigma} p_w(\sigma_k(c), t_\sigma(c)). \tag{1.1}$$

Figure 1.2 shows the total profit and its components of a 4-crop sequence cultivated on a generic plot of class w, over four time periods, when $k = 3$. The profit of the first crop c_1 is not affected by the rotation, since it is the first crop of the sequence. On the other hand, the profit of the other crops are affected by the (at most) 2 preceding crops (since $k = 3$). Using the notation introduced above, for crop c_2, the rotation is given by the subsequence $\sigma_3(c_2) = \langle c_1, c_2 \rangle$. Hence, the profit of c_2 is $p_w(\langle c_1, c_2 \rangle, t_2)$ and it is affected by the types of c_1 and c_2. Similarly, for crop c_3 the rotation is given by the subsequence $\sigma_3(c_3) = \langle c_1, c_2, c_3 \rangle$. In this case, the profit of c_3 is $p_w(\langle c_1, c_2, c_3 \rangle, t_3)$ and it is affected by the types of c_1, c_2, and c_3. To compute the profit of c_4 we must consider the 3-crop rotation $\sigma_3(c_4) = \langle c_2, c_3, c_4 \rangle$, and the profit of c_4, $p_w(\langle c_2, c_3, c_4 \rangle, t_4)$, is affected by the types of c_2, c_3 and c_4. Note that, since $k = 3$, the profit of c_4 is not affected by the type of c_1, because the effect on the soil of cultivating c_1 in t_1 is assumed to be negligible when we are cultivating c_4. The total profit produced by the sequence $\sigma = \langle c_1, c_2, c_3, c_4 \rangle$, as stated by the formula (1.1), on a plot of class H_w is given by

$$\pi_w(\sigma) = p_w(\langle c_1 \rangle, t_1) + p_w(\langle c_1, c_2 \rangle, t_2) + p_w(\langle c_1, c_2, c_3 \rangle, t_3) + p_w(\langle c_2, c_3, c_4 \rangle, t_4).$$

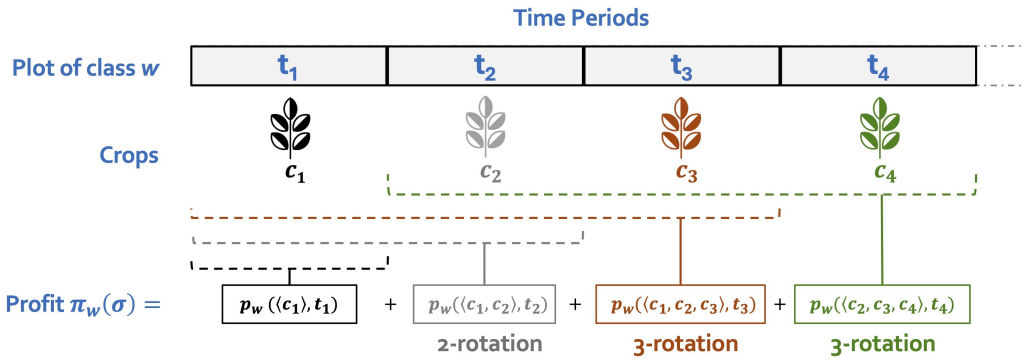

Figure 1.2: Profit of a crop sequence $\sigma = \langle c_1, c_2, c_3, c_4 \rangle$, cultivated in periods t_1, t_2, t_3 and t_4 on a generic plot of class w, with $k = 3$.

1.4.3 Agronomic and management requirements

Crops must be assigned to periods in such a way that the following general *agronomic and management* requirements are respected:

1. Limits exist on the maximum number of times crops of the same *botanical group* (hereafter, also referred to as *botanical family*) can be consecutively replanted on the same plot. These botanical groups are denoted as $B = \{B_1, B_2, \ldots, B_b\}$, and the *maximum replanting* of a botanical group B_g is denoted as $R_{max}(g)$, with $g \in \{1, 2, \ldots, b\}$. Sequences of crops of the same botanical group B_g cannot be longer than $R_{max}(g)$. Note that the maximum replanting of each crop/botanical group is a crop-specific characteristic and does not depend on the length of the planning horizon.

2. For certain crops, there are constraints on the maximum and minimum land surface that must be allocated to crop c in each period t, denoted as max_c^t and min_c^t, respectively. This means that, in each period t, it is mandatory to allocate at least min_c^t and no more than max_c^t land surface to crop c.

3. For certain crops, there are constraints on the minimum land surface, denoted as $minIF_c^t$, that must be allocated to crop c in period t if the farmer chooses to produce it during that period. While it is not mandatory to grow crop c in period t, if the farmer decides to do so, c must be cultivated on a surface of at least $minIF_c^t$. For instance, a farmer might determine in advance that cultivating tomatoes is not worthwhile unless at least ten hectares are allocated. Therefore, the farmer will either not cultivate tomatoes at all or will dedicate a minimum of ten hectares to tomato cultivation.

Note that the dummy fallow crop (c_{fallow}) is not subject to the agronomic and management requirements 1-3.

1.4.4 Problem definition

The crop rotation problem, denoted as CRP-k, can be stated as follows [10]:

> CROP ROTATION PROBLEM – CRP-k:
> *Given* a set of plots H, a set of crops C and a set of periods $T = T_o \cup T_e$:
> *Assign* first semester or annual crops to each plot in each (odd) seeding period in T_o;
> *Decide*, on each plot, whether second semester crops are assigned or not in each (even) seeding period in T_e;
> *Such that* the total profit is maximized, the duration of each annual crop is respected, and the agronomic and management requirements 1–3 of Section 1.4.3 are satisfied.

Note that, CRP-k consists in finding a *crop sequence* on each plot. Since first semester or annual crops must be assigned to each odd period, while second semester crops may or may not be planted in even periods, the length of each sequence (i.e., the number of crops in each sequence) depends on how many even periods are used for second semester crops. It is worth noting that second semester crops appear in a crop sequence only when they are cultivated after a first semester crop, as their planting is optional. Since a crop sequence represents all crops seeded on a plot throughout the planning horizon, two consecutive first semester or annual crops may occur when no second semester crop is planted in the corresponding even period. Hence, the *longest crop sequence* on a plot alternates first and second semester crops, and has length p, while the *shortest crop sequence* only assigns first semester or annual crops to each odd period and has length $p/2$, where p is the last period of the time horizon T.

As proved in [10], the Crop Rotation Problem (CRP-k) is strongly NP-hard for $k = 3$. We refer the reader to that paper for the detailed proof.

In the next section, constraints coming from public regulation (CAP) and private initiatives (CdM) are introduced.

1.5 Including sustainability constraints from public and private regulations in CRP-k

Crop planning decisions are increasingly influenced by sustainability-oriented policies and standards. In the EU, the Common Agricultural Policy (CAP) links a crucial share of farm support to compliance with environmental requirements, affecting both agronomic choices and the profitability of alternative cropping plans. In parallel, private schemes promoted by agri-food companies may reinforce similar objectives through certification or contractual conditions. In this chapter, we also consider the "Carta del Mulino" (CdM), a private initiative introduced by the Barilla Group.

The focus of this section is to provide a simplified view of these policies. Rather than detailing their institutional evolution, we translate their key requirements into model

inputs that shape constraints and incentives. This approach allows us to integrate sustainability criteria directly into CRP-k and motivates the scenario design presented in the next section.

1.5.1 Common Agricultural Policy (CAP)

Sustainable agricultural production is now an explicit objective of the CAP, while acknowledging that the adoption of environmentally oriented practices may entail higher costs or lower revenues. A central policy lever in this transition is the promotion of greater crop diversity and crop rotation, with the aim of improving soil quality, supporting biodiversity, and contributing to climate goals.

The reform agreement entered into force on January 1, 2023, introducing a greener CAP [45]. Relative to the CAP 2014–2020 programming, where greening requirements and crop diversification were largely defined at EU level under Regulation 1307/2013, the CAP 2023–2027 maintains two pillars: direct income support and measures aimed at environmental sustainability [46].

In CAP 2023–2027, compliance with Good Agricultural and Environmental Conditions (GAEC) is required to access EU funds. For arable systems, two provisions are particularly relevant for crop planning. First, GAEC 7 introduces mandatory *Crop Rotation*, representing a significant shift from earlier provisions that primarily encouraged *Crop Diversification* [47]. Second, GAEC 8 addresses the provision of non-productive areas. In parallel, the new CAP offers Eco-schemes that go beyond baseline conditionality and are voluntary for farmers. In Italy, five Eco-schemes have been approved, and this study focuses on Eco-scheme 4 and Eco-scheme 5. Eco-scheme 4 promotes extensive forage systems with multi-year rotations including legumes/forage and the inclusion of at least one protein/oilseed or renewal crop. Eco-scheme 5 introduces specific measures for pollinators through the establishment and maintenance of dedicated nectariferous/polliniferous plant cover over a minimum contiguous area [48].

During the first year of implementation, difficulties in meeting some of the new GAEC requirements contributed to widespread farmer protests across Europe in early 2024. Policy responses led to the adoption of Regulation 2024/1468, introducing additional flexibility within GAEC and Eco-schemes [49]. In particular, compliance with GAEC 7 may also be achieved through *Crop Diversification*, and GAEC 8 was made voluntary [50, 51]. Furthermore, Eco-scheme 5 was redefined into two levels combining targets for non-productive areas and pollinator-friendly cover [52].

In the following, a description of the various CAP regulations considered in this study is provided. These regulations are designed to assess the impact of existing policies that influence European farmers' arable land allocation decisions through economic incentives.

CAP 2014–2020

The farmer is involved in the greening production schemes of the Common Agricultural Policy (CAP), promoted by the European Union following the EU regulation 1307/2013. More precisely, CAP 2014–2020 regulation establishes that:

- *Crop Diversification* - Farms with more than 10 hectares of arable land have to grow at least two crops in each seeding period, while at least three crops are required on farms with more than 30 hectares of arable land. Furthermore, a crop can not cover more than 75% of the arable land, and the land assigned to two crops must not exceed 95% of the arable land.

- *Ecological Focus Area* - Farmers with arable land exceeding 15 hectares must ensure that at least 5% of their land is an *Ecological Focus Area* (EFA). Ecological focus areas may include different kind of features linked to landscapes, grasslands or improving biodiversity crops. For the farms in the case study, located in an area characterized by the production specialization of high-income arable crops, nitrogen fixing crops have been chosen to introduce EFAs. According to the above requirements, in what follows we let $C_{leg} \in C_A \cup C_F$ the set of nitrogen fixing crops that are useful to satisfy the EFA constraints.

CAP 2021–2023

The farmer participates in the production schemes of the CAP 2021-2023, implemented in accordance with EU Regulation 2021/2115. In this scenario, payments are granted based on compliance with GAECs, which are required to access basic payments, and the adoption of voluntary measures, such as Eco-schemes. This regulation establishes:

Mandatory

- *GAEC 7* - The farmer must comply with *Crop Rotation* which specify that if, on a plot, crops of the same botanical group are assigned to the two semesters of a given year, the next crop on that plot must belong to a different group.

- *GAEC 8* - At least 4% of arable land at the farm level must be devoted to non-productive areas and features, including land left fallow.

Voluntary

- *Eco-scheme 4* - Rotation over a two-year period of extensive forage systems, requiring at least one leguminous crop, forage crop, or renewal crop to be sown at the parcel level.

- *Eco-scheme 5* - Protection of pollinators through commitments to cultivate honey-attractive crops or flower strips.

The fallow crop c_{fallow} is used to satisfy the constraint GAEC 8.

CAP 2023–2027 – Diversification

The farmer is involved in the production schemes of the CAP 2023–2027 with derogations approved through the EU regulation 2024/1468. In this scenario the farmer can contribute to compliance with GAEC 7 by *Crop Diversification*, i.e. the same rules of the CAP 2014–2020. GAEC 8 has been incorporated into Eco-scheme 5, dividing it into two levels and making it a voluntary measure. Specifically, with Level 1, the farmer can voluntarily

choose to allocate 4% of his arable land to fallow areas and with Level 2 the farmer can dedicate a cover with nectariferous and pollinic crops. Eco-scheme 5 — Levels 1 and 2 are mutually exclusive; that is, even if a given plot and crop satisfy the requirements of both levels, the farmer cannot comply with both levels within the same period.

CAP 2023 – 2027 – Rotation
The farmer is involved in the production schemes of the CAP 2023–2027 with derogations approved through the EU regulation 2024/1468. In this scenario the farmer complies GAEC 7 with *Crop Rotation* and can voluntary choose to adhere to Eco-schemes, which are the same of the Diversification scenario described above.

1.5.2 Carta del Mulino (CdM)

In addition to the CAP regulations, we also consider the "Carta del Mulino" (CdM) initiative, introduced by the Barilla Group [19]. This initiative complements CAP incentives by promoting sustainable practices at the farm level. In fact, the CdM incentive is an additional premium price paid for wheat cultivated in compliance with rules involving strong agroecological principles.

In particular, the CdM initiative provides economic incentives to the farmers, if the following crop rotation constraints are respected when a CdM-designated wheat crop, denoted as c_{CdM}, is sowed on a plot. More precisely, after c_{CdM} is sown on a plot:

- *Greening constraints:* at least one legume or oilseed must be cultivated within the next four crops on the same plot.
- *Diversification constraints:* at least two different crops must appear within the next four crops on the same plot.
- *Repetition constraints:* crops belonging to a designated repetition-limited set (denoted as CR, with $c_{CdM} \in CR$) cannot be repeated more than once within the sequence of the next four consecutive crops. That is, three crops from CR cannot appear consecutively in such a sequence.

In addition, CdM incentives are received only if 3% of the land area devoted to c_{CdM} is dedicated to flower strips (useful to satisfy the CAP 2023–2027 Eco-scheme 5, too).

Table 1.1 summarizes all the notation presented so far.

Category	**Notation**	**Definition**
Farmland	H	Set of all plots
	m	Number of subsets of homogeneous plots
	H_w	Class of H of homogeneous plots, w $\in \{1, 2, \ldots, m\}$
	s_w	Size of a plot in the homogeneous class H_w
	S_{tot}	Total farmland surface
Planning horizon	T	Planning horizon, i.e., the set of seeding periods
	p	Number of seeding periods in the planning horizon
	t	Generic period in T, t $\in \{1, 2, \ldots, p\}$
	T_o	Set of odd periods, i.e., first semesters
	T_e	Set of even periods, i.e., second semesters
Crops	C	Set of all crops
	n	Number of crops
	C_A	Set of annual crops
	C_F	Set of first semester crops
	C_S	Set of second semester crops
	C_D	Set of impoverishing crops
	C_R	Set of renewal crops
	C_M	Set of improver crops
	C_{leg}	Set of nitrogen-fixing crops
	CR	Designated repetition-limited set of crops introduced by CdM constraints
	c	Generic crop in C
	c_{fallow}	Dummy fallow crop
	c_{CdM}	CdM-designated wheat crop
	d_c	Minimum duration of annual crop c
	max_c^t	Maximum land surface that can be allocated to crop c in period t
	min_c^t	Minimum land surface that must be allocated to crop c in period t
	$minIF_c^t$	Minimum land surface that must be allocated to crop c in period t, if cultivated
	B	Set of all botanical groups
	b	Number of different botanical groups
	B_g	Subset of B representing a single botanical group, g $\in \{1, 2, \ldots, b\}$
	$R_{max}(g)$	Maximum replanting of crops in botanical group $B_g, g \in \{1, 2, \ldots, b\}$
Rotations and Crop sequences	k	Rotation parameter, i.e., rotation length
	$\sigma = \langle c_1, c_2, \ldots, c_q \rangle$	Crop sequence
	$\sigma_k(c)$	Subsequence of k crops including c and the $k-1$ crops immediately preceding c
	$t_\sigma(c)$	Period in which crop c is sown in sequence σ
Profits	$p_w(\sigma_k(c), t\sigma(c))$	Profit earned by growing c in $t_\sigma(c)$ on a plot of H_w when c is the last crop of $\sigma_k(c)$
	$\pi_w(\sigma)$	Total profit earned by cultivating the crop sequence σ on a plot of subset H_w

Table 1.1: Table of notations.

1.6 A CRP-k example

In this section, a small instance of CRP-k with $k = 3$, incorporating the CAP 2014-2020 sustainability constraints, and its optimal solution is presented. The example illustrates how the CRP problem is shaped by agronomic rotation requirements and policy constraints, in particular those imposed by CAP 2014–2020. In Section 1.8, we describe the specific methodologies used to tackle this problem, so as to explain how the subsequent

results are obtained. Section 1.10 presents additional examples and insights derived from the model. The present example is included to illustrate the difficulties introduced by this problem.

In the instance, using the notation defined above, the set of plots H consists of a single class of plots H_1, containing 30 homogeneous plots, each with a size of $s_1 = 1$ hectare (ha). Hence, the total farmland area is $S_{tot} = 30$ ha.
A 5-year planning horizon is considered and the set of planning periods is $T = \{1, 2, \ldots, 10\}$, where the set of odd periods is $T_o = \{1, 3, 5, 7, 9\}$ and the set of even periods is $T_e = \{2, 4, 6, 8, 10\}$.
The set C is composed of the following four crops, $C = \{$Wheat, Soybean, Alfalfa, Sunflower$\}$. Wheat is a first semester crop, Soybean is a second semester crop, while Alfalfa and Sunflower are annual crops. Consequently, $C_F = \{$Wheat$\}$, $C_S = \{$Soybean$\}$, and $C_A = \{$Alfalfa, Sunflower$\}$, implying that Wheat can only be cultivated in periods of T_o, while Soybean can only be cultivated in periods in T_e, provided Wheat was cultivated on the same plot in the previous period. As Alfalfa and Sunflower are annual crops, when they are cultivated in a period of T_o, they must remain on the land also in T_e. Note that Wheat is classified as an impoverishing crop, Sunflower as a renewal crop, and Soybean and Alfalfa as improver crops.
The annual crops have durations of $d_{\text{Alfalfa}} = 8$ periods and $d_{\text{Sunflower}} = 2$ periods. This means that, when cultivated, Alfalfa must remain on the land for at least 4 years (8 periods), and Sunflower for at least 1 year (2 periods).

Each of the four crops belongs to a different botanical group. Consequently, the maximum replanting of each group is determined by the maximum replanting of its crop. Specifically, the maximum replanting are $R_{max}(\text{Wheat}) = 3$, $R_{max}(\text{Soybean}) = 1$, $R_{max}(\text{Alfalfa}) = 4$, and $R_{max}(\text{Sunflower}) = 3$. As an example, this means that Sunflower cannot be cultivated more than 3 consecutive times (i.e., years, being the Sunflower annual) on the same plot without being interrupted by another crop.
Then, assume that the following additional agronomic and management requirements hold: Alfalfa must be cultivated on at least 5 hectares each year, i.e., $min^t_{\text{Alfalfa}} = 5$ ha, $\forall t \in T$; Wheat cannot be cultivated on more than 15 hectares during each odd period, so $max^t_{\text{Wheat}} = 15$ ha, $\forall t \in T_o$; and Soybean cannot be cultivated on more than 9 hectares during each even period, i.e., $max^t_{\text{Soybean}} = 9$ ha, $\forall t \in T_e$. No requirements exist for Sunflower.
Regarding the CAP 2014-2020 constraints (see Section 1.5.1), since Alfalfa is the unique nitrogen-fixing crop (i.e., $C_{leg} = \{$Alfalfa$\}$), it is the only crop suitable to satisfy the *Ecological Focus Area* constraint.

Figure 1.3 reports a table illustrating the optimal solution for the CRP-3 example introduced above. In Rows 2–11 of the table the optimal crop sequences are reported. In Rows 12–16, the land use in each period is given, in terms of number of plots (i.e., hectares) assigned to each crop. As shown in Rows 2–11, the solution consists of 10 different crop sequences. Each sequence is assigned to a specific number of plots from the set H_1, reported in Column 2 of the table (denoted as "# plots"). As an example, the crop sequence 1 is assigned to exactly 4 plots. In each sequence, a "-" in even periods means that the second semester crop (i.e., Soybean) is not grown after the first semester

		Periods									
Crop sequences	**# plots**	1	2	3	4	5	6	7	8	9	10
1	4	Wheat	Soybean	Wheat	-	Sunflower	Sunflower	Wheat	Soybean	Alfalfa	Alfalfa
2	1	Wheat	Soybean	Wheat	Soybean	Wheat	-	Alfalfa	Alfalfa	Alfalfa	Alfalfa
3	2	Wheat	Soybean	Wheat	-	Sunflower	Sunflower	Wheat	Soybean	Wheat	-
4	2	Wheat	Soybean	Wheat	Soybean	Wheat	Soybean	Wheat	Soybean	Wheat	-
5	1	Wheat	-	Sunflower	Sunflower	Wheat	Soybean	Wheat	Soybean	Sunflower	Sunflower
6	5	Wheat	-	Sunflower	Sunflower	Wheat	Soybean	Wheat	-	Alfalfa	Alfalfa
7	9	Alfalfa	Alfalfa	Alfalfa	Alfalfa	Alfalfa	Alfalfa	Alfalfa	Alfalfa	Wheat	Soybean
8	1	Sunflower	Sunflower	Wheat	Soybean	Wheat	Soybean	Wheat	-	Sunflower	Sunflower
9	3	Sunflower	Sunflower	Wheat	Soybean	Wheat	-	Alfalfa	Alfalfa	Alfalfa	Alfalfa
10	2	Sunflower	Sunflower	Wheat	Soybean	Wheat	-	Sunflower	Sunflower	Wheat	-

Crop		Land Use									
Wheat		15	-	15	-	15	-	15	-	15	-
Sunflower		6	6	6	6	6	6	2	2	2	2
Soybean		-	9	-	9	-	9	-	9	-	9
Alfalfa		9	9	9	9	9	9	13	13	13	13

Figure 1.3: Optimal sequences solution for example in Section 1.6

crop (Wheat). This may happen because Soybean cannot be cultivated on more than 9 hectares during each even period. The sequences show that the agronomical constraints of the crops are satisfied: Wheat, which is a first semester crop, is sown exclusively in T_o and remains in the field for a single period before being harvested. Soybean, a second semester crop, is sown exclusively in T_e, following the preceding cultivation of Wheat, while Sunflower is sown in T_o and harvested at the end of the second period after planting. Regarding Alfalfa, since its duration is of 8 periods and recalling that we allow a crop to be harvested after the planning horizon, either it remains in the field for 8 periods after its cultivation (see crop sequence 7), or is planted in a period t for which $t + 8 - 1 > 10$ (see crop sequences 1, 2, 6 and 9). Observe that, since $R_{\max}(\text{Alfalfa}) = 4$, in sequence 7, Alfalfa is replaced by a different crop after its duration of 8 periods.
The profits of each crop sequence are computed by Formula (1.1). As stated in Section 1.4.2, for a given crop c, the profit depends by the $k - 1 = 2$ crops preceding c in the sequence, and the highest profit is attained for c when best practice rotations are performed. In fact, recalling that Wheat is impoverishing, Sunflower is renewal, and Soybean and Alfalfa are improver, in the crop sequences reported in Figure 1.3 best practice rotations are often performed, e.g., *improver-impoverishing-renewal*, *impoverishing-improver-impoverishing*, *renewal-impoverishing-improver* (for more details on best practice rotations and cost computation, see Table 1.4 and [10]).
Regarding the land use, in Rows 12–16 of Figure 1.3, the number of plots of the farmland assigned to each crop in each period of T are shown. This allows an easy checking of the agronomic and management requirements 1 and 2, and of the CAP 2014-2020 constraints. More precisely, we have that $max^t_{\text{Wheat}} = 15$ hectares are assigned to Wheat in each odd period. Similarly, Soybean is assigned to $max^t_{\text{Soybean}} = 9$ hectares in each even period, and the minimum land requirement of $min^t_{\text{alfalfa}} = 5$ hectares of Alfalfa is always respected. The *Crop Diversification* and *Ecological Focus Area* constraints of the CAP (presented in 1.5.1) are also satisfied. More precisely, since no crop is assigned to more than 75% of the arable land (i.e., 22 ha) in each odd period, the first *Crop Diversification* constraint is fulfilled. Similarly, since in each odd period no more than the 95% of the arable land (28 ha) is assigned to crops belonging to two distinct botanical families, the second *Crop Diversification* constraint is also satisfied. Finally, the *Ecological Focus Area* constraint is fulfilled in each odd period, since at least 5% of the arable land (2 ha) is devoted to the nitrogen-fixing crop Alfalfa.

1.7 Problem Scenarios

Based on the various CAP frameworks and on the private initiative "Carta del Mulino" (CdM), introduced in Sections 1.5.1 and 1.5.2, a set of alternative production scenarios are defined. These scenarios reflect different combinations of sustainability constraints and incentive structures, leading to distinct entrepreneurial behaviors and crop planning strategies.

In addition, a hypothetical scenario is introduced, referred to as the *Pure Farmer*, which represents a farmer who voluntarily does not follow both CAP regulations and CdM rules.

Table 1.2 provides an overview of the policy and management scenarios considered in this work.

#	Scenario	Description
1)	Pure Farmer	Production scheme without sustainable regulations. Decisions are based solely on income maximization, agronomic knowledge, and technical considerations. Only the agronomic and management requirements 1-3 are satisfied.
2)	CAP 2014–2020	Based on EU Regulation 1307/2013. Includes the *Crop Diversification* rule and the *Ecological Focus Area* requirement.
3)	CAP 2021–2023	Based on EU Regulation 2021/2115. Includes mandatory GAEC standards and optional Eco-schemes for additional payments.
4)	CAP 2023–2027 - Diversification	Derogation scenario under EU Regulation 2024/1468: GAEC 7 compliance achieved through *Crop Diversification*, while GAEC 8 is shifted to a voluntary Eco-scheme.
5)	CAP 2023–2027 - Rotation	Alternative derogation scenario under the same regulation, where GAEC 7 compliance is achieved through *Crop Rotation*.
6)	CAP 2014–2020 + CdM	Scenario 2 with the additional agroecological constraints introduced by the "Carta del Mulino" (CdM) initiative.
7)	CAP 2021–2023 + CdM	Scenario 3 combined with CdM constraints.
8)	CAP 2023–2027 - Diversification + CdM	Scenario 4 combined with CdM constraints.
9)	CAP 2023–2027 - Rotation + CdM	Scenario 5 combined with CdM constraints.

Table 1.2: Scenarios for CRP-k.

1.8 Solution methods for CRP-k

This section presents the methods developed to solve the Crop Rotation Problem (CRP-k). First, a graph-based representation is introduced, illustrating how the network is constructed and how it captures feasible crop sequences. This model effectively represents seasonality, crop duration, and the agronomic and management requirements expressed in Point 1 of Section 1.4.3. The graph is a multi-layer directed network in which nodes

correspond to crop subsequences and arcs model the rotation of crops in the tail and head nodes. In this model, the weight on an arc is associated with the total profit of the crops in the head node, which is also affected by the rotation given by the crops in the tail node. Therefore, the crop rotation is naturally embedded within this formulation.

Subsequently, different approaches based on the graph are presented, namely:

- **Exact methods:** these methods exploit the graph to provide the optimal solution of the problem. The first method involves an arc flow formulation (see Section 1.8.2.1), denoted as G-ILP, which is based on the graph and provides a compact formulation applicable to the different policy scenarios introduced by the CAP. Moreover, a Dynamic Programming approach has been devised for the CRP-3 case—extendable to any CRP-k—to compute the optimal schedule on a single plot (see Section 1.8.2.2), ensuring that all sequence-based constraints of the problem (both CAP and CdM) are satisfied.

- **Upper bounds and Heuristic methods:** a first method based on Lagrangian relaxation has been developed to provide bounds for CRP-k (see Section 1.8.3.1). Moreover, two heuristic algorithms have been designed to efficiently handle larger instances where exact approaches become computationally challenging. These methods are based on Column Generation and Dynamic Programming (see Section 1.8.3.2 and 1.8.3.3 respectively).

1.8.1 A graph formulation for CRP-k

Description, lemmas, and example

In this section, a graph model for CRP-k is presented. The general idea is to consider the assignments of crops to periods as a path on a graph. In the graph, the nodes represent subsequences of crops assigned to given admissible periods, and an arc between two nodes exists if the resulting crop sequence satisfies all the agronomic constraints (i.e., maximum crop replanting, crop duration, seeding periods). In other words, all the agronomical constraints are embedded in the graph. The flow on the arcs denote the number of plots following a given crop pattern.

In [10], a similar graph concept has been introduced for a special case of CRP-3. In this section, the graph model is extended to the general CRP-k and used to develop an arc flow ILP formulation for the problem. More precisely, in the graph, each node represents a subsequence of τ crops and an arc represents a subsequence of 2τ crops, where τ is properly set to take into account the constraints on the maximum replanting and the rotation parameter k (i.e., the rotation length), required to compute the profit of a crop (which depends on the $k-1$ preceding crops). To determine the number of crops τ in each node, two cases can be distinguished, where $R_{\max} = \max_{B_g \in B} R_{\max}(g)$ denotes the maximum replanting threshold among all botanical groups:

- If $k > R_{\max}$, the value of τ is defined as $\tau = k - 1$. Hence, each node represents a subsequence of $k-1$ crops, i.e., an arc represent a sequence of $2k-2$ crops. In fact, in this case, the $k-1$ crops of the tail node of the arc are sufficient to keep

track of the $k-1$ preceding crops for each crop in the head node (to both properly compute the profits and check for satisfaction of the maximum crop replanting).

- If $k \leq R_{max}$, the value of τ is defined as $\tau = R_{max}$. Hence, each node represents a subsequence of R_{max} crops, and an arc represents a sequence of $2R_{max}$ crops. In this way, for each crop c in the head node of the arc, the R_{max} preceding crops of c can be easily detected, to properly compute the profit of c (considering its $k-1$ preceding crops) and to check the compliance with the maximum replanting constraints (forbidding c to be cultivated more than R_{max} consecutive times).

In practice, setting τ as described above allows to only include in the graph nodes and arcs so that: (i) the crop sequence in any path satisfies the agronomic constraints; (ii) the profits of crops can be properly computed taking into account the crop rotations.

The graph $G = (V, A)$ used to model CRP-k is a multi-layer directed graph where V is the node set and A is the set of arcs. The node set contains one source/sink pair for each plot class H_w, denoted as so_w and de_w, respectively, for $w = 1, \ldots, m$, and a set of intermediate nodes partitioned into layers, say $L_1, \ldots, L_l$. The intermediate nodes are used to model sequences of τ crops and of the related seeding periods. More precisely, an intermediate node is denoted as $(c_1, c_2, \ldots, c_\tau)^{t_1,t_2,\ldots,t_\tau}$, where $c_1, c_2, \ldots, c_\tau$ is a crop subsequence and $t_1, t_2, \ldots, t_\tau$ are the seeding periods of $c_1, c_2, \ldots, c_\tau$, respectively, with $t_1 < t_2 < \ldots < t_\tau$. An arc between two nodes allows to keep track of the τ crops preceding each crop of the subsequence in the head node of the arc. The arcs in A only connect: (a) each source node so_w, $w = 1, 2, \ldots, m$, with all nodes in the first layer L_1; (b) the nodes of two adjacent layers L_i, L_{i+1}, for $i = 1, \ldots, l-1$; (c) a node of a layer with all the demand nodes de_w, $w = 1, 2, \ldots, m$. Regarding point (c), the last node of a path (representing a complete crop sequence from period 1 to period p) is not necessarily located in the last layer L_l. This depends on the length of the crop sequence in terms of the number of crops, which, in turn, is influenced by the number of second-semester crops within the sequence (see also the example in Figures 1.4 and 1.5).

Each arc has a minimum capacity of 0 and a maximum capacity equal to the total number of plots $|H|$. Hence, an arc between two nodes of adjacent and intermediate layers connects two nodes i and j of the form $i = (a_1, a_2, \ldots, a_\tau)^{t_1,t_2,\ldots,t_\tau}$ and $j = (b_1, b_2, \ldots, b_\tau)^{q_1,q_2,\ldots,q_\tau}$. That arc represents the assignment of the 2τ crops $a_1, a_2, \ldots, a_\tau$, $b_1, b_2, \ldots, b_\tau$ to the 2τ periods $t_1, t_2, \ldots, t_\tau, q_1, q_2, \ldots, q_\tau$, respectively.

Observe that, such an arc can only exist if $t_\tau < q_1$ and $q_1 \in \{t_\tau + 1; t_\tau + 2\}$ (since a crop must be assigned to each odd period). Moreover, the arc (i, j) only exists in A if the sequence $\langle a_1, a_2, \ldots, a_\tau, b_1, b_2, \ldots, b_\tau \rangle$ does not contain subsequences violating the maximum replanting constraints, which we can check since $\tau \geq R_{max}$ by definition.

A value p_{ij}^w is assigned to the arc (i, j) for each plot class H_w, $w = 1, \ldots, m$, equal to the profit gained by cultivating crops $b_1, b_2, \ldots, b_\tau$ on a plot of class H_w, knowing the $k-1$ crops preceding each of them. More precisely, the value p_{ij}^w is

$$p_{ij}^w = \sum_{c \in \langle b_1, b_2, \ldots, b_\tau \rangle} p_w(\sigma_k(c), t(c)). \tag{1.2}$$

The number of layers of G and the maximum number of nodes in each layer are now

established. At this aim, we first derive the length of the shortest and longest paths on G, in terms of number of nodes. As already observed in the previous section, the length of a crop sequence, in terms of number of crops, from period 1 to the last period p depends on how many second semester crops are assigned (since first semesters are always assigned to first semester or annual crops). Hence, a path only containing intermediate nodes of the form $(a_1, a_2, \ldots, a_\tau)^{t_1,t_2,\ldots,t_\tau}$ in which $a_1, a_2, \ldots, a_\tau$ are annual or first semester crops, and consequently $t_1, t_2, \ldots, t_\tau$ are consecutive odd periods, is a *shortest path* on G, in terms of nodes. Each intermediate node on such a path allows to "move" of $2\tau - 1$ periods, from period t_1 to $t_\tau = t_1 + 2\tau - 2$. On the other hand, a path only containing intermediate nodes $(a_1, a_2, \ldots, a_\tau)^{t_1,t_2,\ldots,t_\tau}$ in which $\langle a_1, a_2, \ldots, a_\tau \rangle$ alternates first and second semester crops is a *longest path* on G, in terms of nodes. Each intermediate node on such a path allows to "move" of τ periods, from period t_1 to $t_\tau = t_1 + \tau - 1$. The following lemmas establish the maximum number of layers and nodes in G.

Lemma 1.8.1. *The number of node layers in graph G is at most $\lceil p/\tau \rceil$, where p is the last period of the time horizon T.*

Proof. The proof is reported in the Appendix. □

Lemma 1.8.2. *The number of nodes contained in each Layer L_i of G is $O(n^\tau \tau^2 i^2)$ where n is the number of crops in C, for $i = 1, \ldots, \lceil p/\tau \rceil$.*

Proof. The proof is reported in the Appendix. □

Lemma 1.8.3. *The number of nodes in G is $O(n^{\tau-1} p^3)$, where n is the number of crops in C and p is the number of time periods in T.*

Proof. The proof is reported in the Appendix. □

Note that, in the Mediterranean context, since $R_{max} \leq 4$ and $k = 3$, then $\tau = 4$ and the number of nodes in G is $O(n^3 p^3)$. As an example, Figures 1.4 and 1.5 show the graph G for two different CRP-3 instances with a five-year (i.e., 10 periods) planning horizon $T = \{1, 2, \ldots, 10\}$, one class of homogeneous plots, and $R_{max} = 3$, implying that $\tau = 3$. Figure 1.4 illustrates the graph G in the case in which $C = \{c_1, c_2, c_3\}$, where $c_1, c_3 \in C_F$, $c_2 \in C_S$. Hence, c_1 and c_3 can be sown in $T_o = \{1, 3, 5, 7, 9\}$, while c_2 can be sown in $T_e = \{2, 4, 6, 8, 10\}$. In this example, the number of node layers in G is $\lceil p/\tau \rceil = 4$. Figure 1.5 shows the graph G when $C = \{c_1, c_2, c_3\}$ contains three annual or first semester crops. Hence, all crops in C can only be assigned to periods in $T_o = \{1, 3, 5, 7, 9\}$. Note that, in this case, the number of node layers in G is 2. In fact, since no second semester crop exists in C, all the paths in G are shortest paths.

Note that, since the nodes of the graph G model sequences of consecutive crops, the seeding period of each crop c within a sequence depends only on the seasonality of c (i.e., first or second semester, annual) and on the seeding period of the preceding crop. Consequently, the number of seeding periods in each agricultural year does not affect this modeling approach. As a result, the graph model remains valid even in pedo-climatic contexts with more than two seeding periods per year (in these cases the crop set C would be partitioned accordingly). However, in such contexts, the total number of periods in T increases, along with the number of nodes in G.

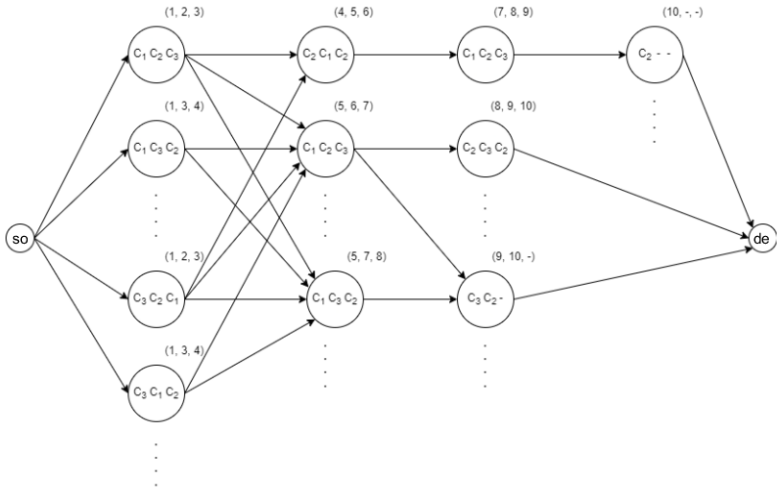

Figure 1.4: The graph G for CRP-3 with $\tau = 3$ and three crops, where c_1 and c_3 are first-semester crops, and c_2 is a second-semester crop.

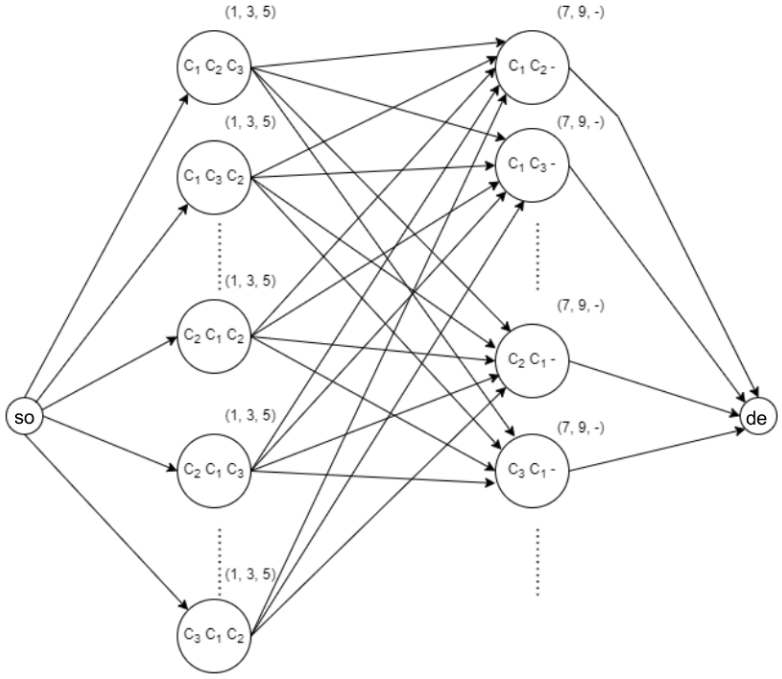

Figure 1.5: The graph G for CRP-3 with $\tau = 3$ and three annual or first-semester crops.

Flexibility of the graph based on key parameters

The graph model can be adapted to different pedo-climatic contexts, which may involve varying values of the rotation parameter k, as well as different sets of crops, resulting in

different values for the maximum replanting value among all crops, denoted as $R_{\max}$.

As previously discussed, the relationship between k and $R_{\max}$ determines the number of crops to be considered in each node, i.e., the value of τ. The following examples illustrate how the graph model remains flexible under different parameter configurations.

Case $k = 3$, $R_{\max} = 2$: Since $k > R_{\max}$, the value of τ is set to $\tau = k - 1 = 2$. In this case, each node represents a subsequence of 2 crops, while each arc corresponds to a subsequence of 4 crops.

Figure 1.6: Example of two nodes, and corresponding arc, between two intermediate layers L_i and L_{i+1} for an instance of CRP-3 and $R_{\max} = 2$.

Figure 1.6 illustrates an example of a CRP-3 instance involving two nodes and the corresponding arc connecting two intermediate layers L_i and L_{i+1}. The crop sequence $\langle c_1, c_2, c_3, c_4 \rangle$, sown in periods t_1, t_2, t_3, t_4 such that $t_1 < t_2 < t_3 < t_4$, satisfies the sequence-based constraints.

This modeling framework effectively manages both the rotation length k and the maximum replanting value $R_{\max}$, since when a new crop is added to the head node (e.g., crop c_3 in this example), the preceding $\tau = 2$ crops are considered. Thus, a total of 3 crops are evaluated at each step.

Case $k = 2$, $R_{\max} = 3$: Since $k \leq R_{\max}$, the value of τ is set to $\tau = R_{\max} = 3$. In this case, each node represents a subsequence of 3 crops, while each arc corresponds to a sequence of 6 crops.

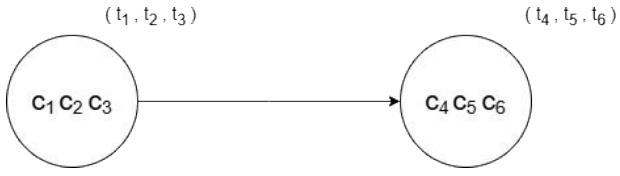

Figure 1.7: Example of two nodes and the corresponding arc between layers L_i and L_{i+1} for an instance of CRP-2 and $R_{\max} = 3$.

Figure 1.7 presents an example of a CRP-2 configuration involving two nodes and their connecting arc across intermediate layers L_i and L_{i+1}. The crop sequence $\langle c_1, c_2, c_3, c_4, c_5, c_6 \rangle$, sown during periods $t_1, t_2, t_3, t_4, t_5, t_6$ such that $t_1 < t_2 < t_3 < t_4 < t_5 < t_6$, respects the sequence-based constraints.

By setting $\tau = 3$, the model evaluates 4 crops when adding a crop in the head node. This allows it to asses the maximum replanting value $R_{\max} = 3$ and the rotation parameter $k = 2$.

Case $k = 3$, $R_{\max} = 4$: Since $k \leq R_{\max}$, the value of τ is set to $\tau = R_{\max} = 4$. In this configuration, each node represents a subsequence of 4 crops, while each arc corresponds to a sequence of 8 crops.

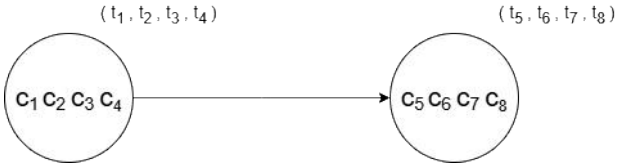

Figure 1.8: Example of two nodes and the corresponding arc between layers L_i and L_{i+1} for an instance of CRP-3 with $R_{\max} = 4$.

Figure 1.8 shows an example of a CRP-3 configuration involving two nodes and their connecting arc across intermediate layers L_i and L_{i+1}. The crop sequence $\langle c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8 \rangle$, sown during periods $t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8$ such that $t_1 < t_2 < t_3 < t_4 < t_5 < t_6 < t_7 < t_8$, complies with sequence-based constraints.

By setting $\tau = 4$, the model evaluates a sequence of 5 crops when appending a new crop to the head node. This setting enables a proper evaluation of the maximum replanting value $R_{\max} = 4$ and the rotation parameter $k = 3$.

1.8.2 Exact methods for CRP-k

1.8.2.1 An arc flow formulation (G-ILP)

Given an instance of CRP-k, let $G = (V, A)$ be the directed graph introduced above. Let $\delta^+(i)$ denote the set of arcs emanating from node i, and $\delta^-(i)$ denote the set of arcs entering node i, for all $i \in V$. Let p_{ij}^w denote the values, i.e., the profits, associated with arc $(i, j) \in A$, for each plot class H_w, as defined in (1.2). Given an intermediate node $i = (a_1, a_2, \ldots, a_\tau)^{t_1, t_2, \ldots, t_\tau} \in V$, the set of the τ crop-period pairs related to node i is denoted by $[a_1, t_1], [a_2, t_2], \ldots, [a_\tau, t_\tau]$. Abusing notation, the pair $[c, u]$ is said to belong to i if $[c, u] \in [a_1, t_1], [a_2, t_2], \ldots, [a_\tau, t_\tau]$.

Let x_{ij}^w be integer variables denoting the number of plots of class H_w that follow the (partial) crop sequence specified by arc $(i, j) \in A$ of G, with $w = 1, \ldots, m$. Let z_c^t be a binary variable equal to 1 if crop c is assigned to period t and 0 otherwise, for all $c \in C$ and $t \in T$. Recall that s_w denotes the surface of each plot in class H_w, S_{tot} denotes the total surface of the farmland. Furthermore, as previously defined, B denotes the set of botanical groups $B = \{B_1, B_2, \ldots, B_l\}$ into which the crops in $C_A \cup C_F$ are partitioned, and $C_{leg} \subseteq C_A \cup C_F$ denotes the set of nitrogen-fixing crops.

The arc flow formulation, denoted as G-ILP, is reported below for different scenarios introduced in 1.7.

Scenario "Pure Farmer"

In this scenario, the farmer operates independently of any production scheme. Consequently, the only constraints applied are the agronomic and management requirements of Section 1.4.3, which include: maximum replanting constraint, duration constraint, which are embedded in the graph, and minimum and maximum crop constraints (defined by the min_c^t, max_c^t and $minIF_c^t$ parameters).

$$\max \sum_{w=1}^{m} \sum_{(i,j)\in A} p_{ij}^w x_{ij}^w \tag{1.3}$$

$$\sum_{(i,j)\in\delta^+(i)} x_{ij}^w - \sum_{(j,i)\in\delta^-(i)} x_{j,i}^w = \begin{cases} |H_w|, \text{ if } i = so_w \\ -|H_w|, \text{ if } i = de_w \\ 0, \text{ if } i \neq so_w,\ i \neq de_w \end{cases} \quad \forall i \in V, w \in \{1,\ldots,m\} \tag{1.4}$$

$$\sum_{w=1}^{m} s_w \sum_{(i,j)\in A:[c,t]\in j} x_{ij}^w \geq min_c^t \qquad \forall c \in C \setminus c_{fallow},\ \forall t \in T \tag{1.5}$$

$$\sum_{w=1}^{m} s_w \sum_{(i,j)\in A:[c,t]\in j} x_{ij}^w \leq max_c^t \qquad \forall c \in C \setminus c_{fallow},\ \forall t \in T \tag{1.6}$$

$$\sum_{w=1}^{m} s_w \sum_{(i,j)\in A:[c,t]\in j} x_{ij}^w \geq minIF_c^t z_c^t \qquad \forall c \in C \setminus c_{fallow},\ \forall t \in T \tag{1.7}$$

$$\sum_{w=1}^{m} s_w \sum_{(i,j)\in A:[c,t]\in j} x_{ij}^w \leq S_{tot} z_c^t \qquad \forall c \in C \setminus c_{fallow},\ \forall t \in T \tag{1.8}$$

$$x_{ij}^w \in \{0,1,2,\ldots,|H_w|\} \qquad \forall (i,j) \in A, \quad \forall w \in \{1,\ldots,m\} \tag{1.9}$$

$$z_c^t \in \{0,1\} \qquad \forall c \in C,\ t \in T \tag{1.10}$$

The objective function (1.3), for this scenario, corresponds to profit maximization over the planning horizon T. Constraints (1.4) state that, for each plot class, the flow entering a node must be equal to the flow exiting the node, except at the source and demand nodes. Specifically, the flow must be $|H_w|$ at the source node and $-|H_w|$ at the demand node, where $|H_w|$ is the number of plots in class H_w. Constraints (1.5) and (1.6) define the minimum and maximum land surfaces that must be allocated to crop c during each period t, denoted as min_c^t and max_c^t, respectively. This implies that, in each period t, it is mandatory to allocate at least min_c^t and no more than max_c^t of land surface to crop c. In contrast to (1.5), Constraints (1.7) do not require crop c to be cultivated during period t. However, if the crop c is cultivated, it must be grown on a land surface of at least $minIF_c^t$. Constraints (1.8) concern the fact that, at each time period, a crop cannot be allocated to a surface greater than the total land surface S_{tot}. It is a logic constraint, forcing variables $x_{i,j}^w$ that contain sequences in which the crop c is assigned to t to be 0 when z_c^t is 0.

In the subsequent Scenarios, Constraints (1.4)–(1.8) and variables definition (1.9)–(1.10) are part of the problem definition but are omitted from the formulations to avoid repetition.

Scenario "CAP 2014–2020"

The regulations defined by the CAP 2014–2020 introduce two types of constraints: *Crop Diversification* and *Ecological Focus Areas*.

These regulations are incorporated into the G-ILP model as follows.

$$\sum_{w=1}^{m} s_w \sum_{(i,j)\in A:[c,t]\in j \ c\in C_{leg}} x_{ij}^w \geq 0.05 S_{tot} \qquad \forall\, t \in T_o \tag{1.11}$$

$$\sum_{w=1}^{m} s_w \sum_{(i,j)\in A:[c,t]\in j \ c\in B_g} x_{ij}^w \leq 0.75 S_{tot} \qquad \forall g \in \{1,\ldots,l\},\ \forall t \in T_o \tag{1.12}$$

$$\sum_{w=1}^{m} s_w \sum_{(i,j)\in A:[c,t]\in j \ c\in B_f \cup B_g} x_{ij}^w \leq 0.95 S_{tot} \qquad \forall f,g \in \{1,\ldots,l\},\ f \neq g\,, \forall t \in T_o \tag{1.13}$$

Constraints (1.11) set the minimum land size to assign to (nitrogen-fixing) legume crops to 5% of the total farmland in each period. Constraints (1.12) impose that crops of the same botanical family can be assigned to at most 75% of the farmland in each period. Constraints (1.13) state that crops of two different botanical families can be assigned to at most 95% of the land in each period. As previously stated, the constraints on maximum and minimum land surface, i.e., the same constraints (1.5)–(1.8) introduced above, as well as the variable definitions, are omitted here to avoid repetition. Moreover, the objective function of this Scenario maximizes total profit as in (1.3).

As shown in Section 1.9.2, G-ILP is able to find optimal solutions on real-world instances with small computational effort. However, CRP-k is stronlgy NP-hard in general [10], and the size of the graph and, consequently, the number of variables in G-ILP increase significantly as the size of the instances increases (in terms of number of crops, periods and maximum replanting).

Scenario "CAP 2021-2023"

In this scenario, which was the first proposal of the new CAP regulation, a set of mandatory and voluntary rules are outlined. Farmers receive payments based on compliance with mandatory GAEC (Good Agricultural and Environmental Conditions), and with the adoption of voluntary measures called Eco-schemes.

GAEC 7 – *Crop Rotation* requires a change in the botanical group every year. This constraint can be incorporated into the graph, as when a crop is sown at a given time period, the crop that precedes it can be easily identified. This capability enables the constraint to be addressed by including nodes and arcs in G that only represent crop sequences respecting the GAEC 7 requirement.

To account for Eco-scheme 4, introduced in 1.5.1, we define $E_4 = \{e_1, e_2, \ldots, e_{p/2}\}$ as the set of all possible two-year time intervals starting in an odd period, except for the last element, $e_{p/2}$, of E_4 which refers to the last two periods of the time horizon, i.e., $e_{p/2} = \{p-1, p\}$. For example, the first two-year interval $e_1 \in E_4$, represents the first four seeding periods, i.e., $e_1 = \{1, 2, 3, 4\}$, corresponding to the four semesters of the first

two years of the planning horizon. We define the new variables:

$$y^e_{E4} = \begin{cases} 1 & \text{if a crop } c \in C_{E4} \text{ is sown in the two-year interval } e \in E_4 \\ 0 & \text{otherwise} \end{cases} \tag{1.14}$$

where C_{E4} is the set of crops eligible to get the Eco-scheme 4 reward. Furthermore, we define real variables, p^e_{E4}, denoting the reward obtained when the Eco-scheme 4 is applied at the interval $e \in E_4$. In the constraints, the value $p_{u_{E4}}$ represents the unit profit per hectare prescribed by Eco-scheme 4.

Instead, to account for Eco-scheme 5, we introduce $C_{flower} \subseteq C_A \cup C_F$ as the set of crops eligible to get the Eco-scheme 5 reward, along with y^t_{L2} and p^t_{L2}. Hence, let:

$$y^t_{L2} = \begin{cases} 1 & \text{if the total area assigned to crops in } C_{flower} \text{ at time } t \in T_o \text{ is } \geq S_{rect} \\ 0 & \text{otherwise} \end{cases} \tag{1.15}$$

and let p^t_{L2} be a real variable denoting the profit obtained at period $t \in T_o$ from Eco-scheme 5. In the constraints, the value $p_{u_{L2}}$ is the unit reward per hectare prescribed by Eco-scheme 5, and S_{rect} is the minimum area that must be devoted to flower crops $c \in C_{flower}$ to get the reward ($S_{rect} = 0.25$ hectares in Eco-scheme 5).

Finally, the G-ILP model reads as follow:

$$\max \sum_{w=1}^{m} \sum_{(i,j)\in A} p^w_{ij} x^w_{ij} + \sum_{t\in T_o} p^t_{L2} + \sum_{e\in E4} p^e_{E4} \tag{1.16}$$

$$\sum_{w=1}^{m} s_w \sum_{(i,j)\in A:[c_{fallow},t]\in j} x^w_{ij} \geq 0.04 S_{tot} \qquad \forall t \in T_o \tag{1.17}$$

$$y^e_{E4} \leq \sum_{w=1}^{m} s_w \sum_{(i,j)\in A:[c,t]\in j \ c\in C_{E4} \ t\in e} x^w_{ij} \qquad \forall e \in E4 \tag{1.18}$$

$$y^e_{E4} \leq 1 - y^{(e-1)}_{E4} \qquad \forall e \in E4 \setminus \{e_1\} \tag{1.19}$$

$$p^e_{E4} \leq p_{u_{E4}} \sum_{w=1}^{m} s_w \sum_{(i,j)\in A:[c,t]\in j \ c\in C_{E4} \ t\in e} x^w_{ij} + M(1 - y^e_{E4}) \qquad \forall e \in E4 \tag{1.20}$$

$$p^e_{E4} \leq M y^e_{E4} \qquad \forall e \in E4 \tag{1.21}$$

$$\sum_{w=1}^{m} s_w \sum_{(i,j)\in A:[c,t]\in j \ c\in C_{flower}} x^w_{ij} \geq S_{rect} y^t_{L2} \qquad \forall t \in T_o \tag{1.22}$$

$$p^t_{L2} \leq p_{u_{L2}} \sum_{w=1}^{m} s_w \sum_{(i,j)\in A:[c,t]\in j \ c\in C_{flower}} x^w_{ij} + M(1 - y^t_{L2}) \qquad \forall t \in T_o \tag{1.23}$$

$$p^t_{L2} \leq M y^t_{L2} \qquad \forall t \in T_o \tag{1.24}$$

Constraints (1.17) model GAEC 8, which is mandatory. Specifically, it requires that at least 4% of the arable land must be allocated to fallow crop c_{fallow}. Constraints (1.18)–(1.21) refer to Eco-scheme 4, which is a voluntary measure.

Constraint (1.18) forces variable y^e_{E4} to be 1 when at least one crop $c \in C_{E4}$ is sown during the two-year period $e \in E_4$. Constraint (1.19) prevents the overlapping of Eco-schemes. For example, if a reward is obtained in the interval $e_1 = \{1, 2, 3, 4\}$, it can not also be obtained in $e_2 = \{3, 4, 5, 6\}$. This ensures that rewards are not duplicated at the overlapping time periods $\{3, 4\}$.

Constraints (1.20) and (1.21) model the requirements of Eco-scheme 4. These constraints calculate the reward value based on the hectares allocated to crops eligible for the reward whenever the variable y^e_{E4} is 1.

In Constraints (1.20) and (1.21), M is a constant sufficiently large to ensure the validity of the constraints when the binary variables y^e_{E4} are both 0 and 1.

Constraints (1.22)–(1.24) refer to Eco-scheme 5. In particular, Constraints (1.22) force y^t_{L2} to be 1 if a crop in C_{flower} is sown for at least S_{rect}. Meanwhile, Constraints (1.23)–(1.24) set the right value of p^t_{L2}.

The constraints on the maximum and minimum land surface (Constraints (1.5)–(1.8)) are included but omitted here and in the following scenarios to avoid repetition.

The objective of this scenario (1.16) is to maximize profit over the planning horizon, too. The key difference compared to (1.3) is that the objective function now includes the variables accounting for the rewards of Eco-schemes 4 and 5.

Scenario "CAP 2023–2027 - Diversification"

This section presents the arc flow model constraints related to the "CAP 2023–2027 – Diversification" Scenario, which includes both mandatory and voluntary rules.

The mandatory rules pertain to *Crop Diversification* are modeled by the Constraints (1.25) and (1.26) reported below. The remaining constraints (1.27)–(1.38) correspond to the voluntary rules, prescribed by the Eco-schemes 4 and 5.

$$\sum_{w=1}^{m} s_w \sum_{(i,j)\in A:[c,t]\in j,\, c\in B_g} x^w_{ij} \leq 0.75 S_{tot} \quad \forall g \in \{1,\dots,l\},\, \forall t \in T_o \tag{1.25}$$

$$\sum_{w=1}^{m} s_w \sum_{(i,j)\in A:[c,t]\in j,\, c\in B_f \cup B_g} x^w_{ij} \leq 0.95 S_{tot} \quad \forall f,g \in \{1,\dots,l\},\, f \neq g,\, \forall t \in T_o \tag{1.26}$$

$$\sum_{w=1}^{m} s_w \sum_{(i,j)\in A:[c,t]\in j\ c\in\{c_{fallow}\}\cup C_{flower}} x^w_{ij} \geq 0.04 S_{tot} y^t_{L1} \qquad \forall t \in T_o \tag{1.27}$$

$$p^t_{L1} \leq p_{u_{L1}} 0.04 S_{tot} + M(1 - y^t_{L1}) \qquad \forall t \in T_o \tag{1.28}$$

$$p^t_{L1} \leq M y^t_{L1} \qquad \forall t \in T_o \tag{1.29}$$

$$\sum_{w=1}^{m} s_w \sum_{(i,j)\in A:[c,t]\in j\ c\in C_{flower}} x^w_{ij} \geq S_{rect} y^t_{L2} \qquad \forall t \in T_o \tag{1.30}$$

$$p^t_{L2} \leq p_{u_{L2}} \sum_{w=1}^{m} s_w \sum_{(i,j)\in A:[c,t]\in j\ c\in C_{flower}} x^w_{ij} + M(1 - y^t_{L2}) \qquad \forall t \in T_o \tag{1.31}$$

$$p^t_{L2} \leq M y^t_{L2} \qquad \forall t \in T_o \tag{1.32}$$

$$y^e_{E4} \leq \sum_{w=1}^{m} s_w \sum_{(i,j)\in A:[c,t]\in j \ c\in C_{E4} \ t\in e} x^w_{ij} \qquad \forall e \in E4 \tag{1.33}$$

$$y^e_{E4} \leq 1 - y^{(e-1)}_{E4} \qquad \forall e \in E4 \setminus \{e_1\} \tag{1.34}$$

$$p^e_{E4} \leq p_{u_{E4}} \sum_{w=1}^{m} s_w \sum_{(i,j)\in A:[c,t]\in j \ c\in C_{E4} \ t\in e} x^w_{ij} + M(1 - y^e_{E4}) \qquad \forall e \in E4 \tag{1.35}$$

$$p^e_{E4} \leq M y^e_{E4} \qquad \forall e \in E4 \tag{1.36}$$

$$y^{t_3}_{L1} + y^e_{E4} \leq 1 \qquad \forall e = \{t_1, t_2, t_3, t_4\} \in E4 \tag{1.37}$$

$$y^{t_1}_{L1} + y^e_{E4} \leq 1 \qquad \forall e = \{t_1, t_2, t_3, t_4\} \in E4 \tag{1.38}$$

As explained in Section 1.5.1, the CAP 2023-2027 with derogations redefines the Eco-scheme 5 and divides it into two levels: Level 1 and Level 2. The requirements of Level 2 are modeled by the Constraints (1.30)–(1.32) which are exactly Constraints (1.22)–(1.24) introduced in the previous formulation. The prescriptions of Level 1 are modeled in a similar way, by Constraints (1.27)–(1.29). To this purpose, a new set of variables are introduced:

$$y^t_{L1} = \begin{cases} 1 & \text{if the total area assigned to } c_{fallow} \text{ and crops in } C_{flower} \\ & \quad \text{at time } t \in T_o \text{ is at least } 0.04 S_{tot} \\ 0 & \text{otherwise} \end{cases} \tag{1.39}$$

Furthermore, new real variables p^t_{L1} are introduced, representing the profit generated at period $t \in T_o$ from Level 1 of Eco-scheme 5. In Constraints (1.27)–(1.29), we denote $p_{u_{L1}}$ as the unit reward per hectare provided by Level 1.

Constraints (1.33)–(1.36) refer to Eco-scheme 4 and are identical to (1.18)–(1.21) already introduced in the previous formulation.

Finally, Constraints (1.37) and (1.38) impose that, in each period, it is not possible to simultaneously get the rewards of both Eco-scheme 4 and Eco-scheme 5 - Level 1.

In this scenario, the objective, reported in (1.40), is to maximize the profit over the planning horizon. It includes the contribution of crop profits, Eco-scheme 4, Eco-scheme 5 - Level 1 and Eco-scheme 5 - Level 2:

$$\max \sum_{w=1}^{m} \sum_{(i,j)\in A} p^w_{ij} x^w_{ij} + \sum_{t\in T_o} (p^t_{L1} + p^t_{L2}) + \sum_{e\in E4} p^e_{E4}. \tag{1.40}$$

Scenario "CAP 2023–2027 - Rotation"

In this scenario, a farmer complies with GAEC 7 through the *Crop Rotation* rules, described in Section 1.5.1 and embedded in the graph G. Since the Eco-schemes 4 and 5 remain unchanged, they are not restated here, and we refer to the formulas in (1.27)–(1.38).

The objective function is the same of the previous scenario and is given in (1.40).

Embedding the "Carta del Mulino (CdM)" constraints in the graph

In this section, an arc-flow formulation based on a graph (referred to as *G-ILP*) is presented for CRP-k. The idea is to represent crop assignments over the planning horizon as paths in a directed graph. In this graph, nodes correspond to subsequences of crops assigned to valid periods, and an arc between two nodes exists if the resulting crop sequence satisfies all agronomic constraints (e.g., maximum replanting constraint, crop duration, seeding periods). In this way, all constraints that are sequence-based can be embedded directly into the graph. For this reason, the agronomic and management requirement 1 (see Section 1.4) is modeled by the graph, along with the GAEC 7 *Crop Rotation* constraints. Moreover, the CdM constraints, i.e. *Greening, Diversification* and *Repetition constraints* (see Section 1.5.2), are also sequence-based and they can be incorporated into the graph, too.
In the graph, the cost of each arc represents the cumulative profit of the crop sequence in the head node, which is also affected by the crops in the tail node, due to the crop rotation. In order to properly compute the profit on each arc, and to model all sequence-based constraints into the graph, each node must represent a subsequence of τ crops, with $\tau = \max\{4, k-1, R\}$, where $R = max_g R_{max}(g)$, $g \in \{1, 2, \ldots, b\}$ and 4 represents the minimum value of τ to check the compliance with CdM constraints (see Section 1.8.1 for the graph flexibility).

1.8.2.2 A Dynamic Programming algorithm for CRP-k on a single plot

In this section, a Dynamic Programming approach to polynomially solve the case of CRP-k on a single plot (i.e. $|H| = 1$) is proposed. In CRP-k with a single plot (and p periods), the problem essentially consists of finding a single crop sequence, $\sigma = \langle c_{i_1}, c_{i_2}, \ldots, c_{i_q} \rangle$, with $q \leq p$, where: (i) an annual or first semester crop is assigned to each odd period; (ii) the duration of each annual crop is satisfied; (iii) second semester crops are possibly assigned to even periods; (iv) the constraints on the maximum replanting of each assigned crop are respected (i.e., the agronomic and management requirement 1 of Section 1.4.3).

Note that, the agronomic and management requirements 2 and 3 of Section 1.4.3, the CAP *Crop Diversification*, EFA and Eco-scheme 5 of Section 1.5.1 do not apply when a single plot exists, since they impose crop diversification over multiple plots.

For the sake of simplicity, in the following, problem CRP-k with $k = 3$ and one single plot is considered, and a Dynamic Programming solution algorithm is presented. Morever, an extension of the formulas to CRP-k, taking into account Eco-scheme 4, GAEC 7 *Crop Rotation*, and CdM constraints, is fairly straightforward and is discussed in this section.

A dynamic programming algorithm for CRP-3 with one plot

In the proposed dynamic programming algorithm, let $P(q)$ be the optimal total profit for CRP-3 with one plot restricted to the first q seeding periods. Furthermore, let $\psi(q, c_1, c_2, r)$ be the optimal solution value of the problem restricted to the first q periods, when the last two crops in the optimal sequence σ are c_1, c_2, in this order, and the current replanting of c_2 is r, i.e., the $r-1$ crops immediately preceding c_2 in σ belong to the

same botanical group of c_2. Then, by definitions of $P(q)$ and $\psi(q, c_1, c_2, r)$, we have:

$$P(q) = \max_{c_1, c_2 \in C;\ r=1,\ldots,R(c_2)} \psi(q, c_1, c_2, r). \tag{1.41}$$

The following recursive formulas hold for $\psi(q, c_1, c_2, r)$, depending on the period q, the types of c_1 and c_2, and the value of r. If q is an even period and c_2 is a first semester or annual crop we simply have

$$\psi(q, c_1, c_2, r) = \psi(q-1, c_1, c_2, r), \tag{1.42}$$

since c_2 can be assigned to odd periods only. For any q and r, if c_1 is an annual crop and c_2 is a second semester crop then

$$\psi(q, c_1, c_2, r) = -\infty, \tag{1.43}$$

because, by definition, an annual crop can be only followed by first semester or annual crops. Furthermore, if q is an odd period and c_2 is a second semester crop we have

$$\psi(q, c_1, c_2, r) = -\infty. \tag{1.44}$$

In fact, since a crop must be assigned to each odd period, then the last crop c_2 in the sequence must be necessarily assigned to period q, but this is not possible if c_2 is a second semester crop.

On the other hand, if q is an even period and c_2 is a second semester crop, or if q is an odd period and c_2 is a first semester or annual crop, then the following recursive formulas (1.45)–(1.47) hold, depending on whether c_1 and c_2 belong to the same replanting group or not. In the formulas below, $B(c_1)$ and $B(c_2)$ denote the replanting groups of c_1 and c_2, respectively, and $p_w(\langle c, c_1, c_2\rangle, q)$, as introduced in Section 1.4, is the profit earned by growing crop c_2 in period q when the 2 crops immediately preceding c_2 are c_1 and c. In what follows, we omit the plot class index w in $p_w(\langle c, c_1, c_2\rangle, q)$, as we are considering the single-plot case.

$$\psi(q, c_1, c_2, r) = -\infty \quad \text{if } 1 < r \leq R(c_2) \quad \text{and} \quad B(c_1) \neq B(c_2) \tag{1.45}$$

$$\psi(q, c_1, c_2, 1) = \max_{c \in C,\ r \in R(c_1)} \{\psi(q-1, c, c_1, r) + p(\langle c, c_1, c_2\rangle, q)\} \quad \text{if } B(c_1) \neq B(c_2) \tag{1.46}$$

$$\psi(q, c_1, c_2, r) = \max_{c \in C} \{\psi(q-1, c, c_1, r-1) + p(\langle c, c_1, c_2\rangle, q)\} \quad \text{if } B(c_1) = B(c_2) \text{ and } r > 1 \tag{1.47}$$

Note that, relation (1.45) holds since the replanting value of c_2 must be exactly 1 when c_1 and c_2 belong to different botanical groups and c_1 immediately precedes c_2.

The values $\psi(q, c_1, c_2, r)$ can be initialized as follows. When $q = 1$, σ contains a single crop, and we denote $\psi(1, -, c, 1) = p(\langle c\rangle, 1)\ \ \forall c \in C_F \cup C_A$, while $\psi(1, -, c, 1) = -\infty\ \ \forall c \in C_S$, since a second semester crop cannot be assigned to the first (odd) period. Then, for $q = 2, 3$ and for all $c_1, c_2 \in C$, if $R(c_2) > 1$ we have:

$$\psi(2, c_1, c_2, 2) = p(\langle c_1\rangle, 1) + p(\langle c_1, c_2\rangle, 2)\ \forall c_1 \in C_F, c_2 \in C_S \quad \text{if } B(c_1) = B(c_2) \tag{1.48}$$

$$\psi(3, c_1, c_2, 2) = p(\langle c_1 \rangle, 1) + p(\langle c_1, c_2 \rangle, 3) \ \forall c_1, c_2 \in C_F \cup C_A \quad \text{if } B(c_1) = B(c_2). \quad (1.49)$$

While, for $r = 1$, if $B(c_1) \neq B(c_2)$ and $R(c_2) = 1$ we have:

$$\psi(2, c_1, c_2, 1) = p(\langle c_1 \rangle, 1) + p(\langle c_1, c_2 \rangle, 2) \ \forall c_1 \in C_F, c_2 \in C_S \quad (1.50)$$

$$\psi(3, c_1, c_2, 1) = p(\langle c_1 \rangle, 1) + p(\langle c_1, c_2 \rangle, 3) \ \forall c_1, c_2 \in C_F \cup C_A, \quad (1.51)$$

while all other possibilities for c_1, c_2 and r leads to $\psi(q, c_1, c_2, r) = -\infty$, when $q = 2, 3$.

Example of Dynamic Programming on CRP-3

As an example, let us consider a single-plot CRP-3 instance with a planning horizon of $p = 6$ periods and two first semester crops, $C = \{a_1, a_2\}$, belonging to different botanical groups, and each with a maximum replanting of 2. Note that, in a feasible sequence only the odd periods are cultivated, i.e., $P(q) = P(q-1)$ and $\psi(q, c_1, c_2, r) = \psi(q-1, c_1, c_2, r)$, for $q \in \{2, 4, 6\}$. In Table 1.3,(a), in Columns 2 and 3, the data of the instance are reported, i.e., the profit $p_w(\sigma_k(c), t_\sigma(c))$ of the last crop in each sequence σ (reported in Column 1) of at most 3 crops, and the total profit of the sequence $\pi_w(\sigma)$ (computed as in (1.1)). The profits $p_w(\sigma_k(c), t_\sigma(c))$ depend on the types, i.e., improver, renewal, impoverishing, of the crops a_1 and a_2. In Table 1.3,(b), the initialization values, $\psi(1, -, c, r)$ and $\psi(3, c_1, c_2, r)$, computed as in relations (1.48)–(1.51) are given.

σ	$p_w(\sigma_k(c), t_\sigma(c))$	$\pi_w(\sigma)$
$\langle a_1 \rangle$	6	6
$\langle a_2 \rangle$	8	8
$\langle a_1, a_1 \rangle$	7	13
$\langle a_1, a_2 \rangle$	8	14
$\langle a_2, a_1 \rangle$	6	14
$\langle a_2, a_2 \rangle$	4	12
$\langle a_1, a_1, a_2 \rangle$	5	18
$\langle a_1, a_2, a_1 \rangle$	10	24
$\langle a_1, a_2, a_2 \rangle$	4	18
$\langle a_2, a_1, a_1 \rangle$	7	21
$\langle a_2, a_1, a_2 \rangle$	9	23
$\langle a_2, a_2, a_1 \rangle$	6	18

(a)

Initialization	
$\psi(1, -, a_1, 1)$	6
$\psi(1, -, a_2, 1)$	8
$\psi(3, a_1, a_1, 2)$	13
$\psi(3, a_1, a_2, 1)$	14
$\psi(3, a_2, a_1, 1)$	14
$\psi(3, a_2, a_2, 2)$	12

(b)

Table 1.3: (*a*) Data of the single-plot instance; (*b*) Initialization values of the dynamic programming algorithm.

By the dynamic programming formulas, we have:

$$\psi(5, a_1, a_2, 1) = \max_{a \in C,\, r \in R(a_1)} \{\psi(3, a, a_1, r) + p(\langle a, a_1, a_2 \rangle, 5)\} =$$
$$\max\{\psi(3, a_1, a_1, 2) + p(\langle a_1, a_1, a_2 \rangle, 5);$$
$$\psi(3, a_2, a_1, 1) + p(\langle a_2, a_1, a_2 \rangle, 5)\}$$

$$= \max\{13 + 5;\ 14 + 9\} = 23$$

$$\psi(5, a_2, a_1, 1) = \max_{a \in C,\ r \in R(a_2)} \{\psi(3, a, a_2, r) + p(\langle a, a_2, a_1 \rangle, 5)\} =$$
$$\max\{\psi(3, a_1, a_2, 1) + p(\langle a_1, a_2, a_1 \rangle, 5),$$
$$\psi(3, a_2, a_2, 2) + p(\langle a_2, a_2, a_1 \rangle, 5)\}$$
$$= \max\{14 + 10;\ 12 + 6\} = 24$$

$$\psi(5, a_1, a_1, 2) = \max_{a \in C}\{\psi(3, a, a_1, r - 1) + p(\langle a, a_1, a_1 \rangle, 5)\} =$$
$$\max\{\psi(3, a_1, a_1, 1) + p(\langle a_1, a_1, a_1 \rangle, 5);$$
$$\psi(3, a_2, a_1, 1) + p(\langle a_2, a_1, a_2 \rangle, 5)\}$$
$$= \max\{-\infty;\ 14 + 7\} = 21$$

$$\psi(5, a_2, a_2, 2) = \max_{a \in C}\{\psi(3, a, a_2, r - 1) + p(\langle a, a_2, a_2 \rangle, 5)\} =$$
$$\max\{\psi(3, a_1, a_2, 1) + p(\langle a_1, a_2, a_2 \rangle, 5);$$
$$\psi(3, a_2, a_2, 1) + p(\langle a_2, a_2, a_2 \rangle, 5)\}$$
$$= \max\{14 + 4;\ -\infty\} = 18$$

where the last two equality hold since the maximum replanting of a_1 and a_2 is two. Hence, we have

$$P(6) = P(5) = \max_{c_1, c_2 \in \{a_1, a_2\};\ r=1,2} \psi(5, c_1, c_2, r)$$
$$= \max\{\psi(5, a_1, a_2, 1);\ \psi(5, a_2, a_1, 1);\ \psi(5, a_1, a_1, 2);\ \psi(5, a_2, a_2, 2)\}$$
$$= \psi(5, a_2, a_1, 1) = 24.$$

Since,

$$\psi(5, a_2, a_1, 1) = \psi(3, a_1, a_2, 1) + p(\langle a_1, a_2, a_1 \rangle, 5)$$

(by the recursion above), the optimal sequence for the single-plot problem returned by the dynamic programming algorithm is $\sigma = \langle a_1, a_2, a_1 \rangle$, and the seeding periods of the three crops are $1, 3, 5$, respectively. Obviously, for longer time horizons $p > 6$, the above formulas can be repeated to compute $\psi(q, c_1, c_2, r)$, for $q = 7, \ldots, p$.

Complexity of the dynamic programming algorithm on CRP-3

Recall that, a feasible solution of CRP-k on a single plot consists of a single crop sequence σ, containing at least $p/2$ and at most p crops (one annual or first semester crop in each odd period and at most $p/2$ second semester crops). Hence, the computation of $\psi(q, c_1, c_2, r)$, for $q = 1, 2, 3$, requires $O(n^2 p)$ operations, where n is the number of crops and p is the number of seeding periods.

For each period $q \geq 3$, $P(q)$ (see Formula (1.41)) is obtained by computing the values $\psi(q, c_1, c_2, r)$, for all $c_1, c_2 \in C$ and $r \leq R_{max}$, i.e., $O(n^2 p)$ values, since $R_{max} \leq p$. By formulas (1.42)–(1.47), the computation of $\psi(q, c_1, c_2, r)$ requires at most $nR_{max} \leq np$ operations, where n is the number of crops (see (1.46)). Hence, the computations of $P(q)$ requires $O(n^3 p^2)$ operations, and, then, the optimal solution of CRP-3 on a single plot can be computed in $O(n^3 p^3)$.

The recursive formulas presented in this section can be easily generalized for CRP-k with a single plot, by defining $\psi(q, c_1, c_2, \ldots, c_{k-1}, r)$ as the optimal solution value of the problem restricted to the first q periods, when the last $k-1$ crops in the sequence σ are $c_1, c_2, \ldots, c_{k-1}$, in this order, and the replanting of c_{k-1} is r. Then, arguments similar to those used for CRP-3 can be applied to show that CRP-k on a single plot can be solved in $O(n^k p^3)$ time.

An extended dynamic programming algorithm for CRP-k with CAP 2023–2027 and CdM constraints

This section presents an extension of the Dynamic Programming algorithm for the single-plot case (i.e., $|H| = 1$), previously introduced in Section 1.8.2.2. The proposed extension integrates the framework defined by the CAP 2023–2027 and the sustainability requirements introduced by the Carta del Mulino (CdM) initiative (see Sections 1.5.1 and 1.5.2, respectively).

In particular, the extended Dynamic Programming algorithm incorporates:

- the GAEC 7 *Crop Rotation* rule from CAP 2023–2027;
- the computation of the additional reward associated with Eco-scheme 4;
- all agronomic and sustainability constraints defined by CdM, i.e. Greening, Diversification, and Repetition constraints.

To achieve this, a generalized version of the algorithm described in Section 1.8.2.2 is developed. The Dynamic Programming formulas 1.42–1.47 are extended to keeps track of the last 4 crops, thus enabling the algorithm to account for the additional constraints, particularly those introduced by CdM. Finally, Section 1.8.3.3 illustrates how this extended Dynamic Programming algorithm is used to generate a set of feasible crop sequences, which are then employed to construct a graph-based model providing feasible solutions for CRP-k.

As already stated in this section, in CRP-k with a single plot (and p periods), the problem consists in finding a single crop sequence, $\sigma = \langle c_1, c_2, \ldots, c_q \rangle$, with $q \leq p$, where all the sequence-based constraints are satisfied. More precisely, for this extension: (*i*) an annual or first semester crop is assigned to each odd period; (*ii*) the duration of each annual crop is satisfied; (*iii*) second semester crops are possibly assigned to even periods; (*iv*) the constraints on the maximum replanting are respected; (*v*) the GAEC 7 *Crop Rotation* is satisfied and the reward of Eco-scheme 4 is taken into account; (*vi*) CdM constraints are fulfilled. (Note that, in CAP 2023-2027, the *Crop Diversification* constraint and Eco-scheme 5 do not apply on a single plot.)

In what follows, for sake of clarity, the formulas are given for the case of $k = 3$ (and one single plot), useful to solve CRP-k in the Mediterranean context. We point out that the approach can be suitably modified for $k > 3$.

Given a crop $c \in C$, let $P(q)$ be the optimal total profit for CRP-3 with one plot, restricted to the first q seeding periods. Let $\psi(q, c_1, c_2, c_3, c_4, r)$ be the optimal solution value of CRP-k on a single plot, restricted to the first q periods, when the last four crops in the sequence σ are c_1, c_2, c_3, c_4, and the current replanting of c_4 is r, i.e., the $r-1$ crops immediately preceding c_4 in σ belong to the same botanical group of c_4. Then, by definitions of $P(q)$ and $\psi(q, c_1, c_2, c_3, c_4, r)$, we have

$$P(q) = \max_{c_1,c_2,c_3,c_4 \in C;\ r=1,\ldots,R(c_4)} \psi(q, c_1, c_2, c_3, c_4, r), \tag{1.52}$$

where $R(c)$ represents the maximum replanting value of the botanical group (in B) c belongs to.

The function $\psi(q, c_1, c_2, c_3, c_4, r)$ can be computed by the following recursive formulas, depending on the period q, the types of c_3 and c_4, and the value of r.

If q is an even period and c_4 is a first semester or annual crop we simply have

$$\psi(q, c_1, c_2, c_3, c_4, r) = \psi(q-1, c_1, c_2, c_3, c_4, r), \tag{1.53}$$

since c_4 can be assigned to odd periods only.

If c_3 is an annual crop and c_4 is a second semester crop or if q is an odd period and c_4 is a second semester crop, we have:

$$\psi(q, c_1, c_2, c_3, c_4, r) = -\infty, \tag{1.54}$$

for any q and r, because, by definition, an annual crop can be only followed by first semester or annual crops and a second semester crop cannot be sown in odd periods. On the other hand, if q is an even period and c_4 is a second semester crop, or if q is an odd period and c_4 is a first semester or annual crop, the following recursive formulas (1.55)–(1.57) hold, depending on whether c_3 and c_4 belong to the same replanting group or not. In the formulas below, $B(c_3)$ and $B(c_4)$ denote the replanting groups of c_3 and c_4, respectively, and $p(\langle c_2, c_3, c_4\rangle, q)$, as introduced in Section 1.4, is the profit earned by growing crop c_4 in period q in a 3-rotation with crops c_3 and c_2, on a single plot.
If $B(c_3) \neq B(c_4)$ then:

$$\psi(q, c_1, c_2, c_3, c_4, r) = -\infty \quad \text{if } 1 < r \leq R(c_4) \tag{1.55}$$

$$\psi(q, c_1, c_2, c_3, c_4, 1) = \max_{c \in C,\ r \in R(c_3)} \{\psi(q-1, c, c_1, c_2, c_3, r) + p(\langle c_2, c_3, c_4\rangle, q)\} \tag{1.56}$$

If $B(c_3) = B(c_4)$ and if $1 < r \leq R(c_4)$ then:

$$\psi(q, c_1, c_2, c_3, c_4, r) = \max_{c \in C} \{\psi(q-1, c, c_1, c_2, c_3, r-1) + p(\langle c_2, c_3, c_4\rangle, q)\} \tag{1.57}$$

In case that $B(c_3) = B(c_4)$ and $r = 1$:

$$\psi(q, c_1, c_2, c_3, c_4, 1) = -\infty \tag{1.58}$$

The procedure must be initialized by computing function ψ for all possible crop sequences σ containing at most four crops, all admissible periods $t \leq 7$ (since for $t > 7$ σ contains more than 5 crops) and all admissible replanting values of r (which depend on the sequence σ). As an example, if $\sigma = \langle c_1, c_2, c_3, c_4 \rangle$ with $c_1, c_3 \in C_F$ and $c_2, c_4 \in C_S$, then q must be 4, and we have:

$$\psi(4, c_1, c_2, c_3, c_4, r) = p(\langle c_1 \rangle, 1) + p(\langle c_1, c_2 \rangle, 2) + p(\langle c_1, c_2, c_3 \rangle, 3) + p(\langle c_2, c_3, c_4 \rangle, 4),$$

where r is the number of crops immediately preceding c_4 belonging to the same botanical group of c_4.

The above formulas allow to compute the optimal solution $P(q)$ (by Formula (1.52)) of CRP-k on a single plot, as well as the optimal crop sequence. By Formulas (1.52)–(1.58), since $r \leq p$, it follows that $P(q)$ can be computed in $O(n^5 p^2)$ operations. Hence, $P(q)$ requires $O(n^5 p^3)$ operations.

Modeling CAP 2023-2027 constraints

The above Dynamic Programming formulas can be extended to take into account the GAEC 7 – *Crop Rotation* requirements and the Eco-scheme 4.

GAEC 7 – *Crop Rotation*: Recall that, in the Rotation Scenario, a change of the botanical group is required after that crops of the same botanical group are assigned to periods of the same year. Hence, given a crop sequence $\sigma = \langle c_1, c_2, \ldots, c_i \rangle$, the crop c_h, with $h = 1, \ldots, i$ satisfies the *Crop Rotation* constraint if c_h is a second semester crop (since crops in C_S always belong to different botanical groups than crops in C_F and C_A), or if $c_h \in C_A \cup C_F$ and one of the following cases holds:

- if $c_{h-1} \in C_F \cup C_A$ and $B(c_h) \neq B(c_{h-1})$;
- if $c_{h-1} \in C_S$.

Eco-scheme 4: Let C_{E4} be the set containing forage and the renewal crops, i.e., the crops required to get Eco-scheme 4 reward. A function $\chi(q, c_1, c_2, c_3, c_4, r)$ is introduced, having the same arguments of ψ, and equal to the last even period l in which the reward of Eco-scheme 4 is get, where $l \in \{0, 4, 6, ..., p\}$. The function is defined as follows:

$$\chi(q, c_1, c_2, c_3, c_4, r) = l$$

The function χ follows the same formulas as in equations (1.53)–(1.58), and the value of χ is update only when ψ is also updated. In particular, when $\psi(q, c_1, c_2, c_3, c_4, r)$ is computed, if $c_4 \in C_{E4}$ and $\chi(q-1, c, c_1, c_2, c_3, r) = l < q$, the reward of Eco-Scheme 4 is added to $\psi(q, c_1, c_2, c_3, c_4, r)$ and $\chi(q, c_1, c_2, c_3, c_4, r)$ is set to a value ranging between

q and $q+3 \leq p$, depending on the period q and the value of $\chi(q-1, c, c_1, c_2, c_3, r)$. Otherwise, if $l \geq q$ or $c_4 \notin C_{E4}$ then $\chi(q, c_1, c_2, c_3, c_4, r) = \chi(q-1, c, c_1, c_2, c_3, r)$.

In particular, the function χ is updated as follows:

- if $q - l \geq 4$, q is an even period:

$$\chi(q, c_1, c_2, c_3, c_4, r) = q$$

- if $q - l \geq 4$, q is an odd period:

$$\chi(q, c_1, c_2, c_3, c_4, r) = q + 1$$

- if $0 < q - l < 4$, q is an even period:

$$\chi(q, c_1, c_2, c_3, c_4, r) = q + 2$$

- if $0 < q - l < 4$, q is an odd period:

$$\chi(q, c_1, c_2, c_3, c_4, r) = q + 3$$

- if $q \leq l$

$$\chi(q, c_1, c_2, c_3, c_4, r) = \chi(q-1, c, c_1, c_2, c_3, r)$$

Modelling CdM constraints

As described in Section 1.5.2, the "Carta del Mulino" (CdM) initiative imposes three types of agronomic constraints. In formulas (1.56)–(1.57), if crop c of $\psi(q-1, c, c_1, c_2, c_3, r)$ is c_{CdM}, the value $\psi(q, c_1, c_2, c_3, c_4, r)$ is evaluated only if the sequence from c on, i.e., $\sigma = \langle c, c_1, c_2, c_3, c_4 \rangle$, satisfies all the CdM constraints, as described below.

- *Greening Constraint:* Let C_L and C_O denote the sets of legume and oilseed crops, respectively. The greening constraint is satisfied if at least one crop in σ belongs to either C_L or C_O.
- *Diversification Constraint:* The constraint is satisfied if σ contains at least three different crops.
- *Repetition Constraint:* Let C_R be the set of crops defined in the repetition constraints already introduced above, with $c_{CdM} \in C_R$. The crops of any triple of consecutive crops in σ must not belong to C_R.

1.8.3 Upper bounds and Heuristic methods for CRP-k

The arc-flow formulation for CRP-k (see Section 1.8.2.1) provides a compact model, but it can become computational demanding for large instances. To address this limitation, firstly, we develop a Lagrangian relaxation to handle the management requirements 2-3 and the CAP constraints, with the aim of computing bounds to the optimum. Then, two classes of heuristic methods are presented: (i) a dynamic-programming-based heuristic (DPH) that exploits single-plot optimal crop sequences to construct high-quality feasible sequences for embedding in the graph G; and (ii) a column-generation-based matheuristic (CGH) that solves a restricted graph and generates promising columns—i.e., crop sequences—by solving pricing problem.

1.8.3.1 A Lagrangian relaxation approach to compute upper bounds for CRP-k under CAP 2014–2020 Scenario

In the context of the CAP 2014–2020 scenario, we developed a Lagrangian relaxation approach to handle the sustainability constraints imposed by the policy and the management requirements 2-3, with the objective of compute bounds for the original problem. This method is particularly suitable due to the structural properties of the underlying network. Since the graph G is acyclic, by relaxing constraints on maximum and minimum land surface and CAP constraints, the resulting problem can be decomposes into m minimum-cost flow problems on G, one for each plot class H_w, $w = 1, \dots, m$.

The key idea is firstly to report the objective function in the minimization form as:

$$max \sum_{w=1}^{m} \sum_{(i,j)\in A} p_{ij}^w x_{ij}^w \rightarrow -min \sum_{w=1}^{m} \sum_{(i,j)\in A} p_{ij}^w x_{ij}^w$$

where the coefficient p_{ij}^w are all negative. Then, the constraints are relaxed and added them into the objective function, each weighted by a corresponding Lagrangian multiplier, and then solve the resulting problem. This produces a valid *lower bound* to the minimization problem, since the relaxation enlarges the feasible region. However, in the Lagrangian relaxation, the entire graph G must be explicitly constructed, which may become computationally demanding for large instances, as discussed in the previous section.

Based on this observation, the idea is to solve the following Lagrangian problem (given in the minimization form):

$$\min \mathcal{L}(x, z, \lambda, \mu, \sigma, \tau, \alpha, \beta, \gamma) \tag{1.59}$$

$$\sum_{(i,j)\in\delta^+(i)} x_{ij}^w - \sum_{(j,i)\in\delta^-(i)} x_{ji}^w = \begin{cases} |H_w|, \text{ if } \ i = so_w \\ -|H_w|, \text{ if } \ i = de_w \\ 0, \ \forall i \in V \setminus \{so_w, de_w\} \end{cases} \qquad \forall w \in \{1, \dots, m\} \tag{1.60}$$

$$x_{ij}^w \leq 0.75 S_{tot} \qquad \forall (i,j) \in A, \ w \in \{1, \dots, m\} \tag{1.61}$$

$$x_{ij}^w \in \{0, 1, 2, \dots, |H_w|\} \qquad \forall (i,j) \in A, \ w \in \{1, \dots, m\} \tag{1.62}$$

$$z_c^t \in \{0, 1\} \qquad \forall c \in C, \ t \in T \tag{1.63}$$

It can be easily seen that model (1.59)–(1.63) is a minimum cost flow problem, where Constraints (1.61) represents an upper bound on each arc. Such bound arises from the

CAP *Crop Diversification* rule, which imposes that crops belonging to the same botanical family can cover at most 75% of the farmland in each odd period. Consequently, each individual crop cannot exceed this surface limit.

The Lagrangian function $\mathcal{L}(x, z, \lambda, \mu, \sigma, \tau, \alpha, \beta, \gamma)$ takes the following form:

$$
\begin{aligned}
\mathcal{L}(x, z, \lambda, \mu, \sigma, \tau, \alpha, \beta, \gamma) = & \sum_{w=1}^{m} \sum_{(i,j)\in A} (-p_{ij}^{w}) x_{ij}^{w} \\
& + \sum_{c\in C\setminus\{c_{fallow}\}} \sum_{t\in T} \lambda_c^t (\min_c^t - \sum_{w=1}^{m} s_w \sum_{(i,j)\in A:[c,t]\in j} x_{ij}^{w}) \\
& - \sum_{c\in C\setminus\{c_{fallow}\}} \sum_{t\in T} \mu_c^t (\max_c^t - \sum_{w=1}^{m} s_w \sum_{(i,j)\in A:[c,t]\in j} x_{ij}^{w}) \\
& + \sum_{c\in C\setminus\{c_{fallow}\}} \sum_{t\in T} \sigma_c^t (\text{minIF}_c^t z_c^t - \sum_{w=1}^{m} s_w \sum_{(i,j)\in A:[c,t]\in j} x_{ij}^{w}) \\
& - \sum_{c\in C\setminus\{c_{fallow}\}} \sum_{t\in T} \tau_c^t (S_{\text{tot}} z_c^t - \sum_{w=1}^{m} s_w \sum_{(i,j)\in A:[c,t]\in j} x_{ij}^{w}) \\
& + \sum_{t\in T_o} \alpha^t (0.05\, S_{\text{tot}} - \sum_{w=1}^{m} s_w \sum_{(i,j)\in A:[c,t]\in j,\, c\in C_{\text{leg}}} x_{ij}^{w}) \\
& - \sum_{g=1}^{l} \sum_{t\in T_o} \beta_g^t (0.75\, S_{\text{tot}} - \sum_{w=1}^{m} s_w \sum_{(i,j)\in A:[c,t]\in j,\, c\in B_g} x_{ij}^{w}) \\
& - \sum_{\substack{f,g=1 \\ f<g}}^{l} \sum_{t\in T_o} \gamma_{f,g}^t (0.95\, S_{\text{tot}} - \sum_{w=1}^{m} s_w \sum_{(i,j)\in A:[c,t]\in j,\, c\in B_f\cup B_g} x_{ij}^{w})
\end{aligned}
$$

For any fixed multipliers, the optimal value of the Lagrangian function is a lower bound for the minimization reformulation. Therefore, its negation provides an upper bound for the original maximization problem. Instead $\lambda, \mu, \sigma, \tau, \alpha, \beta, \gamma$ denote non-negative *Lagrangian multipliers*. Each multiplier is associated with the constraints on maximum and minimum land surface and CAP constraints, and the dimensions of these multipliers differ. Specifically, λ, μ, σ, and τ are indexed by each crop $c \in C$ (excluding the fallow crop) and each planning period $t \in T$, so their dimension is $(|C| - 1) \times |T|$. Meanwhile, α has dimension $|T_o|$, β has dimension $l \times |T_o|$, and γ has dimension $l(l-1)/2 \times |T_o|$, where l is the size of the botanical groups to which crops in $C_A \cup C_F$ belong.
Each multiplier reflects the penalty for violating its corresponding constraint and, since all are non-negative, violations increase the penalty term, making it positive. Thus, the constraints tend to be satisfied if the associated multiplier is sufficiently large. Once the Lagrangian multipliers are fixed, the relaxed problem could be further decomposed with respect to the variables x_{ij}^w and z_c^t. This decomposition is not reported in detail for the sake of simplicity, since, in the minimization setting, the optimal assignment of the binary variables z_c^t is immediate: each variable is set to 0 when its associated coefficient is non-negative and to 1 otherwise. The best bound can be obtained by solving the following

optimization problem:

$$\mathcal{L}(x, z, \lambda^*, \mu^*, \sigma^*, \tau^*, \alpha^*, \beta^*, \gamma^*) = max\{\mathcal{L}(x, z, \lambda, \mu, \sigma, \tau, \alpha, \beta, \gamma) : \lambda, \mu, \sigma, \tau, \alpha, \beta, \gamma \geq 0\},$$

where $\mathcal{L}(x, z, \lambda^*, \mu^*, \sigma^*, \tau^*, \alpha^*, \beta^*, \gamma^*)$ is called *Lagrangian dual*.

Subgradient Algorithm for Lagrangian Relaxation

The Lagrangian dual problem can be computed by using a subgradient algorithm specifically tailored to the CRP-k. The method iteratively updates the multipliers associated with the relaxed constraints by exploiting subgradient information, in order to improve the quality of the dual bound.

At each iteration k, the relaxed problem defined by (1.59)–(1.63) is solved, yielding a solution $(x^{(k)}, z^{(k)})$. The corresponding subgradient vector is then computed, with one component for each relaxed constraint. If all the component of the subgradient vector are less or equal to zero, the current solution is an admissible solution, i.e. a upper bound (UB), in the minimization case, for the original problem and we can update the value. Otherwise, the multipliers are updated according to a step-size rule based on the difference between the best known upper bound UB and the current dual value. The updated multipliers define a new Lagrangian problem for the next iteration, allowing the algorithm to progressively refine the bound.

The algorithm terminates either when the step-size becomes sufficiently small, no improvements are observed after K iterations or in case that the value of the best upper bound UB and the lower bound LB coincide, i.e., an optimal solution is found. This procedure is designed to improve the Lagrangian dual bound while preserving computational tractability for medium-scale instances.

The algorithm 1 shows the tailored subgradient algorithm applied to the CRP-k.

Each component of the subgradient vector s therefore measures the violation of the corresponding relaxed constraint at each step k of the algorithm. The corresponding subgradient s has the following components:

- For each $c \in C \setminus \{c_{\text{fallow}}\},\ t \in T$:

$$s_\lambda = \min_c^t - \sum_{w=1}^{m} s_w \sum_{(i,j)\in A:[c,t]\in j} x_{ij}^w.$$

$$s_\mu = \sum_{w=1}^{m} s_w \sum_{(i,j)\in A:[c,t]\in j} x_{ij}^w - \max_c^t.$$

$$s_\sigma = \text{minIF}_c^t\, z_c^t - \sum_{w=1}^{m} s_w \sum_{(i,j)\in A:[c,t]\in j} x_{ij}^w.$$

$$s_\tau = \sum_{w=1}^{m} s_w \sum_{(i,j)\in A:[c,t]\in j} x_{ij}^w - S_{\text{tot}} z_c^t.$$

Algorithm 1: Subgradient Algorithm

Input: Lagrangian function $\mathcal{L}(x, z, \lambda, \mu, \sigma, \tau, \alpha, \beta, \gamma)$, initial multipliers $\lambda^{(0)}, \mu^{(0)}, \sigma^{(0)}, \tau^{(0)}, \alpha^{(0)}, \beta^{(0)}, \gamma^{(0)}$, maximum number of iterations without improvement K, step-size parameter α_0, initial UB, initial LB
Initialization: $k \leftarrow 0$, $\alpha_k \leftarrow 2$

1. Set $\lambda = \lambda^{(k)}, \mu = \mu^{(k)}, \sigma = \sigma^{(k)}, \tau = \tau^{(k)}, \alpha = \alpha^{(k)}, \beta = \beta^{(k)}, \gamma = \gamma^{(k)}$
2. Compute the optimal solution of the Lagrangian problem $\mathcal{L}^{(k)} = \mathcal{L}(x^{(k)}, z^{(k)}, \lambda^{(k)}, \mu^{(k)}, \sigma^{(k)}, \tau^{(k)}, \alpha^{(k)}, \beta^{(k)}, \gamma^{(k)})$
3. Compute subgradient $s^{(k)}$
4. **If** $s^{(k)} \leq 0$, update UB
5. **If** $\mathcal{L}^{(k)} > LB$, update $LB = \mathcal{L}^{(k)}$
6. Compute step size:
$$\theta^{(k)} = \frac{\alpha_k \left(UB - \mathcal{L}^{(k)}\right)}{\|s^{(k)}\|}$$
7. Update multipliers (here showed just for multipiers λ):
$$\lambda^{(k+1)} \leftarrow \max\left\{0,\ \lambda^{(k)} + \theta^{(k)} \frac{s_\lambda^{(k)}}{\|s^{(k)}\|}\right\}.$$
8. Set $k \leftarrow k + 1$
9. **If** $\mathcal{L}(x, z, \lambda, \mu, \sigma, \tau, \alpha, \beta, \gamma)$ has not improved in the last K iterations, **then** $\alpha_k \leftarrow \alpha_k / 2$
10. **If** $\alpha_k < \alpha_0$, **then** STOP, else go to step 1
11. **If** $UB = LB$, **then** STOP, else go to step 1

- For each $t \in T_o$:
$$s_\alpha = 0.05\, S_{\text{tot}} - \sum_{w=1}^{m} s_w \sum_{(i,j)\in A:[c,t]\in j,\, c\in C_{\text{leg}}} x_{ij}^w.$$

- For each botanic group $g = 1, \ldots, l$, $t \in T_o$:
$$s_\beta = \sum_{w=1}^{m} s_w \sum_{(i,j)\in A:[c,t]\in j,\, c\in B_g} x_{ij}^w - 0.75\, S_{\text{tot}}.$$

- For each pair of botanic groups $f, g \in \{1, \ldots, l\}$, $f < g$, $t \in T_o$:

$$s_\gamma = \sum_{w=1}^{m} s_w \sum_{(i,j)\in A:[c,t]\in j,\, c\in B_f\cup B_g} x_{ij}^w - 0.95\, S_{\text{tot}}.$$

Collecting all these components, the subgradient vector can be written compactly as $s = \Big[[s_\lambda], [s_\mu], [s_\sigma], [s_\tau], [s_\alpha], [s_\beta], [s_\gamma]\Big]$, where the total dimension is $4 \cdot (|C| - 1) \cdot |T| + (1 + l + \frac{l(l-1)}{2}) \cdot |T_o|$. Thus, the subgradient vector has one component for each relaxed constraint, and its dimension coincides with the total number of Lagrangian multipliers.

The results reported in Table 1.16 indicate that this approach is not particularly effective for CRP-k. Although the obtained bounds are relatively tight, the computational effort required to compute them is excessively high. For this reason, we decided not to pursue this direction further and refrained from developing either a branch-and-bound scheme or heuristic methods based on Lagrangian relaxation.

1.8.3.2 A Column Generation based matheuristic (CGH) for CRP-k under the CAP 2014–2020 Scenario

In this section, a matheuristc for CRP-k is presented for CAP 2014–2020 Scenario, hereinafter referred to as CGH, based on solving the arc flow formulation G-ILP (presented in 1.8.2.1) by column generation. The use of a column generation framework for G-ILP emerges as a strategic approach to enable the resolution of large instances that would otherwise be unsolvable. In particular, in CGH, a restricted G-ILP model is first defined, denoted as $\bar{G}$-ILP, in which a reduced number of variables x_{ij}^w, and all variables z_c^t exist. The number of variables z_c^t is polynomial in the problem size and, so, it is not an issue to generate all of them from the start. Then, the algorithm iteratively solves the linear relaxation of the current restricted model and employs a pricing problem to generate new variables x_{ij}^w with positive reduced cost to add to the restricted model, mimicking the Simplex algorithm for LP. When no more variables are found by the pricing problem (or a maximum number of iterations is reached), the iterative core of the algorithm stops and the restricted model obtained so far is solved by a commercial ILP solver. In what follows, let $\bar{G} = (\bar{V}, \bar{A})$ be a reduced graph of $G = (V, A)$, with $\bar{A} \subset A$ and $\bar{V} \subset V$, and let $\bar{G}$-ILP be the corresponding restricted G-ILP model, where a variables x_{ij}^w exist only if $(i, j) \in \bar{A}$, for $w = 1, \ldots, m$.

As shown later (see Section 1.8.3.2), the initial graph $\bar{G}$ and the corresponding model $\bar{G}$-ILP are determined by an algorithm called $\bar{G}$-Start, in such a way that $\bar{G}$-ILP allows a feasible solution for CRP-k. The general scheme of algorithm CGH is reported in Algorithm 2. The solution of the pricing problem for $\bar{G}$-ILP is discussed in the following section, where its strong NP-hardness is proved, too. In the following, a heuristic approach based on Dynamic Programming for solving the pricing problem is presented.

Solving the pricing problem

Consider the dual problem of the linear relaxation of $\bar{G}$-ILP (i.e., of G-ILP on the reduced graph $\bar{G}$). In the dual problem, let α_i^w, β_{ct}, ω_{ct}, γ_{ct}, δ_t, ϵ_t, ν_{gt} and μ_{gt} denote the dual

Algorithm 2: General scheme of *CGH*.

Input: A CRP-k instance

Initialization: $it_{max} = IT$, $it = 0$, STOP=0
Generate the initial graph $\bar{G}$ by algorithm $\bar{G}$-Start (Section 1.8.3.2) and the starting model $\bar{G}$-ILP;

while STOP=0 **and** $it \leq it_{max}$, **do**
  Solve the linear relaxation $\bar{G}$-LP of $\bar{G}$-ILP; Solve the pricing problem using the heuristic algorithm;
  if positive reduced cost variables x_{ij}^w are found **then**
    Add the new variables to $\bar{G}$-LP; $it = it + 1$;
  else
    STOP=1;

Solve the resulting $\bar{G}$-ILP and **return** the solution.

variables associated with the flow balance, maximum and minimum land surface, and CAP constraints, respectively. Recalling that, in G-ILP (and $\bar{G}$-ILP), each variable x_{ij}^w is related to the plot class H_w and to the arc $(i,j) \in G$, where nodes i and j are of the form $i = (a_1, a_2, \ldots, a_\tau)^{t_1,t_2,\ldots,t_\tau}$ and $j = (a_{\tau+1}, a_{\tau+2}, \ldots, a_{2\tau})^{t_{\tau+1},t_{\tau+2},\ldots,t_{2\tau}}$, the dual constraint corresponding to the variable x_{ij}^w reads as:

$$\alpha_i^w - \alpha_j^w + s_w \sum_{l=\tau+1}^{2\tau} (-\beta_{a_l,t_l} + \omega_{a_l,t_l} + \gamma_{a_l,t_l} - \delta_{a_l,t_l} + \epsilon_{t_l} - \nu_{g_l,t_l} - \sum_{f \in B,\, f \neq g_l} \mu_{g_l,f,t_l}) \leq -p_{ij}^w. \tag{1.64}$$

By definition, the profit p_{ij}^w is computed as in (1.2). Recall that p_{ij}^w is the profit attained cultivating all the crops in node j on a plot of class H_w, while the crops in i contribute in determining the specific rotation occurring to the crops in j.

Given the optimal solution of the linear relaxation of $\bar{G}$-ILP and the corresponding optimal solution of the dual problem, the pricing problem consists in finding, if any, an arc $(i,j) \in A \backslash \bar{A}$, where $i = (a_1, a_2, \ldots, a_\tau)^{t_1,t_2,\ldots,t_\tau}$ and $j = (a_{\tau+1}, a_{\tau+2}, \ldots, a_{2\tau})^{t_{\tau+1},t_{\tau+2},\ldots,t_{2\tau}}$, such that the left-hand side of (1.64) is strictly bigger than $-p_{ij}^w$.

Such an arc can be detected by finding a crop sequence $\langle a_1, a_2, \ldots, a_\tau, a_{\tau+1}, a_{\tau+2}, \ldots, a_{2\tau} \rangle$ and the corresponding seeding periods $t_1, t_2, \ldots, t_\tau, t_{\tau+1}, t_{\tau+2}, \ldots, t_{2\tau}$, such that the expression, i.e., the *reduced cost* of the arc (i,j), is strictly positive:

$$\alpha_i^w - \alpha_j^w + s_w \sum_{l=\tau+1}^{2\tau} (-\beta_{a_l,t_l} + \omega_{a_l,t_l} + \gamma_{a_l,t_l} - \delta_{a_l,t_l} + \epsilon_{t_l} - \nu_{g_l,t_l} - \sum_{f \in B,\, f \neq g_l} \mu_{g_l,f,t_l}) + p_{ij}^w > 0. \tag{1.65}$$

Although such a problem has similarities to CRP-k on a single plot, for which a polynomial-time algorithm is given in Section 1.8.2.2, the pricing problem is strongly NP-hard, as shown in the following theorem.

Theorem 1.8.4. *The pricing problem for G-ILP is strongly NP-hard for $k \geq 2$.*

Proof. The proof is reported in the Appendix. □

The above theorem does not hold when $k = 1$, i.e., the profit of a crop depends only on the crop itself. In this case, the following theorem establishes that the pricing problem is (weakly) NP-hard.

Theorem 1.8.5. *The pricing problem for G-ILP for $k = 1$ is NP-hard.*

Proof. The proof is reported in the Appendix. □

A heuristic algorithm for the pricing problem

As shown in Theorem 1.8.4, the pricing problem for G-ILP is hard to solve optimally. In this section, a heuristic algorithm for the pricing problem is proposed, based on the appropriate modification of the Dynamic Programming approach presented in Section 1.8.2.2 for CRP-k on a single plot.

As shown above, the pricing problem consists in finding an arc $(i, j) \in A \setminus \bar{A}$, where $i = (a_1, a_2, \ldots, a_\tau)^{t_1,t_2,\ldots,t_\tau}$ and $j = (a_{\tau+1}, a_{\tau+2}, \ldots, a_{2\tau})^{t_{\tau+1},t_{\tau+2},\ldots,t_{2\tau}}$, such that the reduced cost (1.65) is positive.

In what follows, for the sake of simplicity, we consider the pricing problem for G-ILP for the case $\tau = k = 3$ (i.e., a node in G contains sequences of $\tau = 3$ crops). Given a starting odd period t_1 that will be assigned to the first crop of node i, we denote by $\sigma(q, c_1, c_2, r)$ the optimal crop sequence of a single plot problem restricted to the first q periods, in which: (i) the first crop in the sequence is assigned to period t_1; (ii) the last two crops of the sequence are c_1, c_2 (in this order); (iii) the seeding period of c_2 is at most q; (iv) the replanting of the last crop c_2 of the sequence is r.

By (1.65), we have that the reduced cost of an arc (i, j) depends on the (at most) τ crops of node j, i.e., $a_{\tau+1}, a_{\tau+2}, \ldots, a_{2\tau}$, while node i (and its τ crops $a_1, a_2, \ldots, a_\tau$) only contributes to (1.65) with the term α_i^w. Hence, if $\sigma(q, c_1, c_2, r)$ contains at least τ crops, $a_1, a_2, \ldots, a_d$, with $d \geq \tau$, we associate to $\sigma(q, c_1, c_2, r)$ a value, denoted as $\bar{\psi}(q, c_1, c_2, r)$, defined as follows:

$$\bar{\psi}(q, c_1, c_2, r) = \alpha_i^w + s_w \sum_{l=\tau+1}^{d} \Big(- \beta_{a_l,t_l} + \omega_{a_l,t_l} + \gamma_{a_l,t_l} - \delta_{a_l,t_l} + \epsilon_{t_l} - \nu_{g_l,t_l} - \sum_{f \in B,\, f \neq g_l} \mu_{g_l,f,t_l} \Big) + \sum_{l=\tau+1}^{d} p_w(\sigma_k(a_l), t_l). \tag{1.66}$$

In (1.66), $t_1, t_2, \ldots, t_d$ are the seeding periods of the crops in the sequence $\sigma(q, c_1, c_2, r)$, and α_i^w is the dual variable associated to the node $i = (a_1, a_2, \ldots, a_\tau)^{t_1,t_2,\ldots,t_\tau}$. In the last term of (1.66), $p_w(\sigma_k(a_l), t_l)$ is the profit as defined in Section 1.4.2, i.e., the profit earned by growing crop a_l, in period t_l, on a plot of H_w, when the (at most) $k - 1$ crops immediately preceding a_l on the plot are those given in $\sigma_k(a_l)$. On the other hand, if $\sigma(q, c_1, c_2, r)$ contains less of τ crops, $\bar{\psi}(q, c_1, c_2, r)$ is set to 0.

Recursive formulas similar to those of Section 1.8.2.2 can be devised. As an example, for $k = \tau = 3$, formulas (1.42)–(1.47) become:

$$\bar{\psi}(q, c_1, c_2, 1) = \max_{c \in C,\, r \in R(c_1)} \{\bar{\psi}(q-1, c, c_1, r) + \bar{p}(\langle c, c_1, c_2 \rangle, q) + \rho_i - \rho_j\} \quad \text{if } B(c_1) \neq B(c_2) \tag{1.67}$$

$$\bar{\psi}(q, c_1, c_2, r) = \max_{c \in C} \{\bar{\psi}(q-1, c, c_1, r-1) + \bar{p}(\langle c, c_1, c_2 \rangle, q) + \rho_i - \rho_j\} \quad \text{if } B(c_1) = B(c_2) \text{ and } r > 1 \tag{1.68}$$

where, $\bar{p}(\langle c, c_1, c_2\rangle, q)$, ρ_i and ρ_j are defined in the following, to take into account the optimal dual variables of the linear relaxation of $\bar{G}$-ILP.

According to the reduced cost function (1.65), $\bar{p}(\langle c, c_1, c_2\rangle, q)$ is 0 if $\sigma(q-1, c, c_1, r)$ or $\sigma(q-1, c, c_1, r-1)$ contain at most $\tau - 1$ crops (i.e., c_2 is at most the τ-th crop to be inserted, and hence all crops inserted up to c_2 belong to node i), and equal to the following formula, otherwise:

$$\bar{p}(\langle c, c_1, c_2\rangle, q) = p_w(\langle c, c_1, c_2\rangle, q) + s_w(-\beta_{c_2,q} + \omega_{c_2,q} + \gamma_{c_2,q} - \delta_{c_2,q} + \epsilon_q - \nu_{g_2,q} - \sum_{f \in B,\, f \neq g_2} \mu_{g_2,f,q}), \tag{1.69}$$

where g_2 is the botanical group of c_2, and $\epsilon_q = 0$ if $c_2 \notin C_{leg}$. Observe that $\bar{p}(\langle c, c_1, c_2\rangle, q)$, defined in (1.69), is the contribution of crop c_2 to the (1.65).

The terms ρ_i and ρ_j in formulas (1.67) and (1.68) are used to properly take into account the dual variables α_i^w and α_j^w, which depend on the ending nodes i and j of the arc of $A \setminus \bar{A}$ we are generating and on the plot class H_w. In particular, $\rho_i = 0$ ($\rho_j = 0$) if the sequence $\sigma(q, c_1, c_2, r)$ contains less than τ crops (less than 2τ crops and q is smaller than $p-1$), since in this case the crops in $\sigma(q, c_1, c_2, r)$ are not enough to define a node i (a node j) of G. In fact, to define node i we need at least τ crops in $\sigma(q, c_1, c_2, r)$, while to define node j we require either 2τ crops, or more than τ crops and $q \geq p-1$, i.e., the planning horizon is terminated. If the crops in the sequence $\sigma(q, c_1, c_2, r)$ allow to define a node i and/or a node j of G, then we set $\rho_i = \alpha_i^w$ and/or $\rho_j = \alpha_j^w$ if the nodes already exist in the restricted graph $\bar{G}$, and 0 otherwise.

The above Dynamic Programming procedure is used to compute the term

$$\max_{c_1, c_2 \in C;\, r = 1, \ldots R_{max}} \bar{\psi}(q, c_1, c_2, r)$$

from $q = t_1 + 1$ on. The procedure is stopped either when (i) $\sigma(q, c_1, c_2, r)$ contains 2τ crops, or (ii) $q = p$. In Case (i), the arc of G generated by the procedure is such that the node tail (node head) of the arc contains the first (the last) τ crops of $\sigma(q, c_1, c_2, r)$. In Case (ii), the tail of the generated arc contains the first τ crops of $\sigma(q, c_1, c_2, r)$ while the head of the arc contains all the remaining crops of $\sigma(q, c_1, c_2, r)$. In both cases, the first crop of the tail is assigned to period t_1 and all other crops are assigned to the next admissible periods (i.e., odd periods to first semester and annual crops and even periods to second semester crops).

The overall Dynamic Programming procedure is then applied for each starting period t_1 from which an arc $(i, j) \in G$ can be generated and for each plot class H_w.

Observe that, an arc (i, j) in G and its reduced cost are completely defined when all the crops contained in nodes i and j are known. On the other hand, the Dynamic Programming procedure presented in this section selects new crops to add to the sequence only according to cost evaluations on the previous $\tau - 1$ crops, hence, disregarding all other crops in the sequence (which contribute to the definition of the arc). Consequently, the procedure does not optimally solve the pricing problem.

In Section 1.9.4, experimental results of the column generation based matheuristic CGH are presented, in which the pricing problem is heuristically solved by the Dynamic Programming approach described in this section. The results show the effectiveness of the algorithm in terms of solution quality and computation time.

Finding an initial feasible solution

The column generation based algorithm CGH starts from an initial restricted model $\bar{G}$-ILP that allows an initial feasible solution for CRP-k. At this aim, we devised an algorithm, denoted as $\bar{G}$-Start, that consists in running in cascade modified versions of the Dynamic Programming approach of Section 1.8.2.2, to identify crop sequences, i.e., paths, to be incorporated into $\bar{G}$, in such a way that a feasible solution exists for CRP-k, as discussed at the end of the section.

More precisely, $\bar{G}$-Start employs the following five variants of the algorithm presented in Section 1.8.2.2, denoted as DP_{best}, DP_{min}, DP_{no-max}, DP_L and DP_3:

DP_{best}: The Dynamic Programming algorithm is used to find the three crop sequences with the highest profits. Since limits exist on the land assigned to each crop or botanical family in each period, the highest profit sequence cannot be assigned to all the plots. The other two highest profit sequences can be possibly used for the remaining plots;

DP_{min}: This procedure is run for each crop c with minimum surface restrictions per period, i.e., crops c such that $\min_c^t > 0$ for some t, and is used to create sequences ensuring to satisfy such constraints. For each such c, DP_{min} is run a number of times to generate paths in $\bar{G}$ securing that c can be grown in each t such that $\min_c^t > 0$, without violating the maximum replanting constraints. This guarantees the existence of a solution that does not violate the minimum surface constraints, since, if $\min_c^t > 0$, there will always be a path in $\bar{G}$ involving c in period t.

DP_{no-max}: This procedure finds the highest profit sequences involving only crops with no maximum surface restrictions per period, i.e., crops c such that $\max_c^t = +\infty$ for all t. Also in this case, the three highest profit sequences found by DP_{no-max} are added to $\bar{G}$. DP_{no-max} ensures the existence of a solution that does not violate the maximum surface constraints, since a path involving only crops c such that $\max_c^t = +\infty$ for all t always exists in $\bar{G}$. In fact, recalling that the dummy fallow crop c_{fallow} is not constrained by maximum surface requirements, even when all other crops are subject to such restrictions, DP_{no-max} can generate a sequence that assigns fallow to all periods.

DP_L: This is a constrained version of DP_{best} where only crops from a pre-defined set can be chosen. Initially, the algorithm checks the best sequence seq_{best} determined by DP_{best}. Then, for each period, it only assigns crops whose botanical families are different from those allocated by seq_{best} in the same period, giving precedence to legume crops (in order to meet the CAP 2014–2020 Ecological Focus Area constraints (1.11)). The three highest profit sequences generated by DP_L are added to $\bar{G}$.

DP_3: This procedure is designed to produce the highest profit sequence that, in each period, assigns only crops of botanical families different from those assigned by DP_{best} and DP_L. This sequence is used to satisfy the CAP *Crop Diversification* constraints (1.13), imposing crops of at least three distinct botanical families in each odd period. The three highest profit sequences produced by this algorithm are incorporated into $\bar{G}$.

In algorithm $\bar{G}$-Start, the five Dynamic Programming procedures presented above are executed sequentially, according to the presentation order, i.e., DP_{best}–DP_{min}–DP_{no-max}–DP_L–DP_3, for each plot class $H_1, H_2, \ldots, H_m$.

In this way, it is guaranteed that in $\bar{G}$ for each class of plots: (i) by DP_{min}, there is at least one path involving a crop c in period t for each c, t such that $\min_c^t > 0$; (ii) by DP_{no-max}, there is at least one path only containing crops without maximum surface constraints; (iii) by DP_L, there is at least one path involving a legume crop in each period t; (iv) by DP_3, for each period t, there are at least three paths each involving a different botanical family in t (three different botanical families can be grown in each t). As a consequence, $\bar{G}$ allows a feasible solution to CRP-k. Furthermore, for each run of each algorithmic component, the three best sequences are included in $\bar{G}$, enlarging the solution space to possibly better solutions. The quality of the initial solution is also supported by the use of DP_{best}, which allows to include in $\bar{G}$ the highest profit crop sequences for each plot class.

1.8.3.3 A Dynamic Programming Heuristic Algorithm (DPH) for CAP 2023–2027 + CdM Scenarios

The extended Dynamic Programming algorithm for the single-plot case with CAP 2023-2027 and CdM constraints presented in Section 1.8.2.2 is able to find an optimal sequence satisfying all sequence-based constraints of the problem. However, the general CRP-k features other constraints that are not present in the single-plot case, such as the *Crop Diversification* constraint of the CAP 2023-2027 or the agronomic and management requirements 2 and 3 (see Section 1.4.3). In this section, a heuristic is presented, denoted as DPH, that employs the Dynamic Programming approach for the single-plot case to provide a feasible solution for the general CRP-k .

In a first phase, DPH applies different variants of the single-plot Dynamic Programming algorithm to generate a set of crop sequences, each assigning crops from period 1 to p. In a second phase, a graph $\bar{G}$ (similar to the one introduced in Section 1.8.1 for the arc-flow model G-ILP) is generated to include all and only these sequences. Then, a restricted arc-flow ILP formulation, called $\bar{G}$-*ILP*, is derived from $\bar{G}$ and solved to optimality. The sequences in the first phase are generated in such a way that $\bar{G}$-*ILP* allows a feasible solution to CRP-k.

More precisely, in the first phase, DPH executes the five following variants (denoted as DP_{best}, DP_{min}, DP_{no-max}, DP_2 and DP_3) of the single-plot Dynamic Programming algorithm to generate a set of crop sequences. Each variant consists in applying Formula (1.52) on a different crop set, as explained below (see Section 1.8.3.2 for a detailed description.).

DP_{best}: The single-plot Dynamic Programming algorithm is used to find the three crop sequences with the highest profits.

DP_{min}: This procedure is executed multiple times for each crop c with minimum surface requirement per period. It is used to create a set of sequences such that c is cultivated in each period where a minimum surface requirement exists for c.

DP_{no-max}: This procedure finds the highest three profit sequences involving only crops with no maximum surface restrictions per period. Note that, if all crops in C have a maximum surface restriction, since the fallow crop is not subject to such constraints, DP_{no-max} may assign it to all periods.

DP_2: This is a constrained version of DP_{best} where only crops from a predefined set can be chosen. Initially, the algorithm checks the best sequence seq_{best} determined by DP_{best}. Then, for each period, it only assigns crops whose botanical groups are different from those allocated by seq_{best} in the same period.

DP_3: This procedure is designed to produce the highest profit sequence that, in each period, assigns only crops of botanical families different from those assigned by DP_{best} and DP_2. This sequence is used to satisfy the *Crop Diversification* constraints, imposing crops of at least three distinct botanical groups in each odd period.

In the *Diversification Scenario*, all the five variants of the Dynamic Programming algorithm are executed, whereas in the *Rotation Scenario*, only DP_{best}, DP_{min}, and DP_{no-max} are applied. The three most profitable sequences generated by each variant are then incorporated into $\bar{G}$ and $\bar{G}$-*ILP* is solved to find a feasible solution to CRP-k.

In the following, we show how DPH builds the graph $\bar{G}$ by an example, in which, to be concise, only one crop sequence is produced for each of the five variants described above. Let us consider an instance of CRP-3 with a planning horizon $T = \{1, 2, \ldots, 10\}$, and a set of crops $C = \{c_1, c_2, c_3, c_4\}$, where $c_1, c_3 \in C_F$, $c_2 \in C_S$, and $c_4 \in C_A$. A maximum and a minimum land surface must be allocated to crops c_2 and c_3, respectively. Assume that each crop belongs to a different botanical group and $R = max_g R_{max}(g) = 3$, for $g \in \{1, 2, 3, 4\}$. Hence, on $\bar{G}$, the value of τ is 4, since $k = 3$ and $\tau = max\{k - 1, 4, R\}$. When DPH is executed, let us suppose that the following five crop sequences are produced by DP_{best}, DP_{min}, DP_{no-max}, DP_2, and DP_3 in the Diversification Scenario (As stated, in the Rotation Scenario, DP_2 and DP_3 are not used):

$$\mathrm{DP_{best}} : \sigma = \langle c_1,\ c_2,\ c_1,\ c_2,\ c_1,\ c_2,\ c_1,\ c_2,\ c_1,\ c_2 \rangle$$

$$\mathrm{DP_{min}} : \sigma = \langle c_3,\ c_2,\ c_3,\ c_2,\ c_3,\ c_2,\ c_3,\ c_2,\ c_3,\ c_2 \rangle$$

$$\mathrm{DP_{no\text{-}max}} : \sigma = \langle c_1,\ c_1,\ c_3,\ c_1,\ c_1 \rangle$$

$$\mathrm{DP_2} : \sigma = \langle c_3,\ c_3,\ c_4,\ c_3,\ c_3 \rangle$$

$$\mathrm{DP_3} : \sigma = \langle c_4,\ c_4,\ c_3,\ c_4,\ c_4 \rangle$$

The sequence produced by DP_{min} allows to satisfy the minimum land requirements of c_3, since the crop is assigned to each odd period. On the other hand, the sequence produced by DP_{no-max} allows to have some plots cultivated with crops different from c_2, and so to satisfy the maximum land surface requirement of c_2. Finally, $\mathrm{DP_2}$ and $\mathrm{DP_3}$ allow to satisfy the diversification constraints in the *Diversification Scenario* by cultivating crops of different botanical groups in each period. Note that, in the first two sequences, a crop is seeded in each of the 10 seeding periods, since they alternate first and second semester crops. Differently, in the last three sequences, only 5 crops are cultivated, since there is no second semester crop seeded in the even periods. Based on these sequences, the graph $\bar{G}$ reported in Figure 1.9 is derived, where each node represents a subsequence of $\tau = 4$ crops. Then, $\bar{G}$-*ILP* is generated and solved for finding a solution to CRP-k.

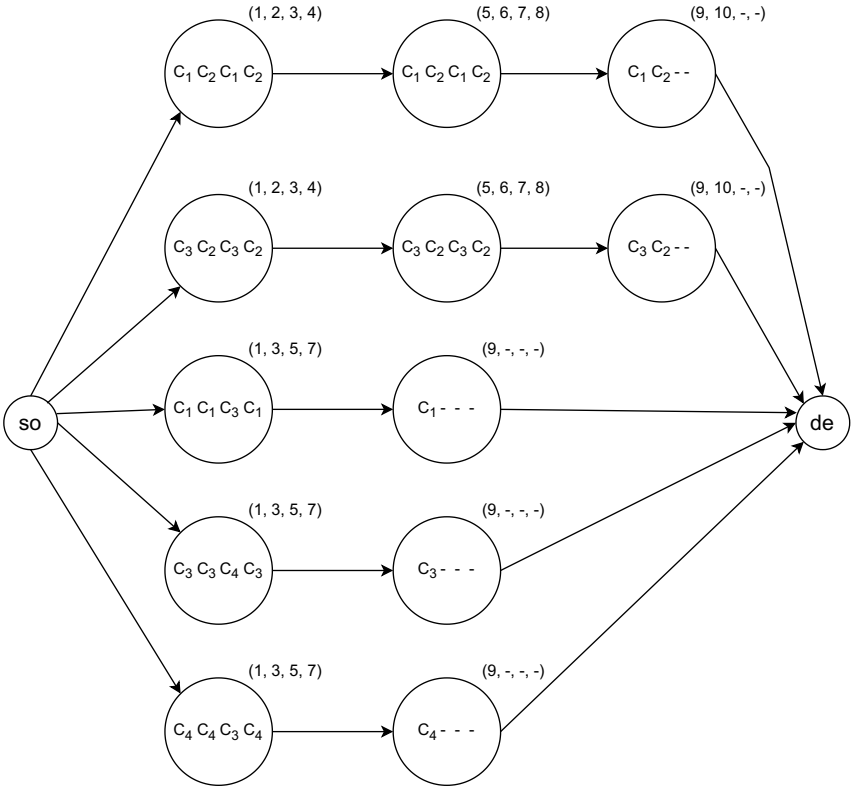

Figure 1.9: Graph $\bar{G}$ given by the sequences of DPH

1.9 Computational Experiments

An extensive experimental campaign was conducted to evaluate Exact and Heuristic methods introduced in Section 1.8. First, Section 1.9.1 reports a detailed description of the instances, including the plot classes, planning horizon, CAP and CdM payments, and the cost increases associated with crop rotation (as derived from [10]). Subsequently, the discussion of the results and the corresponding conclusions could be organized into the following parts:

- **ILP vs G-ILP** (Pure Farmer, CAP 2014–2020, CAP 2014–2020 + CdM).
  Section 1.9.2 compares the proposed arc-flow model (G-ILP), reported in Section 1.8.2.1, with the ILP formulation presented in [10]. The comparison is conducted for the scenarios listed in brackets. Results indicate that G-ILP obtain the same optimal values while drastically reducing computational time and reaching optimality at the root node. Such behaviour confirms the strength of the LP relaxation and its limited sensitivity to the number of plots, establishing G-ILP as the reference model for subsequent analyses.

- **G-ILP performance over different CAP scenarios** (CAP 2021–2023, CAP

2023–2027)
The analysis in 1.9.3 assesses the flexibility of G-ILP in modeling the various CAP policy frameworks – CAP 2021–2023, CAP 2023–2027 - Diversification, and CAP 2023–2027 - Rotation – by solving the corresponding formulations. For each scenario, both the computational effort and the resulting objective values are reported. The results demonstrate the capability of G-ILP to incorporate the additional constraints introduced by the CAP regulations.

- **CGH vs G-ILP** (CAP 2014–2020)
  To assess scalability, in Section 1.9.4, we test the column generation-based matheuristic (CGH) on the CAP 2014–2020 scenario over extended datasets with an increased number of crops. For these larger instances, G-ILP often reaches the time limit, while CGH consistently produces high-quality feasible solutions, confirming its suitability for large-scale settings.

- **DPH vs G-ILP** (CAP 2023–2027 + CdM)
  The Dynamic Programming heuristic (DPH) is analysed in Section 1.9.5 under the CAP 2023–2027 scenario, including the "Carta del Mulino" (CdM) constraints. Results show that DPH yields high-quality feasible solutions compared to G-ILP, while requiring substantially shorter computation times.
  This algorithm thus provides an efficient constructive approach for generating feasible solutions to the CRP-k.

- **Lagrangian relaxation vs G-ILP** (CAP 2014–2020)
  The Lagrangian relaxation developed for the CAP 2014–2020 scenario is evaluated in terms of its ability to produce valid bounds. The results are reported in Section 1.9.6. The method yields tight bounds with small optimality gaps, although at the cost of higher computational time compared to G-ILP.

1.9.1 Instance Description

The experiments were based on the 23 real-world instances introduced in [10] and on larger, newly generated instances derived from real-world data. Real-world instances originated from 23 farms across Italy, ranging from 19 to 229 hectares in size with 5 to 10 crops, primarily located in the Pianura Padana. Each farm has a set of crop choices, largely influenced by historical cultivation practices. The economic data for each crop were gathered through extensive analysis, involving structured surveys and official data sources. In the 23 real-world instances, 18 instances only have first semester or annual crops and five instances (named CN2, FE4, MN1, MN2, MN3 in what follows) also have one second semester crop, i.e., Soybean. For a detailed description of these instances and the data gathering approach, readers are directed to [10].

The use of real-world instances to test the models and algorithms provides solid validation for these decision-support methods, ensuring their effectiveness in assisting farmers in the real world. As the farms considered are in Italy, $k = 3$ is considered in each potential rotation scheme, implying that profits and costs of a crop depend on its two preceding crops (see [10] for details). In all the instances, the farmland is composed of

one plot class, hereafter called H_1, with homogeneous plots of 1-hectare; although the models and solution approaches are also applicable to farmlands with more plot classes and heterogeneous plots. A five year planning horizon has been considered in all the instances. Hence, $T = \{1, 2, \ldots, 10\}$. In the following, the real-world instances are referred to as Set_{real}.

To test the column generation-based metaheuristic (CGH), we generated two new sets of instances with a larger number of crops, based on the data from the 23 real-world instances of Set_{real}. The first set has been generated by including an additional 5 crops (denoted as Set_{+5}) in each instance of Set_{real}, and the second set by including an additional 10 crops (denoted as Set_{+10}) in each instance of Set_{real}. Specifically, only crops that are technically viable for that specific farm have been added. These new sets are designed to allow real farms to evaluate additional crops for potential cultivation, which could lead to improved economic outcomes.

In the Pure Farmer and CAP 2014–2020 scenarios, the crop c_{fallow} is not included, since the model is able to find a feasible solution with the existing crop set. In contrast, in the CAP 2021–2023 and CAP 2023–2027 scenarios, the crop c_{fallow} together with a crop from the set C_{flower} are always considered in order to obtain the reward from the Eco-schemes.

All the farms are specialized in cereals production (wheat and corn), but all of them have a legume crop or an oil crop between the starting crop portfolio. Hence, all the instances have a crop set designed to evaluate and satisfy all the CAP and CdM constraints introduced above.

As mentioned in Section 1.4, when crop rotation does not follow the best agronomic practices, a cost increase occurs depending on the specific crop succession in the rotation scheme. In the experiments, a cost increase ranging from 0% (best rotation practice) to 30% is assumed, depending on the specific rotation of crop types (i.e., renewal, improver, impoverishing). The cost increases considered in the experimental campaign are reported in Table 1.4. As an example, when three impoverishing crops are consecutively assigned to the same plot (the first rotation in Table 1.4), a cost increase of 30% (20%) occurs when the third (second) crop is cultivated.

We refer the reader to [10] for a detail description of the Table.

Crop Rotation scheme			Cost Increase		Crop Rotation scheme			Cost Increase	
Period 0	Period 1	Period 2	Period 1	Period 2	Period 0	Period 1	Period 2	Period 1	Period 2
Impoverishing	Impoverishing	Impoverishing	20%	30%	Renewal	Renewal	Improver	10%	0
Impoverishing	Impoverishing	Renewal	20%	0%	Renewal	Improver	Impoverishing	0	0
Impoverishing	Impoverishing	Improver	20%	0%	Renewal	Improver	Renewal	0	0
Impoverishing	Renewal	Impoverishing	0	10%	Renewal	Improver	Improver	0	10%
Impoverishing	Renewal	Renewal	0	10%	Improver	Impoverishing	Impoverishing	0	20%
Impoverishing	Renewal	Improver	0	0	Improver	Impoverishing	Renewal	0	0
Impoverishing	Improver	Impoverishing	0	0	Improver	Impoverishing	Improver	0	0
Impoverishing	Improver	Renewal	0	0	Improver	Renewal	Impoverishing	0	0
Impoverishing	Improver	Improver	0	10%	Improver	Renewal	Renewal	0	10%
Renewal	Impoverishing	Impoverishing	0	20%	Improver	Renewal	Improver	0	0
Renewal	Impoverishing	Renewal	0	10%	Improver	Improver	Impoverishing	10%	0
Renewal	Impoverishing	Improver	0	0	Improver	Improver	Renewal	10%	0
Renewal	Renewal	Impoverishing	10%	0	Improver	Improver	Improver	0	10%
Renewal	Renewal	Renewal	10%	20%					

Table 1.4: Cost Increase of the rotations.

The percentage values shown in Table 1.4 are based on the experience and primary data collected by surveys involving farmers and technicians in the Po Valley (located in the north of Italy) [53]. However, these values closely depend on the peculiarity of the single land plot and on choices applied to previous crop cultivation and land allocation. As a consequence these values can be adapted according to specific information collected in other real contexts. In practice, the single farmer could select the cost increase according to own specific experience about arable land fields.

Regarding the CAP payments, in the CAP 2014–2020 scenario the payments are set at €320/ha. For the CAP 2021–2023 and CAP 2023–2027 scenarios, the payments are set at €167/ha for GAEC 7, €110/ha for Eco-scheme 4, and €500/ha for Eco-scheme 5. The CdM payment is included in the revenue of the CdM wheat crop.

The experiments, except for the matheuristic CGH, were conducted on an ASUS ExpertBook laptop with a 1.70 GHz processor and 16 GB of RAM. For the Column Generation-based matheuristic (CGH), a MacBook Pro equipped with an Apple M1 Pro chip and 16 GB of RAM was used. The graph G and the Dynamic Programming-based heuristic (DPH) were implemented in Python 3.9, and Gurobi 11.0.0 was employed to solve $\bar{G}$-*ILP* and *G-ILP*. The CGH and the Subgradient Algorithm were instead implemented and executed in the Julia language.

1.9.2 ILP vs *G*-ILP

Scenario "Pure Farmer"

This section presents the results for the scenario Pure Farmer, comparing two formulations: ILP (introduced in [10]) and *G*-ILP (as modeled in 1.8.2.1), on the set of real-world instances Set_{real}. Both approaches certified optimality within relatively short computational times, although their performances vary considerably with the size of the instances.

Table 1.5 summarizes the results. Each row corresponds to one instance, with columns 1–3 reporting the instance Id, number of plots in the plot class H_1, and number of crops, respectively. Column 4 provides the optimal objective value (in €) over the five-year planning horizon. Columns 5–6 report the solution times and branch-and-bound (B&B) nodes explored by ILP. Column 7 gives the time required to build the graph, while column 8 reports the Gurobi solution time for solving the *G*-ILP model. Column 9 shows the total time, i.e., the sum of the graph construction and the Gurobi solution times.

The results highlight a clear advantage for *G*-ILP: the average computational time is 5.85 seconds compared to 13.11 seconds for ILP. Moreover, *G*-ILP consistently proves always optimality at the root node, whereas ILP does so in most instances but requires node exploration in 5 cases.

Instances Set_{real}				ILP		*G*-ILP			
Id	Plots	Crops	Solution Value (€)	Time (s)	B&B nodes	Time Graph (s)	Time Gurobi (s)	Total time (s)	B&B nodes
TO1	19	5	78953.93	0.09	1	0.03	0.11	0.13	1
FE3	23	7	89596.04	0.50	1	1.41	0.63	2.03	1
FE4	27	9	102788.35	3.26	23	5.62	1.23	6.85	1
PR1	29	10	198626.22	0.57	1	13.41	0.68	14.10	1
CN2	33	10	103718.36	0.34	1	16.97	1.44	18.42	1
RO1	49	6	70545.30	0.18	1	0.09	0.06	0.15	1
MI2	51	7	136158.78	0.25	1	0.32	0.11	0.43	1
MI1	53	8	103948.90	0.34	1	1.13	0.20	1.32	1
AL3	54	8	127342.8	9.28	2604	0.45	0.19	0.64	1
CN1	62	7	272821.70	0.34	1	0.20	0.12	0.32	1
MN2	66	9	210945.24	21.53	1161	1.86	1.10	2.96	1
BO3	72	7	186914.88	0.44	1	0.22	0.11	0.33	1
AL4	84	5	289413.6	2.17	1	0.03	0.03	0.06	1
MN3	84	10	1187961.57	2.04	1	2.98	1.19	4.17	1
MN1	86	10	984338.56	2.97	1	18.15	1.50	19.66	1
FE2	89	9	293788.11	0.91	1	1.07	0.39	1.46	1
AL2	120	7	607236.00	2.42	1	0.21	0.12	0.33	1
RO2	134	8	2784483.82	1.28	1	0.47	0.18	0.65	1
AL1	154	10	1141090.72	4.72	1	2.30	0,81	3.10	1
BO1	161	10	494902.73	152.57	3636	15.34	0.73	16.07	1
FE1	177	9	681327.87	2.38	1	1.23	0.82	2.06	1
VEN1	211	7	819954.44	3.86	1	0.81	0.24	1.05	1
BO2	229	10	673816.47	89.04	1825	36.75	1.55	38.30	1
			Average	13.11	402	5.26	0.59	5.85	1

Table 1.5: Comparison between ILP and *G*-ILP for the scenario "Pure Farmer".

Scenario "CAP 2014–2020"

In this section, we present the results of the experiments conducted on *G*-ILP, introduced in Section 1.8.2.1, and compare them with those of ILP, presented in [10]. ILP employs assignment variables to allocate crops to specific periods and plots, whereas *G*-ILP utilizes flow variables associated with the arcs of the graph G that underpins the model. ILP

encompasses both "horizontal constraints" -which govern the sequences of crops on a single plot, such as replanting and duration constraints- and "vertical constraints" that apply across the plots of the entire farmland, such as the minimum/maximum number of plots per crop and CAP 2014–2020 constraints, as well as the management of rotation schemes. In contrast, G-ILP focuses solely on the "vertical constraints", as the constraints related to crop sequences are already incorporated within the structure of G. Moreover, the number of plots in the instance does not affect the size of G-ILP, whereas the number of variables in ILP is polynomially dependent on this value. The differences between the two ILP models are reflected in the results shown in Table 1.6. Each row in the table reports the results for one of the real-world instances of the problem. Columns 1-3 list the instance Id, number of plots in the plot class H_1 and number of crops, respectively. Column 4 reports the optimal solution value for each instance (in €) over the five-years planning horizon. Columns 5-6 and 7-10 display the computational time (in seconds) and the number of B&B nodes explored by ILP and G-ILP, respectively. The results indicate that model G-ILP significantly outperforms ILP, with an average computational time of 6.87 seconds compared to 166.21 seconds of ILP. Furthermore, G-ILP consistently certifies optimality at the root node, while ILP explores an average of 1910 nodes, suggesting that the linear relaxation of G-ILP is stronger than that of ILP. Such a fact is coherent with other works in the literature [54], and confirms that large size arc-flow models provide stronger relaxations than compact MILP models, as the one proposed in [10]. Column 5 indicates that, on average, the computation time for ILP escalates with the number of crops and, for the same number of crops, increases with the number of plots. In contrast, the computational time for G-ILP is influenced only by the number of crops, as expected. It is notable that the performance of the two models is similar on instances with a small number of plots, with ILP outperforming G-ILP in a few cases. Nevertheless, the dependence on the number of plots causes a significant increase in computational time for ILP, which is not observed with G-ILP. In fact, the computational time ranges from 0.09 to 1200 seconds for ILP, while, for G-ILP, it remains fairly consistent, ranging from 0.02 to 41.81 seconds, and is mainly influenced by the number of crops.

It is worth noting that crop duration—through the setting of $R_{\max}$—has a substantial impact on computational time. For instance, MI2 and BO3 feature the same number of crops but differ in the number of plots (BO3 includes 72 plots, whereas MI2 includes 51). Nevertheless, G-ILP solve BO3 faster than MI2. This behaviour is explained by the presence of Alfalfa in the crop set of MI2, which has duration $d_c = 8$ periods (i.e., four years). In MI2, this implies $\tau = 4$, since $R_{\max} = 4$ and $k \leq R_{\max}$. As shown in Lemma 1.8.3 (Section 1.8.2.1), the number of nodes depends on τ; consequently, larger values of τ increase the size of G and, in turn, the computational effort.

These results suggest that G-ILP is a more reliable and efficient formulation to solve CRP-k problems, especially for instances with a large number of plots. It is also worth mentioning that while both models can be adapted to address the problem CRP-k for $k \neq 3$, adapting G-ILP is considerably easier than modifying ILP. Specifically, for ILP, modifications would be required to the constraints of the problem, albeit based on the structure of the current constraints. Conversely, for G-ILP, it would suffice to adjust the parameter τ in G.

Instances Set_{real}				ILP		*G*-ILP			
Id	Plots	Crops	Solution Value (€)	Time (s)	B&B nodes	Time Graph (s)	Time Gurobi (s)	Total Time (s)	B&B nodes
TO1	19	5	105443.43	0.30	1	0.02	0.02	0.05	1
FE3	23	7	124126.67	0.27	1	0.18	0.13	0.31	1
FE4	27	9	160616.45	3.54	173	2.70	1.70	4.40	1
PR1	29	10	202644.52	3.35	23	23.80	1.58	25.37	1
CN2	33	10	163665.70	2.74	1	34.00	7.81	41.81	1
RO1	49	6	164796.70	21.03	4314	0.51	0.48	0.99	1
MI2	51	7	269264.29	35.92	5189	2.12	0.76	2.89	1
MI1	53	8	245462.05	24.30	4815	2.62	0.51	3.13	1
AL3	54	8	231957.19	78.14	4022	0.60	0.43	1.03	1
CN1	62	7	447262.58	7.49	1	0.23	0.21	0.44	1
MN2	66	9	365743.48	60.24	3574	2.06	3.50	5.56	1
BO3	72	7	312660.16	4.91	207	0.23	0.17	0.40	1
AL4	84	5	385550.48	4.40	1	0.02	0.05	0.07	1
MN3	84	10	1483616.80	133.37	3529	3.10	3.18	6.27	1
MN1	86	10	1164167.92	18.23	208	18.53	3.07	21.60	1
FE2	89	9	418549.48	18.56	400	1.09	0.93	2.03	1
AL2	120	7	1142377.44	87.16	3578	0.22	0.17	0.38	1
RO2	134	8	2853565.51	22.13	1	0.48	0.39	0.87	1
AL1	154	10	1661735.32	344.62	2529	2.17	2.01	4.18	1
BO1	161	10	1060005.41	1200.38	2587	15.84	1.28	17.11	1
FE1	177	9	1032126.09	481.60	2711	1.18	0.85	2.04	1
VEN1	211	7	1724085.23	455.63	2737	0.33	0.12	0.45	1
BO2	229	10	1620454.55	814.47	3344	15.49	1.15	16.64	1
			Average	166.21	1910	5.54	1.33	6.87	1

Table 1.6: Comparison between ILP and *G*-ILP for the CAP 2014–2020 Scenario.

Scenario "CAP 2014–2020 + CdM"

In this scenario, the two formulations, ILP (introduced in [10]) and *G*-ILP, are compared under the CAP 2014–2020 with the "Carta del Mulino" (CdM) constraints introduced in Section 1.5.2. These constraints impose stricter agronomic and sustainability requirements on crop sequences.

The results, reported in Table 1.7, show that both formulations are able to certify optimality across all real-world instances. However, their performance differs substantially. On average, the ILP requires 252.72 seconds and explores around 7111 branch-and-bound nodes, leading to considerably higher computational times. By contrast, the *G*-ILP certifies optimality at the root node. As shown in column 7, most of the computational effort in the *G*-ILP is spent on construction of the graph G, which, on average, is higher than in the scenario without the CdM constraints. This increase is consistent with the fact that the CdM are sequence-based constraints and are embedded in the graph. Meanwhile, on average, the time required by Gurobi to solve *G*-ILP is the same as in the scenario without the CdM constraints. This is expected, since the "vertical constraints" (i.e., management requirements 2–3, CAP *Crop Diversification* and EFA) are identical. The average solution time of *G*-ILP remains significantly lower than that of ILP, highlighting the robustness and effectiveness of the graph-based formulation also in the CdM scenario. Further experiments on scenarios incorporating the CdM constraints are not reported, as this analysis is sufficient to show that such constraints can be embedded in the *G*-ILP model.

Given the computational performance of *G*-ILP discussed in this section and its flexibility in handling different constraints, we use it as the modeling framework for the CAP-driven scenarios presented in Section 1.7.

Instances Set$_{real}$				ILP		*G*-ILP			
Id	Plots	Crops	Solution Value (€)	Time (s)	B&B nodes	Time Graph (s)	Time Gurobi (s)	Total Time (s)	B&B nodes
TO1	19	5	105443.43	0.34	1	0.04	0.05	0.09	1
FE3	23	7	127190.99	2.71	75	1.15	0.16	1.31	1
FE4	27	9	165734.45	26.06	4335	24.72	2.66	27.39	1
PR1	29	10	204162.52	16.10	5459	5.44	0.44	5.88	1
CN2	33	10	165217.70	27.13	2386	6.04	1.57	7.61	1
RO1	49	6	164796.70	97.55	25967	0.26	0.09	0.35	1
MI2	51	7	269321.17	33.80	2394	0.32	0.14	0.46	1
MI1	53	8	245473.33	122.25	28513	1.13	0.39	1.52	1
AL3	54	8	239396.61	98.98	2663	2.95	0.47	3.42	1
CN1	62	7	449903.96	8.37	1	0.82	0.20	1.02	1
MN2	66	9	371402.00	1200.81	64196	12.54	2.35	14.88	1
BO3	72	7	314693.36	4.84	1	0.86	0.18	1.04	1
AL4	84	5	390679.91	4.14	1	0.06	0.04	0.10	1
MN3	84	10	1479627.41	431.65	2740	18.15	2.00	20.15	1
MN1	86	10	1169513.56	161.08	2909	14.84	2.51	17.35	1
FE2	89	9	432725.47	251.41	2606	10.71	0.86	11.56	1
AL2	120	7	1148803.20	55.54	371	0.82	0.19	1.01	1
RO2	134	8	2853565.51	61.15	2858	3.66	0.90	4.56	1
AL1	154	10	1661735.32	526.29	2853	77.05	4.48	81.54	1
BO1	161	10	1065047.43	539.56	4876	32.02	2.25	34.27	1
FE1	177	9	1032126.09	538.32	2874	25.44	1.72	27.16	1
VEN1	211	7	1728254.67	400.52	3271	0.73	0.27	1.01	1
BO2	229	10	1644302.89	1200.61	2225	28.60	1.63	30.24	1
			Average	252.57	7111	11.67	1.11	12.78	1

Table 1.7: Comparison between ILP and *G*-ILP for the CAP 2014–2020 + CdM Scenario.

1.9.3 *G*-ILP performance over different CAP scenarios

Scenarios "CAP 2021–2023" and "CAP 2023–2027"

In this section, *G*-ILP is tested on different scenarios introduced by the CAP regulations, namely CAP 2021-2023, CAP 2023-2027 - Diversification, and CAP 2023–2027 - Rotation, as described in Section 1.5.1. The set of instances considered in the following experiments is extended by two additional crops: the fallow crop c_{fallow} and one crop $c \in C_{flower}$, which are included in all instances. This ensures compliance with GAEC 8 for CAP 2021–2023 and Eco-scheme 5, both Level 1 and Level 2, for CAP 2023–2027.

Table 1.8 reports the results for the CAP 2021–2023 scenario. Under this scenario, 4% of the land must be allocated to the fallow crop c_{fallow} as a mandatory requirement, and Eco-schemes 4 and 5 are evaluated.

Table 1.9 presents the results for the CAP 2023–2027 - Diversification scenario. This scenario introduces Constraints (1.25)–(1.38). These constraints increase the computational difficulty of the model, mainly due to the use of big-M formulations. Note also that this scenario includes the *Crop Diversification* Constraints (1.25)–(1.26), which, as previously discussed, further increase problem complexity. As a result, the computational effort required by the solver (Time Gurobi) is higher than in the previously reported scenarios.

Table 1.10 summarizes the results for the CAP 2023–2027 - Rotation scenario. As introduced in 1.5.1, *Crop Rotation* constraints restrict the admissible crop sequences in the graph. Consequently, the set of feasible crop sequences embedded in the graph are less than in the scenarios above, resulting in a smaller graph G. This graph size reduction explains the lower computational times observed on average, compared to the Diversific-

ation scenario. In fact, the Time Graph resembles the CAP 2021–2023 scenario, where the same *Crop Rotation* constraint applies. In all these scenarios, *G*-ILP certifies optimality at the root node, showing the strongness of its linear relaxation also with this set of constraints.
Overall, these results demonstrate the flexibility of the *G*-ILP approach, since sequence-based constraints can be embedded directly into the graph G, while "vertical constraints" only require changes in the definition of constraints within the ILP model.

Instances Set_{real}			G-ILP				
Id	Plots	Crops	Solution Value (€)	Time Graph (s)	Time Gurobi (s)	Time Total (s)	B&B nodes
TO1	19	7	97664.58	0.46	0.56	1.01	1
FE3	23	9	110866.96	6.41	1.38	7.79	1
FE4	27	11	138003.24	74.25	9.31	83.57	1
PR1	29	12	184087.97	26.76	2.98	29.73	1
CN2	33	12	149612.00	29.41	4.99	34.39	1
RO1	49	8	143734.86	1.13	0.46	1.59	1
MI2	51	9	172377.00	2.07	0.64	2.71	1
MI1	53	10	153873.10	7.40	1.32	8.72	1
AL3	54	10	179708.55	12.93	1.03	13.96	1
CN1	62	9	300310.80	5.79	1.23	7.02	1
MN2	66	11	314624.10	41.33	3.79	45.13	1
BO3	72	9	285978.30	4.43	0.78	5.21	1
AL4	84	7	344639.52	0.45	0.23	0.68	1
MN3	84	12	1110609.12	74.61	5.66	80.26	1
MN1	86	12	1007001.96	59.56	10.77	70.34	1
FE2	89	11	386655.82	35.31	1.54	36.84	1
AL2	120	9	736780.80	5.91	0.72	6.62	1
RO2	134	10	2363670.24	12.02	1.16	13.19	1
AL1	154	12	1263084.13	108.69	5.15	113.85	1
BO1	161	12	854942.20	54.85	1.95	56.80	1
FE1	177	11	908961.05	36.51	2.13	38.65	1
VEN1	211	9	1415812.24	1.63	0.48	2.10	1
BO2	229	12	1060714.66	49.51	1.86	51.37	1
Average			594944.05	28.32	2.61	30.94	1

Table 1.8: *G*-ILP results for the CAP 2021–2023 Scenario.

Instances Set$_{real}$			G-ILP				
Id	Plots	Crops	Solution Value (€)	Time Graph (s)	Time Gurobi (s)	Time Total (s)	B&B nodes
TO1	19	7	102332.83	3.36	0.94	4.30	1
FE3	23	9	115512.74	17.79	0.98	18.77	1
FE4	27	11	150493.15	160.49	20.02	180.51	1
PR1	29	12	196639.92	59.01	5.95	64.97	1
CN2	33	12	154659.64	68.58	14.25	82.83	1
RO1	49	8	148001.28	2.98	2.12	5.09	1
MI2	51	9	178527.36	4.63	1.47	6.09	1
MI1	53	10	161422.60	15.49	4.52	20.01	1
AL3	54	10	188144.37	34.11	4.53	38.64	1
CN1	62	9	336483.58	12.89	4.65	17.54	1
MN2	66	11	323475.20	106.23	17.30	123.53	1
BO3	72	9	293935.20	12.08	2.24	14.32	1
AL4	84	7	385829.67	0.88	0.72	1.60	1
MN3	84	12	1255321.76	140.51	16.22	156.73	1
MN1	86	12	1113700.52	126.19	23.09	149.28	1
FE2	89	11	395036.26	85.48	12.51	97.98	1
AL2	120	9	778857.48	12.73	2.65	15.38	1
RO2	134	10	2788542.51	35.80	5.92	41.72	1
AL1	154	12	1344734.40	181.04	15.30	196.34	1
BO1	161	12	897388.55	99.22	13.84	113.05	1
FE1	177	11	947684.25	96.33	14.55	110.89	1
VEN1	211	9	1531985.54	4.30	1.44	5.74	1
BO2	229	12	1108213.97	100.79	7.85	108.64	1
Average			647692.29	60.04	8.39	68.43	1

Table 1.9: *G*-ILP results for the CAP 2023–2027 - Diversification scenario.

1.9.4 CGH vs *G*-ILP

As shown in Section 1.9.2, both the *G*-ILP and ILP find the optimal solutions for all the real-world instances in CAP 2014–2020 scenario, but *G*-ILP proves to be a more efficient and reliable model than ILP, eliminating the dependency of the model size on the number of plots. In other words, *G*-ILP is already a very effective method to support a considerable number of real-world farms in their crop rotation planning decisions. However, the number of nodes and arcs in graph G, and consequently the number of variables in *G*-ILP, still depends on the number of crops of the instance. As detailed in Lemma 1.8.2, the dimension of the graph increases rapidly as the number of crops grows, which in turn affects the generation time of G, used to produce *G*-ILP. Moreover, even if the graph is generated within a reasonable time, the number of flow variables of *G*-ILP would be extremely large, making the solution of the problem impractical. Such an effect is already evident in the results presented in Table 1.6: although all instances in Set$_{real}$ are solved efficiently by *G*-ILP, a comparison between instances with 5 and 10 crops reveals that the computation time required to find the optimal solution can increase by up to 153 times. Even though the real-world instances in Set$_{real}$ have a manageable number of crops, many farms may potentially cultivate a larger variety. However, preliminary experiments showed that, for most of the larger instances of Set$_{+5}$ and Set$_{+10}$, *G*-ILP was unable to find the optimal solution within a reasonable timeframe. Consequently, the CGH algorithm introduced in Section 1.8.3.2 was used to solve the larger instances. To validate the algorithm, CGH was first tested on Set$_{real}$ and its results were compared

Instances Set_{real}			G-ILP				
ID	Plots	Crops	Solution Value (€)	Time Graph (s)	Time Gurobi (s)	Time Total (s)	B&B nodes
TO1	19	7	102324.31	1.81	0.74	2.54	1
FE3	23	9	115493.81	5.71	0.85	6.56	1
FE4	27	11	146273.80	74.49	7.65	82.15	1
PR1	29	12	193809.32	26.35	1.53	27.89	1
CN2	33	12	158096.60	27.95	4.57	32.52	1
RO1	49	8	148425.90	1.15	0.48	1.63	1
MI2	51	9	180718.50	2.05	0.53	2.57	1
MI1	53	10	162042.73	7.56	0.85	8.41	1
AL3	54	10	188251.56	13.24	1.03	14.27	1
CN1	62	9	309425.88	5.76	0.73	6.49	1
MN2	66	11	323782.80	41.87	4.22	46.09	1
BO3	72	9	297234.72	4.35	0.57	4.92	1
AL4	84	7	355170.48	0.48	0.22	0.70	1
MN3	84	12	1140389.40	73.92	5.03	78.95	1
MN1	86	12	1032238.68	59.37	9.39	68.77	1
FE2	89	11	401623.18	34.93	1.18	36.11	1
AL2	120	9	755505.60	5.78	0.61	6.39	1
RO2	134	10	2421081.18	12.02	0.94	12.96	1
AL1	154	12	1301782.02	110.70	4.27	114.97	1
BO1	161	12	884751.35	54.78	2.43	57.21	1
FE1	177	11	945592.41	42.37	1.91	44.29	1
VEN1	211	9	1455484.33	1.76	0.57	2.33	1
BO2	229	12	1099827.46	52.10	2.02	54.12	1
Average			613883.74	28.72	2.27	30.99	1

Table 1.10: *G*-ILP results for the CAP 2023–2027 - Rotation scenario.

with those of model *G*-ILP. These experiments are detailed in Table 1.11, where, for convenience, the instance data, optimal solution values, and performance metrics of *G*-ILP are reported in Columns 1-6. Columns 7-9 display the performance of CGH. Specifically, Column 7 reports the computational time and Column 8 shows the gap between the final solution of CGH and the initial solution found on the restricted graph $\bar{G}$ generated with the heuristic $\bar{G}$-Start. Finally, Column 9 shows the gap between the solution of the linear relaxation of the model resulting at the end of the column generation procedure and the corresponding integer solution found by Gurobi on the same model. These experiments demonstrated that the column generation based matheuristic CGH can efficiently find the optimal solution for all real-world instances of Set_{real}. Notably, the gap between the final and initial solutions is relatively small, averaging 3.41%, indicating that $\bar{G}$-Start produces a graph $\bar{G}$ which allows to find good quality solutions. As shown in the last column of Table 1.11, the gap between the final integer solution and its linear relaxation is very small, averaging only 0.17%, which confirms the strength of the linear relaxation of *G*-ILP.

CGH was also tested on instances from Set_{+5} and Set_{+10}, and compared to the results of *G*-ILP on those instances. For these experiments, a time limit of 1200 seconds was set for *G*-ILP, including the graph generation, and a maximum of 400 iterations was also set for CGH. Note that the time needed by $\bar{G}$-Start to generate the initial reduced graph is negligible and included in the computational time of CGH. The results for Set_{+5} and Set_{+10} are reported in Tables 1.12 and 1.13, respectively, showcasing the performance

Instances Set$_{real}$				G-ILP		Column Generation Heuristic CGH		
Id	Crops	Plots	Solution Value (€)	Time (s)	B&B nodes	Time (s)	Gap $\tilde{G}$-Start/CGH (%)	Gap LP/ILP (%)
TO1	5	19	105443,43	0,09	1	10.71	1.61	0.09
AL4	5	84	385550.48	0.04	1	0.12	8.66	0.09
RO1	6	49	164796.70	0.12	1	9.45	0.83	0.11
FE3	7	23	124126.67	0.32	1	0.27	0.00	0.01
MI2	7	51	269264.29	0.42	1	1.33	2.24	0.16
CN1	7	62	479148.50	0.29	1	0.14	1.81	0.68
BO3	7	72	312660.16	0.27	1	0.22	1.19	0.14
AL2	7	120	1142377.44	0.27	1	9.51	3.85	0.00
VE1	7	211	1724085.23	0.39	1	0.39	4.02	0.06
MI1	8	53	245462.05	1.36	1	0.70	0.97	0.07
AL3	8	54	231957.19	0.64	1	0.90	0.33	0.28
RO2	8	134	2853565.51	0.65	1	0.37	10.52	0.36
FE4	9	27	160616.45	3.84	1	16.48	1.85	0.11
MN2	9	66	365743.48	3.71	1	17.67	2.33	0.03
FE2	9	89	418549.48	1.58	1	0.59	2.79	0.10
FE1	9	177	1032126.09	1.56	1	0.62	0.65	0.01
PR1	10	29	202644.52	10.04	1	12.22	4.72	0.57
CN2	10	33	163665.70	13.91	1	28.75	3.57	0.26
MN3	10	84	1483616.80	4.61	1	12.08	7.61	0.13
MN1	10	86	1164167.92	13.53	1	5.95	10.55	0.39
AL1	10	154	1661735.32	3.47	1	0.71	3.76	0.12
BO1	10	161	1060005.41	10.41	1	3.84	2.50	0.11
BO2	10	229	1620454.55	10.26	1	2.61	2.06	0.03
			Average	3.56	1	5.90	3.41	0.17

Table 1.11: Comparison between CGH and G-ILP on Set$_{real}$.

of each approach on larger instances. Similarly to previous tables, in Tables 1.12 and 1.13, Columns 1-3 report the Id and dimensions of each instance, Columns 4-6 display the performance of model G-ILP and Columns 7-10 show the performance of CGH.

Instances Set$_{+5}$			G-ILP			Column Generation Heuristic CGH			
Id	Crops	Plots	Solution Value (€)	Time (s)	B&B nodes	Solution Value (€)	Time (s)	Gap $\tilde{G}$-Start/CGH (%)	Gap LP/ILP (%)
TO1$_{+5}$	10	19	206255.25	10.72	1	206255.25	11.09	5.51	0.27
AL4$_{+5}$	10	84	-	1200	-	415504.86	1.69	4.35	0.25
RO1$_{+5}$	11	49	-	1200	-	314157.04	13.25	6.64	0.65
FE3$_{+5}$	12	23	249462.54	46.89	1	249462.54	1.15	4.55	0.68
MI2$_{+5}$	12	51	-	1200	-	551090.28	16.89	3.66	0.33
CN1$_{+5}$	12	62	-	1200	-	479148.50	5.34	1.81	0.68
BO3$_{+5}$	12	72	-	1200	-	380014.32	4.87	3.01	0.02
AL2$_{+5}$	12	120	-	1200	-	1142377.44	5.56	3.85	0.00
VE1$_{+5}$	12	211	-	1200	-	2307872.38	16.78	2.88	0.08
MI1$_{+5}$	13	53	-	1200	-	355289.98	7.98	1.65	0.22
AL3$_{+5}$	13	54	-	1200	-	232704.79	8.59	0.71	0.25
RO2$_{+5}$	13	134	2983682.38	109.30	1	2983682.38	1.65	7.50	0.28
FE4$_{+5}$	14	27	197124.80	272.03	1	197124.80	345.06	3.18	0.05
MN2$_{+5}$	14	66	-	1200	-	366886.72	150.94	1.71	0.06
FE2$_{+5}$	14	89	1522274.20	272.03	1	1522274.20	4.15	4.31	0.40
FE1$_{+5}$	14	177	1916561.88	280.11	1	1916561.88	2.51	3.14	0.09
PR1$_{+5}$	15	29	-	1200	-	225620.98	19.05	1.53	0.79
CN2$_{+5}$	15	33	-	1200	-	207092.90	225.87	2.46	0.57
MN3$_{+5}$	15	84	-	1200	-	1483616.80	43.01	7.61	0.13
MN1$_{+5}$	15	86	-	1200	-	1164167.92	66.01	10.55	0.39
AL1$_{+5}$	15	154	1661735.32	612.97	1	1661735.32	4.25	3.53	0.12
BO1$_{+5}$	15	161	-	1200	-	1748566.14	31.98	3.69	0.20
BO2$_{+5}$	15	229	-	1200	-	1634858.33	641.14	4.41	0.03
			Average	904,52	1		70,82	4,01	0.28

Table 1.12: Comparison between CGH and G-ILP on instances of Set$_{+5}$.

Table 1.12 illustrates that, on Set$_{+5}$ (having a minimum of 10 crops per instance) G-ILP fails to find the optimal solution within the time limit for most cases. Specifically, the

optimal solution is found, at the root node, in only 7 out of 23 instances, with an average computation time of 229.15 seconds for these instances. Conversely, CGH successfully finds solutions for all instances with an average computational time of 70.82 seconds. Notably, the algorithm achieves the optimum in all instances where G-ILP certified it, with an average time of 144.42 seconds, thus outperforming G-ILP on those instances. The overall average computational time remains at 70.82 seconds, and all solutions are considered to be at least of high quality, given that the model consistently finds the optimal solutions in all instances solvable by G-ILP. Furthermore, the average gap between the initial solution produced by solving G-ILP on the graph generated by $\bar{G}$-Start and the final solution provided by CGH is relatively small, about 4%, and the gap for the linear relaxation is only 0.28%.

Table 1.13 reports the results on the largest instances of Set_{+10}. As the results show, the G-ILP model manages to solve only the smallest instance (TO1_{+10}) to optimality, requiring 430.84 seconds at the root node. In contrast, CGH successfully finds the optimal solution for TO1_{+10} in 79 seconds and, generally, solutions for all instances of Set_{+10} with an average computational time of around 660 seconds. The gaps between the initial and final solutions, as well as between the linear relaxation and the integer solutions, remain consistent with those observed in other cases. This consistency suggests that solving G-ILP on the graph generated by $\bar{G}$-Start is already quite effective and that the strength of the linear relaxation of G-ILP is not compromised by an increased number of crops.

In conclusion, the proposed solution methods, G-ILP, $\bar{G}$-Start, and CGH, effectively address crop rotation planning problems with sustainability constraints in real-world contexts. As show in Section 1.10, these methods can be successfully integrated into decision-support systems to assist farmers with complex decisions, including crop-mix selection, evaluation of what-if scenarios, adherence to sustainability initiatives, and optimal production planning.

As already specified in Section 1.9.1, both the tests on G-ILP and column generation based matheuristic (CGH) algorithm were made with a MacBook Pro equipped with an Apple M1 Pro chip and 16 GB of RAM.

Instances Set $_{+10}$			G-ILP			Column Generation Heuristic CGH			
Id	Crops	Plots	Solution Value (€)	Time (s)	B&B nodes	Solution Value (€)	Time (s)	Gap $\bar{G}$-Start/CGH (%)	Gap LP/ILP (%)
$TO1_{+10}$	15	19	210751.01	430.84	1	210751.01	79.09	5.30	0.26
$AL4_{+10}$	15	84	-	1200	-	485014.28	21.55	3.20	0.03
$RO1_{+10}$	16	49	-	1200	-	317790.04	47.19	4.12	0.60
$FE3_{+10}$	17	23	-	1200	-	269894.58	599.84	4.75	1.76
$MI2_{+10}$	17	51	-	1200	-	556634.64	92.64	4.34	0.18
$CN1_{+10}$	17	62	-	1200	-	771871.84	705.64	7.99	0.35
$BO3_{+10}$	17	72	-	1200	-	472910.66	478.37	1.68	0
$AL2_{+10}$	17	120	-	1200	-	1172721.72	60.71	3.58	0
$VE1_{+10}$	17	211	-	1200	-	2408175.86	993.16	8.81	0.05
$MI1_{+10}$	18	53	-	1200	-	368713.28	101.25	2.62	0.20
$AL3_{+10}$	18	54	-	1200	-	578703.42	86.26	4.01	0.40
$RO2_{+10}$	18	134	-	1200	-	2994038.88	20.87	8.20	0.26
$FE4_{+10}$	19	27	-	1200	-	206917.58	1238.16	0.37	0.38
$MN2_{+10}$	19	66	-	1200	-	490295.40	1304.37	1.83	0.07
$FE2_{+10}$	19	89	-	1200	-	1533814.19	13.98	5.02	0.31
$FE1_{+10}$	19	177	-	1200	-	2083112.86	1399.80	7.05	0.07
$PR1_{+10}$	20	29	-	1200	-	291303.65	1270.58	5.39	-0.09
$CN2_{+10}$	20	33	-	1200	-	235060.67	1266.57	3.70	0.11
$MN3_{+10}$	20	84	-	1200	-	1486369.30	454.80	8.39	0.10
$MN1_{+10}$	20	86	-	1200	-	1180758.12	993.84	10.59	0.22
$AL1_{+10}$	20	154	-	1200	-	1779017.84	1313.94	4.15	0.10
$BO1_{+10}$	20	161	-	1200	-	1873243.24	1295.97	4.74	0.15
$BO2_{+10}$	20	229	-	1200	-	2616439.40	1325.70	4.79	0.08
			Average	1166.56	1		659.32	4.98	0.24

Table 1.13: Comparison between CGH and G-ILP on instances of Set$_{+10}$.

1.9.5 DPH vs G-ILP

Hereafter, the results for the Dynamic Programming heuristic (DPH) introduced in Section 1.8.3.3 are reported.

Tables 1.14 and 1.15 report the comparison between DPH and G-ILP under the CAP 2023–2027 and "Carta del Mulino" (CdM) constraints. The comparison focuses on both the objective function values, i.e., the overall profit (€) and computational time (seconds). Overall, DPH produces feasible and high quality solutions with significantly reduced computational effort.

Columns 8 show the percentage gap between the objective value obtained by DPH compared to the optimal solution. Instead, Columns 9 face the percentage gap, in terms of runtime, between DPH and G-ILP.
In Diversification scenario, DPH produce solutions with a value of 5.01% lower than the optimal solution found by G-ILP, on average. Meanwhile, in Rotation scenario, the same average relative gap is only 0.25%. In term of computational time, DPH achieves an average reduction of over 73% and 91% in the Diversification and Rotation scenarios, respectively.
Note that, on instances featuring a second semester crop, the gap is sensibly worse than the average in both scenarios. This is because second semester crops are subject to a maximum land surface constraint (i.e., a "vertical constraint"), whereas DPH is designed to handle sequence-based constraints. On the other hand, on larger instances DPH seems to be particularly faster than G-ILP, and able to find good feasible solutions. Notably, the heuristic is able to find the optimal solution in almost all instances in the CAP 2023–2027 - Rotation scenario (Table 1.15). Figure 1.10 presents a visual comparison of the solution quality and CPU time of DPH and G-ILP across all instances, under both the Diversification and Rotation scenarios. The figure clearly illustrates that the significant

reduction in computational time achieved by DPH more than compensates its marginal loss in solution quality.

These results highlight that DPH represents a valuable trade-off between solution quality and computational effort, especially in practical settings where timely decisions are needed, and exact methods may be computationally too expensive.

Instances Set_{real}			DPH		*G*-ILP		GAP (%)	
Id	Crop	Ha	Value (€)	Time (s)	Value (€)	Time (s)	$\frac{\text{G-ILP}-\text{DPH}}{\text{G-ILP}}$	$\frac{\text{T}_{\text{G-ILP}}-\text{T}_{\text{DPH}}}{\text{T}_{\text{G-ILP}}}$
TO1	5	19	98776.13	1.12	102332.83	5.20	3.45	78.4
FE3	7	23	118527.41	2.81	118527.41	13.32	0.00	78.9
FE4	9	27	143736.73	6.43	152096.15	168.76	5.50	96.2
PR1	10	29	187715.00	8.99	197786.16	55.75	5.09	83.9
CN2	10	33	149252.37	12.31	157233.24	74.18	5.08	83.4
RO1	6	49	146960.64	3.56	148001.28	6.13	0.70	41.9
MI2	7	51	176098.45	5.99	178527.36	6.96	1.36	14.0
MI1	8	53	160820.23	9.73	161422.60	20.23	0.37	51.9
AL3	8	54	193367.57	9.04	195908.17	43.93	1.30	79.4
CN1	7	62	318843.74	6.09	341055.76	17.26	6.51	64.7
MN2	9	66	316350.28	14.35	331694.36	121.32	4.63	88.2
BO3	7	72	290535.88	6.06	294098.40	15.74	1.21	61.5
AL4	5	84	362589.50	1.99	386010.62	1.58	6.07	26.3
MN3	10	84	1051241.07	22.54	1254182.27	200.46	16.19	88.8
MN1	10	86	961433.14	24.90	1119622.70	153.61	14.14	83.8
FE2	9	89	393033.26	7.33	395036.26	128.81	0.51	94.3
AL2	7	120	720207.60	2.52	786214.08	19.89	8.39	87.3
RO2	8	134	2488763.25	3.69	2788542.51	43.17	10.75	91.5
AL1	10	154	1240966.02	8.91	1344734.40	218.99	7.72	95.9
BO1	10	161	861805.21	9.00	900773.57	119.09	4.33	92.4
FE1	9	177	942886.53	5.60	947684.25	138.44	0.51	96.0
VEN1	7	211	1372498.08	2.53	1532769.40	6.17	10.45	59.0
BO2	10	229	1128230.56	9.54	1140418.55	161.08	1.07	94.1
Avg			601071.24	8.04	651072.71	75.70	5.01	73.00

Table 1.14: DPH vs *G*-ILP in CAP 2023–2027 - Diversification + CdM

1.9.6 Lagrangian relaxation vs *G*-ILP

In this section, we report the computational results of the Lagrangian relaxation, introduced in Section 1.8.3.1, and compare them with *G*-ILP. Both models were tested on the same set of real-world instances, under the constraints introduced by CAP 2014–2020.

Both the models rely on flow variables defined on the graph G representing feasible crop sequences, the Lagrangian method employs a relaxation of the "vertical constraints" (i.e., maximum and minimum land surface constraints and CAP constraints) to obtain an upper bound on the optimal solution value. Unlike *G*-ILP, which directly enforces all vertical constraints in the model, the Lagrangian relaxation approach solves sequentially a min cost flow problem through a subgradient algorithm, iteratively tightening the bounds.

Table 1.16 presents a detailed comparison between *G*-ILP and Lagrangian relaxation. Columns 1–3 list the instance Id, number of plots in the plot class H_1, and number of crops, respectively. Column 4 reports the optimal solution value obtained by *G*-ILP in € over the five-years planning horizon, and Column 5 provides the total computational time for *G*-ILP, including graph generation and ILP solving. Columns 6–9 summarize the Lagrangian relaxation results, including the upper bound value (Column 6), the time

			DPH		*G*-ILP		GAP (%)	
Id	Crop	Ha	Value (€)	Time (s)	Value (€)	Time (s)	$\frac{\text{G-ILP}-\text{DPH}}{\text{G-ILP}}$	$\frac{\text{T}_{\text{G-ILP}}-\text{T}_{\text{DPH}}}{\text{T}_{\text{G-ILP}}}$
TO1	5	19	102324.31	0.21	102324.31	0.88	0.00	75.90
FE3	7	23	119237.75	0.87	119237.75	5.66	0.00	84.58
FE4	9	27	148914.10	2.73	149051.90	154.06	0.09	98.23
PR1	10	29	193809.32	2.36	193809.32	62.75	0.00	96.24
CN2	10	33	161095.40	2.01	161095.40	59.51	0.00	96.63
RO1	6	49	148425.90	0.40	148425.90	4.51	0.00	91.11
MI2	7	51	180718.50	0.75	180718.50	6.62	0.00	88.72
MI1	8	53	162042.73	1.26	162042.73	17.89	0.00	92.97
AL3	8	54	196220.88	1.00	196220.88	31.35	0.00	96.81
CN1	7	62	320530.70	0.69	320530.70	16.95	0.00	95.91
MN2	9	66	316657.88	2.24	330921.80	50.39	4.31	95.56
BO3	7	72	297234.72	0.67	297234.72	5.75	0.00	88.36
AL4	5	84	355170.48	0.20	355170.48	0.74	0.00	72.82
MN3	10	84	1142511.60	3.60	1145760.60	81.74	0.28	95.60
MN1	10	86	1029739.36	3.36	1040658.40	68.82	1.05	95.11
FE2	9	89	401623.18	1.48	401623.18	39.41	0.00	96.24
AL2	7	120	755505.60	0.66	755505.60	7.01	0.00	90.65
RO2	8	134	2421081.18	0.91	2421081.18	13.60	0.00	93.34
AL1	10	154	1301782.02	2.51	1301782.02	129.69	0.00	98.06
BO1	10	161	884751.35	2.37	884751.35	64.09	0.00	96.30
FE1	9	177	945592.41	1.43	945592.41	48.41	0.00	97.04
VEN1	7	211	1455484.33	0.60	1455484.33	2.16	0.00	72.37
BO2	10	229	1143007.70	2.23	1143007.70	56.02	0.00	96.02
Avg			616672.23	1.50	617914.40	40.35	0.25	91.50

Table 1.15: DPH vs *G*-ILP in CAP 2023–2027 - Rotation + CdM

required to generate the graph (Column 7), the subgradient algorithm time (Column 8), and the total computational time (Column 9), i.e. the sum of the graph generation time and the subgradient algorithm time. Column 10 shows the relative optimality gap (%) between the upper bound obtained through the Lagrangian relaxation and the optimal value of *G*-ILP.

The results clearly show that *G*-ILP consistently computes the exact optimal solution with very low computational effort, with total times ranging from 0.07 to 45.46 seconds. In contrast, the Lagrangian relaxation method, while providing tight upper bounds with gaps below 1.7% in all instances, requires additional computational effort due to the iterative subgradient procedure. Although Lagrangian relaxation can be useful for obtaining high-quality bounds, it is generally less efficient than *G*-ILP. For these computational reasons, the Lagrangian relaxation approach was not pursued further in this study, since *G*-ILP is more efficient and reliable for solving the instances considered.

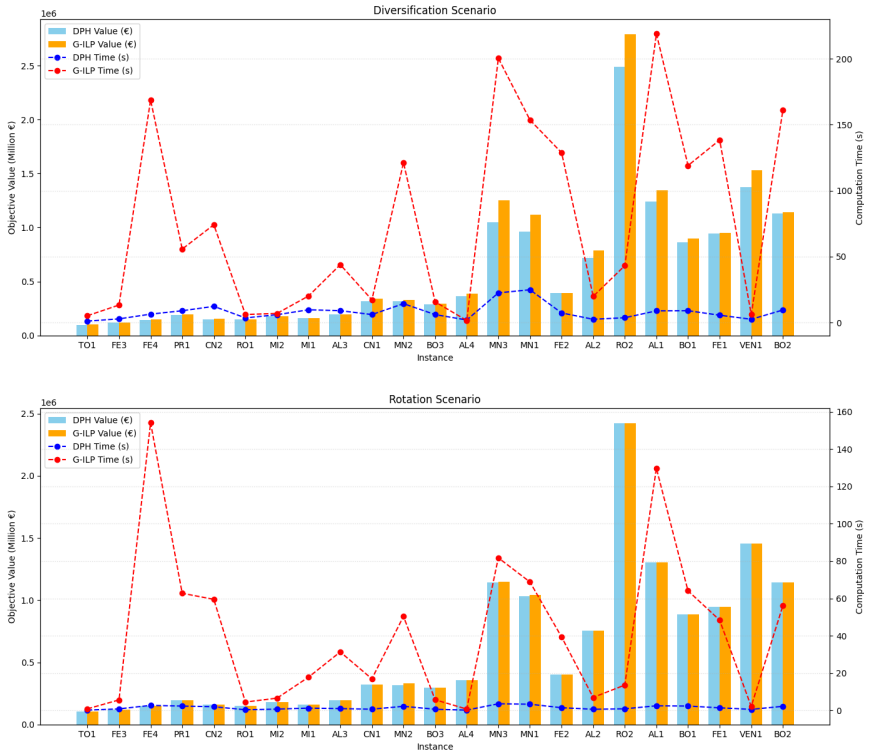

Figure 1.10: Comparison of DPH and *G*-ILP solutions in terms of objective value and computational time in Diversification and Rotation Scenarios.

Instances Set$_{real}$			G-ILP		Lagrangian Relaxation				GAP (%)
Id	Plots	Crops	Solution Value (€)	Time (s)	Upper Bound (€)	Time Graph (s)	Subgradient Time (s)	Total Time (s)	$\frac{UB-G-ILP}{G-ILP}$
TO1	19	5	105443,43	0,05	105443,80	0,06	1,90	1,96	0,01
FE3	23	7	124126,67	0,31	124126,67	0,40	1,84	2,24	0,00
FE4	27	9	160616,45	4,40	160669,06	4,36	32,57	36,93	0,03
PR1	29	10	202644,52	25,37	202828,69	16,39	10,87	27,26	0,09
CN2	33	10	163665,70	41,81	164259,09	19,30	25,53	44,83	0,36
RO1	49	6	164796,70	0,99	164973,56	0,17	2,94	3,11	0,11
MI2	51	7	269264,29	2,89	270245,26	0,48	2,85	3,33	0,36
MI1	53	8	245462,05	3,13	245803,97	1,63	4,99	6,62	0,14
AL3	54	8	231957,19	1,03	235168,32	0,80	7,36	8,16	1,38
CN1	62	7	447262,58	0,44	449623,68	0,35	3,98	4,33	0,53
MN2	66	9	365743,48	5,56	366228,37	3,54	25,70	29,24	0,13
BO3	72	7	312660,16	0,40	313098,94	0,36	3,88	4,24	0,14
AL4	84	5	385550,48	0,07	386018,47	0,06	2,10	2,16	0,12
MN3	84	10	1483616,80	6,27	1492324,75	4,25	33,10	37,35	0,59
MN1	86	10	1164167,92	21,60	1177460,80	45,46	48,49	93,95	1,14
FE2	89	9	418549,48	2,03	419872,47	1,93	28,86	30,79	0,32
AL2	120	7	1142377,44	0,38	1143342,36	0,39	7,86	8,25	0,08
RO2	134	8	2853565,51	0,87	2864047,92	0,83	6,93	7,76	0,37
AL1	154	10	1661735,32	4,18	1689942,89	3,94	39,71	43,65	1,70
BO1	161	10	1060005,41	17,11	1060916,16	21,51	51,23	72,74	0,09
FE1	177	9	1032126,09	2,04	1033190,71	2,17	22,97	25,14	0,10
VEN1	211	7	1724085,23	0,45	1726013,59	0,53	3,08	3,61	0,11
BO2	229	10	1620454,55	16,64	1621077,17	16,45	13,60	30,05	0,04
Average			753907,72	6,87	757246,81	6,32	16,62	22,94	0,35

Table 1.16: Lagrangian relaxation vs *G*-ILP in CAP 2014–2020 Scenario.

1.10 Insights of the model

When applied to a CRP-k instance, the optimization models outlined in Section 1.8 determine the crop plan that maximizes farm profit over the specified time horizon. For each scenario, the models assign a crop to every plot in each period, taking into account the impact of crop rotations on cultivation costs, as well as agronomic and sustainability constraints arising from management requirements and policy instruments (e.g., CAP, CdM).

This section illustrates the insights that can be obtained from the models for the instances described in Section 1.9.1. First, we present two CRP-k examples that describe profitability and land use for two representative farms, showing that CAP-compliant plans can, in some cases, be more profitable than non-compliant ones. We then analyse, for a single instance, how profitability and land use change across the different CAP scenarios. Next, we compare profitability across all instances and scenarios and investigate the main factors driving farm income. Finally, we use the CGH algorithm to show that expanding the crop portfolio can significantly increase farm profits.

CRP-k example: profitability and land-usage under CAP 2014–2020

The models and algorithms presented in Section 1.8 are effective tools for evaluating the economic and environmental implications of sustainability policies, such as the CAP. As an illustrative case, we report a CRP-k instance drawn from Section 1.9.1. Since the instances refer to Italian farms, the rotation length is set to $k = 3$. This example is intended to show how the model can support farmers in assessing the profitability of adopting, or not adopting, the CAP requirements introduced at the EU level, as well as in understanding how these policies may affect farm land use.

Figures 1.11 and 1.12 respectively illustrate the impact of joining or not the CAP 2014–2020 on the 5-year crop rotation plans of two real Italian farms of Set_{real}. Figure 1.11 (a) and Figure 1.11 (b) show the optimal crop rotation plans for instance MN1 with and without the CAP constraints. As shown in Figure 1.11 (a), monoculture schemes dominate when the CAP sustainability constraints are not considered: only two crop sequences are applied to the overall land of 86 hectares, and a single crop is cultivated on the entire arable land in each period. Notably, Tomato is grown continuously for two years (correspondent to its maximum replanting) and then it is replaced by Wheat. The two sequences only differ in the even period 6, where, in sequence 2, Soybean is grown on 32 plots (i.e., hectares). Soybean is not cultivated in sequence 1, since a $max^t_{Soybean}$ restriction of exactly 32 plots exists for this crop. In contrast, when the farm chooses to comply with the CAP 2014–2020 (Figure 1.11 (b)), the optimal crop rotation plan undergoes significant changes. The arable land is diversified with several crops grown each year, including a minimum allocation for nitrogen-fixing crops like Pea. While Tomato, a highly profitable crop, remains extensively cultivated, the *Crop Diversification* and EFA constraints of the CAP prevent unsustainable monoculture schemes. A similar trend is observed in Figure 1.12. When farm BO1 does not participate in the CAP 2014–2020

(Figure 1.12 (a)), the optimal crop rotation plan consists of a simple sequence of Corn and Alfalfa, with intensive cultivation of the same crop across the entire arable land each year. However, when the CAP constraints are applied (Figure 1.12 (b)), five different crops are cultivated over the entire planning horizon, diversifying each year and including Soybean as a nitrogen-fixing crop.

Notably, the results indicate that the economic incentives introduced by the CAP not only compensate for the additional sustainability constraints but also result in a profit increase of 18% and 114% for MN1 and BO1, respectively, upon joining the CAP. Such an analysis, enabled by the models and algorithms presented in this work, provides farmers with valuable insights into the complexity and benefits of participating in sustainability initiatives. Simultaneously, the models and algorithms allow policymakers to predict farmers' behavior, helping to fine-tune incentives and assess the impact of their policies on land use and food security.

		Periods									
Crop sequences	**# plots**	1	2	3	4	5	6	7	8	9	10
1	54	Tomato	Tomato	Tomato	Tomato	Wheat	-	Tomato	Tomato	Tomato	Tomato
2	32	Tomato	Tomato	Tomato	Tomato	Wheat	Soybean	Tomato	Tomato	Tomato	Tomato

Crop		Land Use									
Tomato		86	86	86	86	-	-	86	86	86	86
Wheat		-	-	-	-	86	-	-	-	-	-
Soybean		-	-	-	-	-	86	-	-	-	-

(a) - Optimal crop rotation plan of farm MN1 when it does not join the CAP. The total profit is 984′338.56 €.

		Periods									
Crop sequences	**# plots**	1	2	3	4	5	6	7	8	9	10
1	12	Wheat	Soybean	Tomato	Tomato	Tomato	Tomato	Wheat	Soybean	Tomato	Tomato
2	5	Wheat	Soybean	Tomato	Tomato	Tomato	Tomato	Pea	Pea	Tomato	Tomato
3	17	Tomato	Tomato	Wheat	Soybean	Tomato	Tomato	Tomato	Tomato	Wheat	Soybean
4	5	Tomato	Tomato	Tomato	Tomato	Wheat	-	Tomato	Tomato	Tomato	Tomato
5	32	Tomato	Tomato	Tomato	Tomato	Wheat	Soybean	Tomato	Tomato	Tomato	Tomato
6	5	Tomato	Tomato	Tomato	Tomato	Pea	Pea	Tomato	Tomato	Tomato	Tomato
7	5	Tomato	Tomato	Pea	Pea	Tomato	Tomato	Tomato	Tomato	Pea	Pea
8	5	Pea	Pea	Tomato	Tomato	Tomato	Tomato	Wheat	Soybean	Tomato	Tomato

Crop		Land Use									
Wheat		17	-	17	-	37	-	17	-	17	-
Soybean		-	17	-	17	-	32	-	17	-	17
Tomato		64	64	64	64	44	44	64	64	64	64
Pea		5	5	5	5	5	5	5	5	5	5

(b) - Optimal crop rotation plan of farm MN1 when it joins the CAP. The total profit is 1′164′167.92 €.

Figure 1.11: Impact of the CAP on the crop rotation plan of farm MN1, in the province of Mantova, Italy, with a total arable land of 86 hectares.

		Periods									
Crop sequences	# plots	1	2	3	4	5	6	7	8	9	10
1	161	Corn	Corn	Corn	Corn	Alfalfa	Alfalfa	Alfalfa	Alfalfa	Alfalfa	Alfalfa

Crop	Land Use									
Corn	161	161	161	161	-	-	-	-	-	-
Alfalfa	-	-	-	-	161	161	161	161	161	161

(a) - Optimal crop rotation plan of farm BO1 when it does not join the CAP. The total profit is 494′902.73 €.

		Periods									
Crop sequences	# plots	1	2	3	4	5	6	7	8	9	10
1	9	Corn	Corn	Beet	Beet	Beet	Beet	Wheat	-	Beet	Beet
2	9	Beet	Beet	Wheat	-	Beet	Beet	Beet	Beet	Corn	Corn
3	9	Beet	Beet	Beet	Beet	Wheat	-	Beet	Beet	Beet	Beet
4	70	Beet	Beet	Beet	Beet	Soybean	Soybean	Beet	Beet	Beet	Beet
5	32	Beet	Beet	Soybean	Soybean	Beet	Beet	Beet	Beet	Alfalfa	Alfalfa
6	32	Soybean	Soybean	Beet	Beet	Beet	Beet	Soybean	Soybean	Beet	Beet

Crop	Land Use									
Wheat	-		9	9	9	-	9	9	-	-
Beet	120	120	120	120	82	82	120	120	120	120
Corn	9	9	-	-	-	-	-	-	9	9
Soybean	32	32	32	32	70	70	32	32	-	-
Alfalfa	-	-	-	-	-	-	-	-	32	32

(b) - Optimal crop rotation plan of farm BO1 when it joins the CAP. The total profit is 1′060′005.41 €.

Figure 1.12: Impact of the CAP on the crop rotation plan of farm BO1, in the province of Bologna, Italy, with a total arable land of 161 hectares.

CRP-k example: profitability and land-usage under different CAP scenarios

In what follows, we present an illustrative CRP-k example (with $k = 3$) under the different scenarios induced by CAP constraints. This example helps highlight how CAP policy changes over the years may affect both farm profitability and land use decisions.

Figure 1.13 shows the optimal crop sequences on each plot, of a generic class H_1 with homogeneous plots of 1-hectare, in the five scenarios (Pure Farmer, CAP 2014–2020, CAP 2021–2023, CAP 2023–2027 - Diversification, CAP 2023–2027 - Rotation) for an instance located in the province of Turin, encompassing a total area of 19 hectares (the instance is called "TO1"). For each scenario, the hectares are given on the columns and the seeding periods are reported on the rows (where "I" and "II" are the first and second semester of each of the five years). The last column reports the total expected profit associated with the (most profitable) solution. The color of each block is related to a specific crop, as showed in the legend. For each scenario, the model identifies, from a predefined set of options, the optimal crop to be cultivated in each semester of the planning horizon on each plot. The Pure Farmer scenario (top panel) exhibits a monoculture-oriented approach, with the entire land almost exclusively dedicated to a single crop throughout the planning horizon. Predominantly, high-income crops such as grain maize are cultivated on most hectares. The CAP 2014–2020 scenario (second panel) introduces *Crop Diversification* constraints, which require more varied cropping patterns. Note that, as introduced in Section 1.5.1, since the total farmland surface of this instance is less than 30 hectares, at least two crops must be grown in each seeding period, and no single crop may cover more than 75% of the total area. While grain maize still dominates, there is an increased allocation to soy. The CAP 2021–2023 scenario (middle

panel) shows further diversification, with a significant shift towards set-aside areas (i.e., fallow crops) and soy cultivation. The CAP 2023–2027 - Diversification scenario (fourth panel) continues to emphasize diversification, with a balanced distribution of grain maize and soy across hectares. In the CAP 2023–2027 - Rotation scenario (bottom panel) soy cultivation is more prominent. Furthermore, leveraging the input data, the model estimates the expected profit generated under each scenario over a five-year period. In this TO1 instance, the CAP 2014–2020 exhibits the higher five-year income. Such a result can be justified by the national average value of direct payments of 320 €/ha (see 1.9.1), and since PAC constraints allow a larger portion of the area to be allocated to more profitable crops (soy and maize), similar to the Pure Farmer scenario. It is worth noting that in CAP 2023–2027 - Diversification scenario, the model take care also of the reward of Eco-scheme 4. In fact, since that soy is a leguminous crop, i.e. a crop that belong to C_{E4}, in plots 10-14 the farmer get the award in the two-years period represented by year 1 and 2 and by the year 5. Similarly, in plots 15-19 the farmer get the reward in the two-years period 1-2 and 4-5. In CAP 2023–2027 - Rotation scenario, instead, the reward of Eco-scheme 4 is get in the two-years period represented by year 1-2 and 3-4.

Profitability and computational time comparison across CAP scenarios

Table 1.17 reports the results obtained for G-ILP in all the considered CAP scenarios, along with Pure Farmer scenario. The table summarizes, for each instance, the optimal value of the objective function (expressed in € over the five-years planning horizon) and the corresponding computational time (in seconds) across five different policy settings: Pure Farmer, CAP 2014–2020, CAP 2021–2023, and the two variants of CAP 2023–2027 (Diversification and Rotation).

This table serves a dual purpose: (*i*) it provides a comprehensive overview of the computational performance of the model across scenarios, highlighting both the value achieved and the time required for its resolution; (*ii*) it offers a insight into the potential impact of different CAP policies on a set of representative farms. In this sense, the table can be a useful tool for policy makers, as it allows them to assess how changes in the Common Agricultural Policy may affect farm profitability. It is worth noting that the CAP 2014–2020 scenario leads to the highest average profits across all instances, with a five-year mean of €753907.72, higher than in the other scenarios.

This insight highlights that the model introduced in Section 1.8, and in particular G-ILP, is able to provide an overview of the potential profitability and land-use outcomes for a range of small- and medium-scale instances within short computational times.

Income effects of CAP policy scenarios

The optimization models enable a comprehensive analysis of the economic implications of different CAP scenarios for farmers. In particular, this section provides an overview of the impact on operating income (i.e., the direct economic return from selling crops on the market) and CAP payments across policy settings, expressed in euros per hectare per

	Pure Farmer		CAP 2014–2020		CAP 2021–2023		CAP 2023–2027 - Diversification		CAP 2023–2027 - Rotation	
Id	Value (€)	Time (s)	Value (€)	Time (s)	Value (€)	Time (s)	Value (€)	Time (s)	Value (€)	Time (s)
TO1	78953.93	0.13	105443.43	0.05	97664.58	1.20	102332.83	4.31	102324.31	3.44
FE3	89596.04	2.03	124126.67	0.31	110866.96	5.95	115512.74	12.97	115493.81	8.17
FE4	102788.35	6.85	160616.45	4.40	138003.24	92.83	150493.15	183.64	146273.80	88.37
PR1	198626.22	14.10	202644.52	25.37	184087.97	31.73	196639.92	59.62	193809.32	29.14
CN2	103718.36	18.42	163665.70	41.81	149612.00	35.76	154659.64	105.18	158096.60	33.92
RO1	70545.30	0.15	164796.70	0.99	143734.86	1.88	148001.28	12.08	148425.90	2.00
MI2	136158.78	0.43	269264.29	2.89	172377.00	3.04	178527.36	14.38	180718.50	3.08
MI1	103948.90	1.32	245462.05	3.13	153873.10	9.83	161422.60	45.64	162042.73	9.42
AL3	127342.80	0.64	231957.19	1.03	179708.55	16.69	188144.37	59.69	188251.56	15.67
CN1	272821.70	0.32	447262.58	0.44	300310.80	8.18	336483.58	19.87	309425.88	7.50
MN2	210945.24	2.96	365743.48	5.56	314624.10	59.30	323475.20	150.31	323782.80	50.37
BO3	186914.88	0.33	312660.16	0.40	285978.30	5.08	293935.20	46.06	297234.72	5.80
AL4	289413.60	0.06	385550.48	0.07	344639.52	0.72	385829.67	3.54	355170.48	0.85
MN3	1187961.57	4.17	1483616.80	6.27	1110609.12	84.42	1255321.76	290.99	1140389.40	87.38
MN1	984338.56	19.66	1164167.92	21.60	1007001.96	66.63	1113700.52	282.26	1032238.68	73.67
FE2	293788.11	1.46	418549.48	2.03	386655.82	37.05	395036.26	203.56	401623.18	38.66
AL2	607236.00	0.33	1142377.44	0.38	736780.80	10.16	778857.48	26.13	755505.60	7.17
RO2	2784483.82	0.65	2853565.51	0.87	2363670.24	16.51	2788542.51	43.20	2421081.18	13.99
AL1	1141090.72	3.10	1661735.32	4.18	1263084.13	117.99	1344734.40	248.48	1301782.02	127.86
BO1	494902.73	16.07	1060005.41	17.11	854942.20	56.73	897388.55	163.48	884751.35	63.89
FE1	681327.87	2.06	1032126.09	2.04	908961.05	38.49	947684.25	116.19	945592.41	56.09
VEN1	819954.44	1.05	1724085.23	0.45	1415812.24	2.13	1531985.54	6.35	1455484.33	2.54
BO2	673816.47	38.30	1620454.55	16.64	1060714.66	51.91	1108213.97	114.33	1099827.46	59.56
Average	506116.28	5.85	753907.72	6.87	594944.05	30.94	647692.29	68.43	613883.74	30.99

Table 1.17: Comparison between Pure Farmer - CAP 2014–2020 - CAP 2021–2023 - CAP 2023–2027 (Diversification and Rotation) scenarios.

year to measure profitability at the farm scale.

The results reveal a downward trend in operating income across scenarios, suggesting increasing economic pressure associated with stricter CAP requirements (e.g., *Crop Diversification*, *Crop Rotation*). Figure 1.14 shows the variations in operating income, CAP payments, and total income (the sum of operating income and CAP payments), expressed in euros per hectare per year across the various policy scenarios. Operating income decreases from 783.12 €/ha per year in the Pure Farmer scenario to 603.72 €/ha per year under CAP 2021–2023, mainly because the Pure Farmer scenario is not subject to agro-ecological constraints. CAP payments experience a marked decline from CAP 2014–2020 to CAP 2021–2023 and then remain relatively stable under CAP 2023–2027. Total income reaches its maximum under CAP 2014–2020 (1252.79 €/ha per year), before decreasing in the subsequent policy periods and stabilizing under the CAP 2023–2027 scenarios.

Effect of increasing the number of crops

As shown by the results reported in Section 1.9.4, adding 5 or 10 crops to the crop mix of a farm leads to an increase of about 25% and 49% in the average total profit across all instances, respectively. This effect is illustrated in Figure 1.15, which reports the profit per hectare for each farm in the three sets Set_{real}, Set_{+5}, and Set_{+10}. These results indicate that increasing crop diversity can be economically advantageous for farmers.

From a practical perspective, models such as CGH enable the solution of these large-instances planning problems and can be embedded in decision-support systems to assist farmers in crop-mix selection, evaluation of what-if scenarios, and optimal production planning under sustainability requirements.

1.11 Conclusion

In this chapter, the Crop Rotation Problem (CRP-k) has been addressed. CRP-k is a multi-period problem, in which crops are assigned to heterogeneous plots of a farmland, under agronomic and management requirements and sustainability-driven constraints, arising from both public regulation (CAP) and private schemes (CdM). The modelling framework explicitly captures the fact that the profit of a crop depends on the crops sown on the past (as described by crop rotation).

A graph-based model is presented to represent feasible crop sequences and exploited to derive an arc-flow formulation (G-ILP). A set of policy-driven scenarios was also defined to represent the admissible combinations of regulatory requirements considered in the experimental campaign. To cover a broad spectrum of instance sizes and constraint regimes, both exact and heuristic solution approaches were developed: (*i*) an exact arc-flow model for small-to-medium instances; and a dynamic programming procedure for the single-plot case; (*ii*) matheuristic strategies based on column generation (CGH) and dynamic programming (DPH) techniques to compute high-quality solutions when exact optimization becomes computationally demanding; and (*iii*) a Lagrangian relaxation designed to compute tight bounds on the optimal solution. Computational results indicate that the proposed approaches provide reliable decision support across multiple CAP scenarios and under additional agroecological constraints, while enabling informative what-if analyses on crop portfolios and policy participation.

From an agricultural decision-making perspective, the optimization models provide two complementary benefits. First, they yield implementable, profit-oriented rotation plans that remain consistent with sustainability commitments, supporting farm managers in balancing income stability with soil-health and biodiversity objectives. Second, they offer a quantitative basis to assess how incentives and requirements reshape land use and crop diversity, which is valuable for both farm-level planning and policy evaluation. Overall, the chapter shows that suitably designed optimization models can translate complex agronomic knowledge and evolving regulatory frameworks into tractable decision-support tools for intensive arable systems.

1.12 Appendix: Proofs of theorems and lemmas

Proof of Lemma 1.8.1.

Proof. Recall that the time periods in the time horizon $T = \{1, \ldots, p\}$ correspond to first and second semesters of a given number of consecutive years. Since intermediate nodes of a path in G belong distinct layers, the number of layers is equal to the length, in terms of nodes, of a longest path on G, omitting the source node so_w and the demand node de_w of the path. Then, since each intermediate node on the path contains a sequence of τ crops (the last intermediate node in the path may contain a smaller number of crops), it is easy to see that a longest path from s to d must contain $\lceil p/\tau \rceil + 2$ nodes. Hence, the number of layers in G is $\lceil p/\tau \rceil$. □

Proof of Lemma 1.8.2.

Proof. Recall that, by definition, a node in each layer L_i is of the form $(a_1, a_2, \ldots, a_\tau)^{t_1,t_2,\ldots,t_\tau}$. We first compute the number of possible values that can be taken by the first and last periods of a node in layer L_i , i.e., t_1, t_τ. We do this by determining the minimum and the maximum values that can be taken by t_1 and t_τ.

As already observed, there exists on G a shortest path P_s only containing sequences of annual or first semester crops. Let $(a_1, a_2, \ldots, a_\tau)^{t_1,t_2,\ldots,t_\tau}$ and $(b_1, b_2, \ldots, b_\tau)^{q_1,q_2,\ldots,q_\tau}$ be the two (consecutive) nodes on the path P_s belonging to layers L_i and L_{i+1}, respectively. Then, periods $t_1, t_2, \ldots, t_\tau$ and $q_1, q_2, \ldots, q_\tau$ are consecutive odd periods, and we have $t_\tau = t_1+2\tau-2$, $q_1 = t_\tau+2 = t_1+2\tau$ and $q_\tau = q_1+2\tau-2 = t_1+4\tau-2$. Hence, traversing arc from $(a_1, a_2, \ldots, a_\tau)^{t_1,t_2,\ldots,t_\tau}$ to $(b_1, b_2, \ldots, b_\tau)^{q_1,q_2,\ldots,q_\tau}$, we move from period t_1 to $q_\tau = t_1+4\tau-2$. Observe that, no other sequence of 2τ crops produces a move bigger than this, in terms of number of periods. As a consequence, the node $(a_1, a_2, \ldots, a_\tau)^{t_1,t_2,\ldots,t_\tau} \in P_s$ of layer L_i has first and last period values, i.e., t_1 and t_τ respectively, that are not bigger than those of any other node of the layer. And we have: $t_1 = 2\tau i - 2\tau + 1$ and $t_\tau = 2\tau i - 1$. As already observed, a longest path exists on G, say P_l, in which the nodes contain subsequences of first and second semester crops. Hence, consecutive intermediate nodes on P_l are of the form $(a_1, a_2, \ldots, a_\tau)^{t_1,t_2,\ldots,t_\tau}$ and $(b_1, b_2, \ldots, b_\tau)^{q_1,q_2,\ldots,q_\tau}$, where $t_1, t_2, \ldots, t_\tau$ and $q_1, q_2, \ldots, q_\tau$ are consecutive periods. Since such nodes belong to adjacent layers, say L_i and L_{i+1}, an arc connecting them allows a "move", in terms of periods, from t_1 to $t_1 + 2\tau - 1$. Note that, no other sequence of 2τ crops produces a move smaller than this, in terms of number of periods. Hence, the minimum values taken by the first and last periods associated to nodes of layer L_i are the periods t_1 and t_τ of the node $(a_1, a_2, \ldots, a_\tau)^{t_1,t_2,\ldots,t_\tau}$ on P_l, for which we have $t_1 = \tau(i-1)+1$ and $q = \tau i$, respectively. By the above discussion, it follows that the first (the last) period t_1 (t_τ) of a node in layer L_i takes values between $\tau(i-1)+1$ and $2\tau i-2\tau+1 = 2\tau(i-1)+1$ (between $2\tau i-1$ and τi), i.e., $O(\tau i)$ possible values. Hence, the number of possible period sequences, $t_1, t_2, \ldots, t_\tau$ associated to nodes of L_i is $(\tau i)^2$. Recalling that, the nodes in the intermediate layer L_i contain crop sequences of at most τ crops, in which crop repetitions are allowed, the nodes in L_i are at most $\tau^2 i^2 n^\tau$, where n is the number of crops. □

Proof of Lemma 1.8.3.

Proof. The thesis easily follows by Lemmas 1.8.1 and 1.8.2 and since $\sum_{i=1}^{\lceil p/\tau \rceil} \tau^2 n^\tau i^2 = n^\tau \tau^2 (\lceil p/\tau \rceil (\lceil p/\tau \rceil + 1)(2 \lceil p/\tau \rceil + 1)/6$, i.e., $O(n^{\tau-1} p^3)$. □

Proof of Theorem 1.8.4.

Proof. We prove the theorem for $k = 2$ (i.e., the profit of a crop depends on the previous crop), by reduction from the Traveling Salesman Problem (TSP). In the TSP, given a directed graph $\mathcal{H} = (\mathcal{N}, \mathcal{A})$ with n nodes, where N is the set of nodes and A is the set of arcs, and a cost $d_{u,v}$ for each arc $(u, v) \in \mathcal{A}$, the problem is of finding a Hamiltonian path on $\mathcal{H}$ with minimum cost. Given an instance of TSP, we define an instance of CRP-2 with $2n$ annual crops, $p = 4n$ periods, a single class of homogeneous plots H_w, and $R_{max} = n$. Hence, we have $\tau = n$. The crops are partitioned into two sets $A = \{a_1, a_2, \ldots, a_n\}$ and $B = \{b_1, b_2, \ldots, b_n\}$, and there is a one-to-one correspondence between the crops in B and the nodes in the graph $\mathcal{H}$. In the pricing problem we set all dual variables to 0, but α_i^w and α_j^w. On this instance, the pricing problem consists in finding a sequence of $2n$ crops, say $\langle c_{u_1}, c_{u_2}, \ldots, c_{u_{2n}} \rangle$, with $c_{u_h} \in A \cup B$ for $h = 1, \ldots, 2n$, respectively assigned to the odd periods in $\{1, 2, \ldots, 4n-1\}$, corresponding to an arc $(i, j) \in G_1$ with positive reduced cost. We have that $i = (c_{u_1}, c_{u_2}, \ldots, c_{u_n})^{1,3,\ldots,2n-1}$ and $j = (c_{u_{n+1}}, c_{u_{n+2}}, \ldots, c_{u_{2n}})^{2n+1,2n+3,\ldots,4n-1}$ and that the reduced cost (1.65) of the arc is $p_{ij}^w + \alpha_i^w - \alpha_j^w$, since all dual variables but α_i^w and α_j^w are 0. Recalling that the profit p_{ij}^w of an arc $(i, j) \in G$ is given by the sum of the profits produced by the crops in node j and that $k = 2$ (i.e., the profit of a crop depends on the previous crop), the profit of the arc (i, j) is $p_{ij}^w = \sum_{h=n+1}^{2n} p_w(\langle c_{u_{h-1}}, c_{u_h} \rangle, 2h-1)$, where $p_w(\langle c_{u_{h-1}}, c_{u_h} \rangle, 2h-1)$ is the profit of crop c_{u_h} when it is preceded by crop $c_{u_{h-1}}$ (and assigned to period $2h-1$). The profit $p_w(\langle c_{u_{h-1}}, c_{u_h} \rangle, 2h-1)$ is defined as follows. If both crops $c_{u_{h-1}}$ and c_{u_h} belong to the set B, then $p_w(\langle c_{u_{h-1}}, c_{u_h} \rangle, 2h-1) = D - d_{c_{u_{h-1}}, c_{u_h}}$, where $D = \max\{d_{u,v} : (u, v) \in A\}$ (recall that a one-to-one correspondence exists between the crops in B and the nodes in H). If crops $c_{u_{h-1}}$ or c_{u_h} belong to the set A, then $p_w(\langle c_{u_{h-1}}, c_{u_h} \rangle, 2h-1) = 0$. The dual variables α_i^w and α_j^w depend on the nodes i and j, respectively, and are set as follows. If in the sequence related to node i (i.e., $\langle c_{u_1}, c_{u_2}, \ldots, c_{u_n} \rangle$) a crop of the set B exists, then $\alpha_i^w = -nD$ and 0 otherwise. If in the crop sequence related to node j (i.e., $\langle c_{u_{n+1}}, c_{u_{n+2}}, \ldots, c_{u_{2n}} \rangle$) a crop is repeated more than once or a crop of the set A exists, then $\alpha_j^w = nD$ and 0 otherwise. Observe that, since by definition $p_{ij}^w < nD$, the reduced cost $p_{ij}^w + \alpha_i^w - \alpha_j^w$ of the arc (i, j) is negative if node i (node j) contains a crop of the set B (set A) or if node j contains a crop repetition. On the other hand, if nodes i and j contain only crops of the sets A and B, respectively, and no crop repetition exists in j, then the crop sequence defined by j, i.e., $\langle c_{u_{n+1}}, c_{u_{n+2}}, \ldots, c_{u_{2n}} \rangle$, is a permutation of the n crops of the set B. This crop permutation corresponds to a Hamiltonian path P on $\mathcal{H}$ of cost $C(P)$, and the profit of the arc (i, j) is $(n-1)D - C(P)$, not negative by definition of D and since P contains exactly $n-1$ arcs. In fact, by definition, the profit produced by the first crop, $c_{u_{n+1}}$, of the sequence related to node j, i.e., $p_w(\langle c_{u_n}, c_{u_{n+1}} \rangle, 2n+1)$, is 0, since $c_{u_n} \in A$ and $c_{u_{n+1}} \in B$ ($c_{u_{n+1}}$ is the last crop of the sequence related to node i). As a consequence, the crop sequence related to node j, $\langle c_{u_{n+1}}, c_{u_{n+2}}, \ldots, c_{u_{2n}} \rangle$, of an arc with maximum reduced cost $(n-1)D - C(P^*)$ corresponds to a Hamiltonian path

P^* on $\mathcal{H}$ with minimum cost. On the other hand, given a Hamiltonian path P^* on $\mathcal{H}$ of minimum cost, the arc $(i,j) \in G_1$, in which the n-crop sequence in j is ordered as the nodes in P^* and where node i contains any n-crop sequence of crops in A, has maximum profit equal to $(n-1)D - C(P^*)$. Since CRP-k with $k > 2$ generalizes CRP-2, the pricing problem for G-ILP when $k > 2$ is strongly NP-hard, too. □

Proof of Theorem 1.8.5.

Proof. The proof is by reduction from the Equal Cardinality Partition (ECP). Given a set S of $2n$ positive integers, ECP consists in finding a partition of S into two subsets, S_1 and S_2, of equal cardinality (i.e., $|S_1| = |S_2| = n$), such that the sum of the numbers in S_1 equals the sum of the numbers in S_2.

Given an instance of ECP, let $a_1, a_2, \ldots, a_{2n}$ be the $2n$ positive numbers, and let $\sum_{i=1}^{2n} a_i = 2P$. Let us consider an instance of CRP-1, with $2n$ annual crops, $c_1, c_2, \ldots, c_{2n}$, $p = 4n$ periods and $\tau = n$. Then, the pricing problem consists in finding a crop sequence of length $2\tau = 2n$ maximizing (1.65), in which a crop is assigned to each odd period. Let us consider an instance of the pricing problem in which all dual variables but α_j^w are 0. Hence, the pricing problem consists in finding a sequence of $2n$ crops, say $\langle c_{u_1}, c_{u_2}, \ldots, c_{u_{2n}} \rangle$ assigned to each odd period in $\{1, 2, \ldots, 4n-1\}$, corresponding to an arc (i,j) of the graph G, such that $p_{ij}^w - \alpha_j^w$ is maximum (recall that all dual variables but α_j^w are 0). In this sequence, node i and j are of the form $i = (c_{u_1}, c_{u_2}, \ldots, c_{u_n})^{1,3,\ldots,2n-1}$ and $j = (c_{u_{n+1}}, c_{u_{n+2}}, \ldots, c_{u_{2n}})^{2n+1,2n+3,\ldots,4n-1}$, respectively. Recall that the profit of an arc is given by the profits of the crops in node j. Since $k = 1$, we set the profit of crop, c_{u_i}, to a_{u_i}, and, consequently, the profit of the arc (i,j) is $p_{ij}^w = \sum_{h=n+1}^{2n} a_{u_h}$. The dual variable α_j depends on the node in j and is set to $2P$ when (1) a crop appears more than once in the $2n$ crop-sequence, or (2) $\sum_{h=n+1}^{2n} a_h \neq P$, and 0 otherwise. Hence, the reduced cost $p_{ij}^w - \alpha_j^w$ of the arc (i,j) is positive and equal to P if and only if the two crop subsequences $\langle c_{u_1}, c_{u_2}, \ldots, c_{u_n} \rangle$ and $\langle c_{u_{n+1}}, c_{u_{n+2}}, \ldots, c_{u_{2n}} \rangle$ correspond to a partition $S_1 = \{a_{u_1}, a_{u_2}, \ldots, a_{u_n}\}$ and $S_2 = \{a_{u_{n+1}}, a_{u_{n+2}}, \ldots, a_{u_{2n}}\}$ of the set S of ECP, such that the sum of the numbers in S_1 and S_2 is the same. □

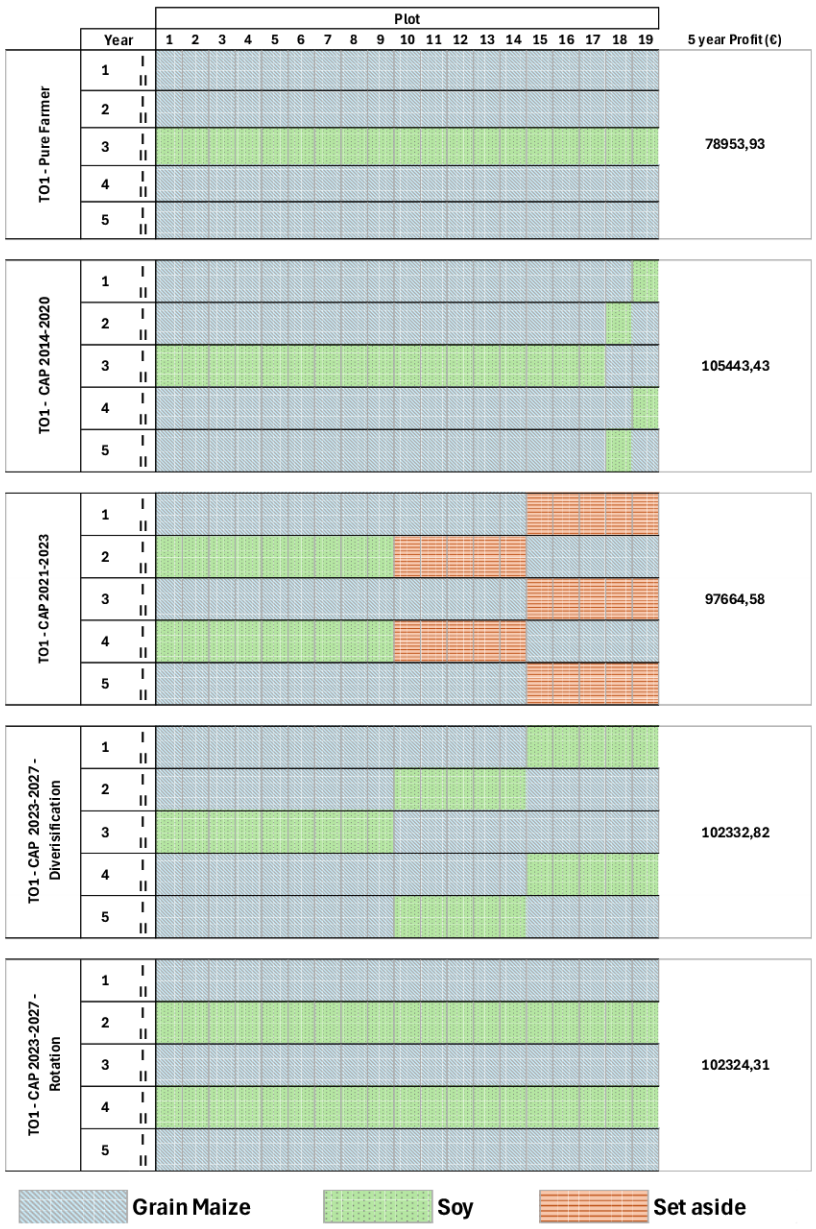

Figure 1.13: Profitability and land-use changes for instance TO1 under the following scenarios: Pure Farmer, CAP 2014–2020, CAP 2021–2023, CAP 2023–2027 - Diversification, and CAP 2023–2027 - Rotation.

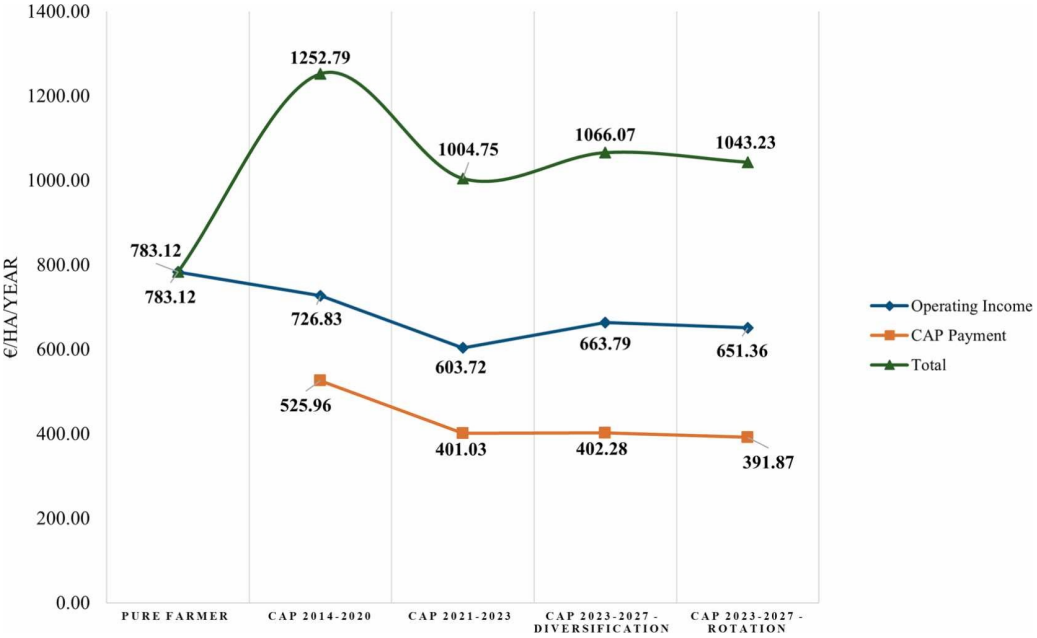

Figure 1.14: Changes in Crop Distribution Under Different CAP Policy Scenarios.

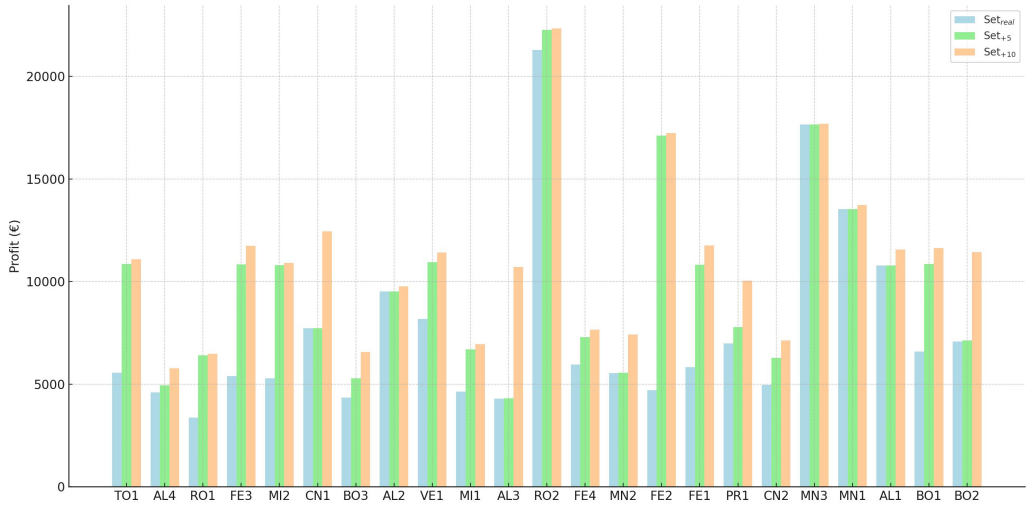

Figure 1.15: Profit per hectare of land for all instances on the three scenarios Set_{real}, Set_{+5} and Set_{+10}.

Chapter 2

Split Delivery VRP in Field Fertilization

2.1 Introduction

Field fertilization in viticulture has increasingly benefited from the integration of precision agriculture and optimization-based decision support. In particular, the use of Variable Rate Application (VRA) technologies, which determine the locally appropriate fertilizer dose rather than applying a uniform rate over the entire field, enables site-specific management of nitrogen inputs. Application rates can thus be adapted to the spatial variability in vine vigor and yield potential, improving both agronomic performance and resource-use efficiency [55, 56, 57]. While the agronomic motivation of VRA is well established, its implementation modifies the operational organization of field work. The amount of fertilizer required by each field becomes heterogeneous and is no longer directly proportional to area, which introduces a non-trivial routing and refilling problem. As a result, field fertilization becomes not only a matter of agronomic prescription, but also a problem of logistics and operations research.

In this context, we focus on the routing component of field fertilization, formulated as a single-vehicle variant of the Split Delivery Vehicle Routing Problem (SDVRP) with a binary refilling policy. The tractor, equipped with a fertilizer spreader of limited capacity, must visit a set of fields, each with a known fertilizer demand derived from the VRA prescription process. Refilling operations can only occur at a central depot, where fertilizer is stored in discrete units. The binary refilling rule implies that when the tank is empty, a full refill is performed, while in the presence of residual fertilizer, only a partial refill (half of the tank capacity) is allowed. Each field can be served in multiple trips (split deliveries) until its total demand is fully satisfied. The resulting problem has the objective of minimizing the total travel distance while respecting capacity and refilling constraints.

This specific problem can be modeled as a complete graph where nodes represent fields and the depot, and edges correspond to travel distances between them. The tractor acts as a single vehicle performing a sequence of trips, each starting and ending at the depot. The load carried at the beginning of a trip depends on the residual fertilizer from the previous trip and the applied refilling policy. Consequently, the trips are not independent, and the sequence in which they are executed affects both feasibility and optimality. This sequential dependency is one of the key differences between this formulation and the classical multi-vehicle SDVRP [58, 59].

As stated, the model's goal is to minimize the total distance traveled while guaranteeing complete satisfaction of the fertilizer demand for all fields. Feasibility requires respecting the vehicle capacity, ensuring that no trip exceeds the maximum load, and

that the total fertilizer delivered over all trips equals the total demand. The possibility of split deliveries provides additional flexibility and may reduce total distance; however, the refilling operations that link consecutive trips introduce a more complex coordination.

To formally capture these interactions, a Integer Linear Programming (ILP) formulation is proposed. It defines binary routing variables indicating whether a given arc is traversed in a specific trip, continuous variables describe fertilizer quantities moved along arcs, and additional binary variables govern the refilling decisions. The model enforces trip concatenation and binary refilling, together with demand satisfaction, load-conservation, and subtour elimination constraints. It provides an exact formulation for small- and medium-scale instances and serves as the reference for benchmarking algorithmic approaches.

Beyond the ILP formulation, a tailored Branch-and-Cut (B&C) algorithm is developed. The B&C starts from a relaxed model and incrementally introduces set of cuts when violated. When an integer solution is found, its feasibility is tested in a two-edge commodity-flow formulation. If the solution is feasible, it can be certified as a feasible solution for the SDVRP problem. Then, no-good cuts are generated to exclude the solution.

In addition, a dedicated post-processing algorithm is developed to translate the model solution into an executable fertilization plan. Starting from the optimal solution, this algorithm reconstructs, for each trip, which rows of each field are treated (possibly only partially), the corresponding working and travel times, and the required refills and breaks. It also aggregates trips into daily schedules under an 8-hour working limit. This post-processing step bridges the gap between the mathematical solution and field operations, providing a realistic and detailed implementation plan for the nitrogen application.

Finally, computational experiments are conducted on both real and synthetic instances, comparing the ILP formulation and the B&C algorithm in terms of computational time. Operational results are also reported for a real-world instance to highlight the practical relevance of explicitly modeling binary refilling and split deliveries, showing that the formulation captures realistic field constraints and yields efficient fertilization plans. Overall, the proposed approach links agronomic demand estimation (from VRA) with optimization-based routing, providing a rigorous decision-support framework for field fertilization logistics.

This chapter is organized as follows. Section 2.2 reviews the related literature. Section 2.3 introduces the problem definition and notation. Section 2.4 presents the ILP formulation. Section 2.5 describes the Branch-and-Cut framework and the separation algorithm. Section 2.6 details the Post-processing algorithm used to generate executable fertilization plans and daily schedules. Section 2.7 reports computational results on real and synthetic instances.

2.2 Literature review

The Split Delivery Vehicle Routing Problem (SDVRP) was originally introduced by Dror *et al.* [58, 60] as a relaxation of the classical Vehicle Routing Problem (VRP) in which customer demand may be served through multiple visits. Over the years, this variant has

attracted considerable attention within the operations research community due to both its practical relevance and its impact on routing efficiency.

Archetti and Speranza [61] provide a comprehensive survey of the SDVRP, reviewing modelling variants and both exact and heuristic algorithms, and stressing that in almost all contributions vehicle capacity is handled in the standard way: each route leaves the depot with full load and refilling is not explicitly modelled. In one of the seminal exact studies, Dror *et al.* [62] propose an ILP formulation for the SDVRP, strengthen it with several families of valid inequalities, and complement the model with heuristic procedures, showing that allowing split deliveries can substantially reduce total travel cost compared to the classical VRP. More recent work has extended the basic SDVRP in several directions. Gouveia *et al.* [63] study a multi-depot SDVRP and develop a compact ILP model together with a branch-and-cut algorithm, with particular emphasis on the interplay between depot assignment and split deliveries. Bianchessi *et al.* [64] consider the SDVRP with time windows, introduce a route-based ILP formulation, and design a branch-and-cut-and-price algorithm strengthened by tailored capacity, connectivity and time-window inequalities. In a related line, Archetti *et al.* [59] propose a branch-and-price approach in which routes are generated dynamically in a pricing subproblem, and the column generation scheme is mainly used to obtain high-quality heuristic solutions on large instances. Munari and Savelsbergh [65] propose a tailored branch-and-cut algorithm for the SDVRP with time windows, built on a two-index relaxed formulation. Whenever the relaxation yields an integer solution, they first verify the regularity property and then test feasibility using a two-edge ILP formulation. If the solution is infeasible, feasibility cuts are added to eliminate it; otherwise, it is accepted as a new incumbent and additional optimality cuts are generated and inserted into the relaxed model to further tighten the formulation. A different modelling perspective is adopted in [66], where two relaxations of the SDVRP are analysed: a two-index vehicle-flow formulation and a single-commodity flow formulation. On top of these relaxations, the authors develop a branch-and-cut algorithm with capacity, connectivity and pairing inequalities; candidate integer solutions of the relaxed model are tested for feasibility on a route-based formulation and, when infeasible, are eliminated through no-good cuts. Ropke *et al.* [67] further develop this line of research by designing both exact and heuristic methods for the SDVRP built around a two-index formulation and a branch-and-cut scheme: capacity and connectivity cuts are separated (either exactly or heuristically), integer solutions of the relaxed model are forced on a two-edge formulation, derived from [65], and the same class of no-good cuts as in [66] is used in case of infeasibility to refine the relaxation. Moreover, they embed the exact method in an ALNS-based metaheuristic to initialize the branch-and-cut algorithm with a strong incumbent. The last two contributions are the closest to the approach developed in this thesis, since our branch-and-cut framework is also based on a two-index vehicle-flow relaxation, a separation procedure for capacity and connectivity inequalities, and a feasibility subproblem defined on a two-edge commodity-flow formulation. However, all of the above works treat vehicle capacity in the standard SDVRP sense: each trip starts at the depot with full capacity, refilling decisions are not modelled, and the load of each trip does not depend on the sequence of preceding trips, so there is no explicit trip concatenation. By contrast, the problem studied in this chapter incorporates a binary

refilling mechanism at the depot, whereby the vehicle can refill only in fixed quantity and the load available on each trip depends on the residual load carried over from previous trips and on the refilling decisions. The methodological contribution is precisely to give a exact ILP formulation to this case of SDVRP, adapt the two-index relaxation and the branch-and-cut framework to this setting.

2.3 Problem Definition

The problem addressed in this chapter can be modeled as a variant of the Split Delivery Vehicle Routing Problem (SDVRP). The SDVRP consists in serving a set of customers by means of a fleet of capacitated vehicles whose routes start and end at a depot. The objective is the minimization of the total travelling cost of the routes. As already reported, SDVRP is a relaxation of the classical Vehicle Routing Problem (VRP), in which the restriction of visiting each customer only once is removed, allowing multiple deliveries to the same customer. Thus, the demand of a customer may also be greater than the capacity of a vehicle.

Formally, the SDVRP can be modeled by a graph $G = (V, A)$, with $V = \{0, 1, \ldots, n\}$ the set of nodes, where the set $C = \{1, \ldots, n\}$ includes all customers, and 0 is the depot. Each customer $i \in C$ has a non-negative demand $d_i \geq 0$, and a non-negative cost, $c_{ij} \geq 0$, is associated with traveling between each arc $(i, j) \in A$. It is assumed that the cost matrix is symmetric, i.e., $c_{ij} = c_{ji}$ $\forall (i, j) \in A$ and that the triangular inequality holds. A fleet of vehicles is available to serve customers. Each vehicle has the same capacity Q and can perform a single route starting from the depot with full load.

The routing problem arising from the Nitrogen Application problem addressed in this chapter can be modeled as a variant of SDVRP with a single vehicle (the tractor) that can travel multiple routes (i.e., trips) starting and ending at the depot. Each field can be modeled as a customer requiring a given amount of nitrogen, and the demand of each customer can be satisfied by performing multiple visits (each visit consists of spreading fertilizer on a portion of the field). Hence, in the graph G it is observed a one-to-one correspondence between the fields and the customer nodes $C = \{1, \ldots, n\}$, while node 0 of G is the depot where the tractor is refilled.

The tractor has a spreader with a tank of capacity Q, and it can be refilled at the end of each trip. The load at the beginning of a trip also depends on the nitrogen remaining in the tank at the end of the preceding trip. Furthermore, constraints exist on the refilling process. In fact, the nitrogen is stored in bags of $R = Q/2$ each and, when a bag is open, all the nitrogen must be poured in the tank of the spreader (otherwise, all the unstocked fertilizer would be lost). Therefore, two modes exist to refill the tank of the spreader: (i) with two entire bags, reaching the full capacity Q; (ii) with one bag, and so the next trip will start with a load of R plus the fertilizer remaining (if any) at the end of the previous trip. Note that mode (i) can be only used when the tank is empty.

Summarizing, the routing problem consists in: (1) finding a set of trips to be performed sequentially, (2) deciding the fertilizer to load at the beginning of each trip (i.e., R or $2R$ kg), in such a way that the customers' demand is fulfilled and the total traveled distance is minimized.

Note that a lower bound on the number of trips performed by the vehicle is

$$K_{\min} = \left\lceil \frac{\sum_{i \in C} d_i}{2R} \right\rceil,$$

i.e., the minimum number of trips required to serve all customers in $C = \{1, \ldots, n\}$, when the load of the vehicle at the beginning of each route is at its maximum value of $Q = 2R$. On the other hand, the maximum number of trips required to serve all customers is $K_{\max} = 2K_{\min}$ (see [59]). The set of all trips that can be performed is $K = \{1, 2, \ldots, K_{\max}\}$.

2.4 An Integer Linear Programming (ILP) model

In what follows, an Integer Linear Programming (ILP) formulation is presented, in which all the problem characteristics have been considered.

The ILP model is based on the following variables. Given the graph $G = (V, A)$, let y_{ij}^k be a continuous variable denoting the load of the vehicle at the trip k when arc $(i, j) \in A$ is traversed. Let $x_{ij}^k \in \{0, 1\}$ be a binary variable equal to 1 if the vehicle traverses arc $(i, j) \in A$ during trip k, and 0 otherwise. Finally, let $r^k \in \{0, 1\}$ be a binary variable equal to 1 if a maximum amount of $Q = 2R$ is loaded on the vehicle at the beginning of the trip k, and 0 if the load is equal to R. In the formulas below, given a node $i \in V$ of the graph, let $\delta^+(i)$ be the set of arcs emanating from node i and $\delta^-(i)$ be the set of arcs entering node i. The ILP model reads as follows.

$$\min \sum_{(i,j) \in A} \sum_{k=1}^{K_{max}} c_{i,j} x_{i,j}^k \tag{2.1}$$

$$\sum_{(j,i) \in \delta^-(i)} x_{j,i}^k - \sum_{(i,j) \in \delta^+(i)} x_{i,j}^k = 0, \quad \forall i \in N, \qquad k = 1, \ldots, K_{max} \tag{2.2}$$

$$\sum_{(i,j) \in \delta^+(i)} x_{i,j}^k \leq 1, \quad \forall i \in N, \qquad k = 1, \ldots, K_{max} \tag{2.3}$$

$$\sum_{(0,i) \in \delta^+(0)} x_{0,i}^k \leq \sum_{(0,i) \in \delta^-(0)} x_{0,i}^{k-1}, \qquad k = 2, \ldots, K_{max} \tag{2.4}$$

$$r^k - \sum_{(0,i) \in \delta^+(0)} x_{0,i}^k \leq 0, \qquad k = 1, \ldots, K_{max} \tag{2.5}$$

$$\sum_{(j,i) \in \delta^-(i)} y_{j,i}^k - \sum_{(i,j) \in \delta^+(i)} y_{i,j}^k \geq 0, \quad \forall i \in C, \qquad k = 1, \ldots, K_{max} \tag{2.6}$$

$$\sum_{(j,0) \in \delta^-(0)} y_{j,0}^k = \sum_{(0,j) \in \delta^+(0)} y_{0,j}^k + \sum_{j \in C} \Big(\sum_{(j,h) \in \delta^+(j)} y_{j,h}^k - \sum_{(h,j) \in \delta^-(j)} y_{h,j}^k \Big), \qquad k = 1, \ldots, K_{max} \tag{2.7}$$

$$\sum_{k=1}^{K_{max}} \Big(\sum_{(h,i) \in \delta^-(i)} y_{h,i}^k - \sum_{(i,h) \in \delta^+(i)} y_{i,h}^k \Big) = d_i, \qquad \forall i \in C \tag{2.8}$$

$$\sum_{(0,i) \in \delta^+(0)} y_{0,i}^k \geq \sum_{(i,0) \in \delta^-(0)} y_{i,0}^{k-1} + (r^k + 1)R - Q\Big(1 - \sum_{(i,0) \in \delta^-(0)} x_{0,i}^k\Big), \qquad k = 2, \ldots, K_{max} \tag{2.9}$$

$$\sum_{(0,i)\in\delta^+(0)} y^1_{0,i} = (r^1+1)R \tag{2.10}$$

$$\sum_{(0,i)\in\delta^+(0)} y^k_{0,i} \geq R \sum_{(0,i)\in\delta^+(0)} x^k_{0,i} + Rr^k \quad k = 1,\ldots,K_{max} \tag{2.11}$$

$$\sum_{(0,i)\in\delta^+(0)} y^k_{0,i} \leq \sum_{(i,0)\in\delta^-(0)} y^{k-1}_{i,0} + (r^k+1)R \quad k = 2,\ldots,K_{max} \tag{2.12}$$

$$\sum_{(i,0)\in\delta^-(0)} y^{k-1}_{i,0} + (r^k+1)R \leq Q, \quad k = 2,\ldots,K_{max} \tag{2.13}$$

$$y^k_{i,j} \leq Qx^k_{i,j}, \quad \forall i,j \in N \quad k = 1,\ldots,K_{max} \tag{2.14}$$

$$y^k_{i,j} \geq 0,\ x^k_{i,j} \in \{0,1\} \quad \forall i,j \in N \quad k = 1,\ldots,K_{max} \tag{2.15}$$

$$r^k \in \{0,1\}, \quad k = 1,\ldots,K_{max} \tag{2.16}$$

$$r^1 = 1 \tag{2.17}$$

The objective function (2.1) accounts for the traveled distance minimization. Constraints (2.2) is the flow conservation at customer nodes and Constraints (2.3) state that each vehcile exit a customer at most once in each trip k. Constraints (2.4) impose that trip k can be performed only if trip $k-1$ has been performed too, and Constraints (2.5) set to 0 variables r^k if trip k is not performed. Constraints (2.6) state that the load of the vehicle entering at a customer node cannot be greater than the load of vehicle leaving that node. Constraints (2.7) impose that, on each trip k, the load that returns to the depot must correspond to what was loaded minus what was delivered to the customers and Constraints (2.8) establish that all the customer demand must be satisfied. Constraints (2.9) link y and r variables and set the load of the vehicle at the beginning of trip k to the maximum value, i.e., $Q = 2R$, when $r^k = 1$. Constraints (2.10)–(2.13) regulate the vehicle load at the beginning of each trip. In particular, Constraint (2.10) initializes the load of the first trip equal to the vehicle capacity Q. Constraints (2.11) and (2.12) force equality with the initial load value at each trip. More precisely, Constraint (2.11) sets the load to be at least R when the trip is performed, and equal to 0 otherwise. Constraint (2.12) imposes an upper bound on the load at the departure of trip k, ensuring that it does not exceed the load returned from the previous trip plus the refilling amount. Finally, Constraint (2.13) guarantees that the sum of the remaining load at trip $k-1$ and the refilled quantity does not exceed the vehicle capacity Q. Constraints (2.7)–(2.8), along with Constraints (2.14), prevent the generation of subtours that are not connected to the depot.

2.5 Branch and Cut algorithm

Hereafter, a description of the Branch-and-Cut (B&C) algorithm is presented. First, we detail the two-index vehicle-flow relaxation (2IDX). Then, we describe the B&C procedure at the root node and at each node of the search tree. This includes the separation routines used to generate capacity and connectivity cuts, as well as a feasibility test based on the two-edge commodity-flow formulation used to verify whether an integer solution of the 2IDX formulation is feasible for the original SDVRP. Whenever feasibility is confirmed, the

corresponding objective value provides an upper bound to the optimal solution, denoted UB_{best}.

Algorithm 3 provides an overview of the complete B&C procedure employed for this variant of the SDVRP.

2.5.1 Two-Index Vehicle Flow Relaxation (2IDX)

The B&C framework builds on the two-index vehicle-flow formulation, hereafter referred to as 2IDX, originally proposed by Belenguer *et al.* [68] and closely related to the variant presented in [66]. This model provide lower bounds to the optimum of the Split Delivery Vehicle Routing Problem (SDVRP). Since vehicle flows are aggregated across edges, individual trips are not explicitly enumerated and the refilling strategy is not represented. Consequently, since that an integer feasible solution of this relaxation is not necessarily feasible for the SDVRP, additional feasibility checks are required.

Let $G = (V, E)$ be an undirected complete graph with depot 0 and customer set $C = \{1, \ldots, n\}$. The edge set is defined as $E = \{(i, j) : i < j,\, i, j \in V\}$. For any subset $S \subseteq C$, we define $\delta(S) = \{(i, j) \in E : (i \in S, j \notin S) \text{ or } (i \notin S, j \in S)\}$, that is the set of edges with exactly one endpoint in S. Let $E_c = \{(i, j) \in E : i < j,\, i, j \in C\}$ be the set of edges connecting two customers. Customer demands are denoted by d_i for $i \in C$, and the vehicle capacity by Q. For any subset S, the minimum number of vehicle trips needed to serve S is $k_S = \left\lceil \frac{\sum_{i \in S} d_i}{Q} \right\rceil$.

The model defines two sets of decision variables. The first set, x_{ij} for all $(i, j) \in E$, denotes the number of vehicles traversing edge (i, j). The second set, z_i for all $i \in V$, represents the number of visit to node i. The travel cost associated with edge (i, j) is c_{ij}.

The two-index vehicle-flow relaxation (2IDX) is formulated as follows:

$$\min \sum_{(i,j) \in E} c_{ij}\, x_{ij} \tag{2.18}$$

$$\sum_{(i,j) \in \delta(i)} x_{ij} = 2 z_i \qquad \forall i \in V, \tag{2.19}$$

$$\sum_{(i,j) \in \delta(S)} x_{ij} \geq 2 k_S \qquad \forall S \subseteq C,\ |S| \geq 1, \tag{2.20}$$

$$\sum_{i \in C} z_i - |C| \leq z_0 - 1, \tag{2.21}$$

$$x_{ij} \in \{0, 1\} \qquad \forall (i, j) \in E_c, \tag{2.22}$$

$$x_{ij} \in \mathbb{Z}_+ \qquad \forall (i, j) \in E \setminus E_c, \tag{2.23}$$

$$z_i \in \mathbb{Z}_+ \qquad \forall i \in V, \tag{2.24}$$

$$z_i \geq 1 \qquad \forall i \in C, \tag{2.25}$$

$$z_0 \geq k_C \tag{2.26}$$

The objective function (2.18) minimizes the total travel cost. Constraints (2.19) link edge flows and visit counts, ensuring that the total incident flow on a node equals twice

Algorithm 3: B&C scheme for the SDVRP

Set upper bound to the optimal solution $UB_{best} = +\infty$
Root-node cutting plane phase
Relax integrality of x, z in 2IDX
repeat
  Solve LP relaxation and get $(\bar{x}, \bar{z})$
  Separate capacity and connectivity cuts
  if *cuts found* **then**
    Add all cuts and continue
  else if $(\bar{x}, \bar{z})$ *is integer* **then**
    Test the 2IDX solution on the two-edge commodity-flow formulation 2.38–2.64
    if *feasible* **then**
      Get z_{2ECF}
      if $z_{2ECF} < UB_{best}$ **then**
        $UB_{best} \leftarrow z_{2ECF}$
        Add constraint $\sum_{(i,j)\in E} c_{ij}\, x_{ij} \leq z_{2ECF}$
      else if $z_{2ECF} = UB_{best}$ **then**
        return Optimal solution found (UB_{best})
    Add no-good cut and continue
until *LP solution is fractional and no violated cuts are found*
Remove non-tight cuts and restore integrality of x, z

Branch-and-cut tree search
Start B&C on 2IDX with cut callbacks
while *there is an open node* **do**
  Let $(\hat{x}, \hat{z})$ be the node LP solution
  if $(\hat{x}, \hat{z})$ *fractional* **then**
    Separate capacity and connectivity cuts
  else if $(\hat{x}, \hat{z})$ *integer* **then**
    Separate capacity and connectivity cuts
    if *no capacity and connectivity cuts* **then**
      Test the 2IDX solution on the two-edge commodity-flow formulation 2.38–2.64
      if *feasible* **then**
        Get z_{2ECF}
        if $z_{2ECF} < UB_{best}$ **then**
          $UB_{best} \leftarrow z_{2ECF}$
          Add constraint $\sum_{(i,j)\in E} c_{ij}\, x_{ij} \leq z_{2ECF}$
      Add no-good cut
return Optimal solution UB_{best}

the number of visits to that node. Constraints (2.20) represents the capacity constraints, which guarantee both vehicle capacity feasibility and subtour elimination. Capacity con-

straints are omitted from the initial formulation and are subsequently generated by heuristic and exact separation procedure (see Section 2.5.3). Constraint (2.21) strengthens the formulation by limiting the number of visits to the customers. For each customer $i \in C$, the quantity $z_i - 1$ represents the number of split deliveries at customer i. The constraint enforces that the total number of splits over all customers is strictly smaller than the number of routes, which is given by the number of visits to the depot z_0. This condition directly follows from *Property 3* in [66]. Constraints (2.22)–(2.26) specify that edges between customers are binary, edges involving the depot are non-negative integers, and visit counts are positive integers. In particular, for the depot, this value must be at least the minimum number of trips required to serve all customers. No explicit limit on the number of vehicles and refilling strategy, that leads to the concatenation of trips, in accordance with the relaxation's role as a lower bound for the SDVRP.

2.5.2 Root-node scheme

The B&C procedure starts at the root node, hereafter a detailed description of the procedure applied to the 2IDX model is reported.

At the root node, we do not rely solely on the MILP solver's default cuts. Instead, we implement a dedicated cutting-plane scheme on the 2IDX model, in which capacity cuts, connectivity cuts, and no-good cuts are iteratively added before launching the global B&C procedure. We first solve the LP relaxation of 2IDX. Given the resulting solution, we apply the separation routines for capacity and connectivity cuts described in Section 2.5.3 to detect violated inequalities. If any are found, they are added to 2IDX and the LP is reoptimized. If no violated capacity and connectivity cuts are identified, we then check the integrality of the current LP solution. If the LP solution of 2IDX is integer, we apply the feasibility check described in Section 2.5.4. When an integer solution of the 2IDX formulation is feasible in the two-edge commodity-flow model, the corresponding solution is also feasible for the SDVRP and can therefore be accepted as a candidate incumbent. Let z_{2ECF} denote the objective value returned by the two-edge commodity-flow formulation 2.38–2.64. If $z_{2ECF} < UB_{best}$, where UB_{best} is the best (smallest) upper bound available so far, we update $UB_{best} \leftarrow z_{2ECF}$ and strengthen 2IDX by adding the following optimality cut:

$$\sum_{(i,j)\in E} c_{ij}\, x_{ij} \leq z_{2ECF}. \tag{2.27}$$

Finally, regardless of feasibility, we add the no-good cut (2.65) to exclude the current integer configuration and resume the root-node cutting-plane loop.

The root-node phase stops when no additional violated capacity and connectivity cuts can be found, and when the LP solution is fractional, so that no further no-good cuts are generated. Before proceeding, we remove from the model all inequalities that are not tight at the final LP solution, in order to limit the size of the model passed to the subsequent B&C phase. We then restore integrality on the x and z variables.

During the tree search, the same separation logic is applied within callbacks: fractional node solutions trigger the separation of capacity and connectivity cuts, while integer node solutions are checked for SDVRP feasibility via the two-edge subproblem. It is important

to note that feasibility is tested only when no capacity and connectivity cuts are found.

2.5.3 Separation algorithms

The B&C algorithm incorporates three families of valid inequalities: *capacity cuts* and *connectivity cuts*, the same as introduced in [66], together with a set of *cumulative capacity cuts*, proposed in this work for the considered SDVRP variant in order to further strengthen the formulation.

Since Constraints (2.20) are exponentially many, enumerating all of them is computationally prohibitive for large instances, and they are therefore omitted from the initial formulation. To address this issue, we use both a heuristic and an exact separation procedure for capacity cuts. The heuristic procedure is applied whenever a fractional solution is encountered at a node of the B&C tree, whereas the exact separation is performed at the root node and whenever an integer solution is found. This choice allows to test the feasibility of an integer solution only when it is also feasible for 2IDX. In addition to capacity cuts, a polynomial-time procedure is employed to separate connectivity cuts.

Heuristic separation of capacity cuts

The approach is inspired by the method presented in [67], which is similar to the *greedy rounded capacity heuristic* of Naddef *et al.* [69].

The main idea is to identify subsets S that violate the capacity cuts, i.e., those such that $\sum_{(i,j)\in\delta(S)} x_{ij} < 2k_S$. Starting from a seed node, the separation procedure attempts to expand S by adding the customer $r \in C \setminus S$ that yields the greatest violation, and continues this process until no further violated cut can be found.

The heuristic procedure starts from a seed customer $i \in C$, initially defining $S = \{i\}$. The subset S is then expanded iteratively by adding, at each step, the customer r that maximise the violation of the Constraints (2.20). For each candidate expansion, we compute the corresponding violation:

$$v(S \cup \{r\}) = 2k_{S\cup\{r\}} - \sum_{(i,j)\in\delta(S\cup\{r\})} x_{ij},$$

where, for a given subset $S \cup \{r\}$, $k_{S\cup\{r\}} = 2\left\lceil \frac{\sum_{h\in S\cup\{r\}} d_h}{Q} \right\rceil$. Whenever $v(S \cup \{r\}) > 0$, a violated cut is detected. Among all such candidates, we add to S the customer r that yields the largest violation. For a given set S, the process terminates when no further customer leads to a positive violation.

To enhance diversification and prevent deterministic growth patterns, a random noise term is added to the evaluation of $v(S \cup \{r\})$, introducing stochasticity in the selection process. To further improve coverage, the heuristic is repeated multiple times for each seed node, exploiting the randomization to generate diverse subsets.

Exact separation of capacity cuts

As already mentioned above, exact separation of capacity cuts is performed at the root node and whenever an integer solution is obtained at a node of the B&C tree. The aim

is to identify, if it exists, a subset S that yields the maximum violation of Constraints (2.20).

To this end, we solve the following ILP, the same as presented in [67], on the undirected graph $G = (V, E)$ as introduced in Section 2.5.1. Given a solution of the 2IDX model (2.18)–(2.26), we introduce binary variables y_i indicating whether customer i belongs to S, binary variables w_{ij} indicating whether node i or node j belongs to S, and an integer variable k denoting the minimum number of vehicle trips required to satisfy the demand of subset S, i.e., $k = \lceil \sum_{i \in S} d_i / Q \rceil$. The separation problem reads as follows:

$$\min \sum_{(i,j) \in E} \tilde{x}_{ij}\, w_{ij} - 2k \tag{2.28}$$

$$w_{ij} \;\geq\; y_i - y_j, \qquad \forall (i,j) \in E, \tag{2.29}$$

$$w_{ij} \;\geq\; y_j - y_i, \qquad \forall (i,j) \in E, \tag{2.30}$$

$$\sum_{i \in C} d_i\, y_i \;\leq\; Qk, \tag{2.31}$$

$$\sum_{i \in C} d_i\, y_i \;\geq\; 1 + Q(k-1), \tag{2.32}$$

$$y_0 = 0, \tag{2.33}$$

$$y_i \in \{0,1\} \qquad \forall i \in V, \tag{2.34}$$

$$w_{ij} \in \{0,1\} \qquad \forall (i,j) \in E, \tag{2.35}$$

$$k \in \left\{ 1, 2, \ldots, \left\lceil \frac{\sum_{i \in C} d_i}{Q} \right\rceil \right\}. \tag{2.36}$$

The objective function (2.28) searches for the most violated capacity inequality. Constraints (2.29)–(2.30) force w_{ij} to take value 1 whenever the tail node or the head node of the arc (i, j) belongs to S, and 0 otherwise, so that $w_{ij} = 1$ if and only if $(i, j) \in \delta(S)$. Constraints (2.31)–(2.32) ensure that k is equal to $\lceil \sum_{i \in S} d_i / Q \rceil$. Constraint (2.33) fixes $y_0 = 0$, since the depot does not belong to any customer subset. Constraints (2.34)–(2.36) define the domains of the decision variables.

The subset S associated with a violated capacity cut is then given by $S = \{\, i \in C : y_i = 1 \,\}$ in the optimal solution of the separation problem.

Separation of cumulative capacity cuts

Along with these standard capacity cuts, we also add a family of *cumulative capacity cuts*:

$$\sum_{(i,j) \in \delta(S)} x_{ij} \;+\; \sum_{(i,j) \in \delta(C \setminus S)} x_{ij} \;\geq\; 2\big(k_S + k_{C \setminus S} + 1\big) - 2 \sum_{(i,j) \in E(S, C \setminus S)} x_{ij}, \tag{2.37}$$

where $C \setminus S$ is the complement of S in C, $k_{C \setminus S}$ is the minimum number of trips required to satisfy the demand of $C \setminus S$, i.e., $k_{C \setminus S} = \left\lceil \frac{\sum_{i \in C \setminus S} d_i}{Q} \right\rceil$, and $E(S, C \setminus S)$ denotes the set of edges with one endpoint in S and the other in $C \setminus S$:

$$E(S, C \setminus S) = \big\{ (i,j) \in E : (i \in S,\, j \in C \setminus S) \text{ or } (i \in C \setminus S,\, j \in S) \big\}.$$

These inequalities are motivated by the specific structure of our SDVRP variant: in some cases, we can certify that at least one additional trip is required in total, and we strengthen the capacity cuts accordingly. This refinement exploits the fact that, due to the concatenation of trips induced by the refilling process, a vehicle may enter S with a residual load rather than with full capacity Q.

Given a subset S and its complement $C \setminus S$, we compute, for both, a *residual capacity* and a *residual demand*. The residual capacity, denoted by $charge_S$ (respectively $charge_{C\setminus S}$), represents the remaining load in the vehicle after serving the subset with trips of full capacity Q. We define it, for the subset S, as:

$$charge_S = \left\lceil \frac{\sum_{i\in S} d_i}{Q} \right\rceil Q - \sum_{i\in S} d_i.$$

If refilling of size R is allowed (i.e., the residual capacity is at most R), this value is added to $charge_S$. The same computation is performed for $charge_{C\setminus S}$, replacing S with $C \setminus S$.

The residual demand, denoted by $residue_S$ (respectively $residue_{C\setminus S}$), measures the unserved demand after an integer number of full capacity trips in S:

$$residue_S = \sum_{i\in S} d_i - \left\lfloor \frac{\sum_{i\in S} d_i}{Q} \right\rfloor Q,$$

which can also be expressed, before adding the refilling amount, as the complement of $charge_S$ with respect to Q, i.e., $residue_S = Q - charge_S$. The same definitions apply to the subset $C \setminus S$.

If the residual capacity of S is strictly smaller than the residual demand of $C \setminus S$, and, symmetrically, the residual capacity of $C \setminus S$ is strictly smaller than the residual demand of S, then at least one additional trip is required, and we add the *cumulative capacity cut* (2.37). It is important to note that this inequality is relevant only when there are no edges between S and $C \setminus S$, i.e., when $\sum_{(i,j)\in E(S,C\setminus S)} x_{ij} = 0$.

Connectivity cuts

To separate the connectivity cuts, we apply a max–flow/min–cut procedure on G, using the current values of x_{ij} as arc capacities. The resulting minimum cut defines a candidate subset S that may correspond to a violated connectivity inequality.

Iteratively, each node $i \in C$ is considered as the source, while the depot acts as the sink. For each of these graph G, we compute the maximum flow and its correspondent minimum cut that separates node i from the depot.

The connectivity constraint associated with a subset of nodes $S \subseteq C$ and with node $i \in S$ takes the following form:

$$\sum_{(i,j)\in\delta(S)} x_{ij} \;\geq\; 2\, z_i, \qquad\qquad \forall S \subseteq C, |S| \geq 2, \forall i \in S$$

where $\delta(S)$ denotes the set of edges crossing the subset S. This inequality ensures that the flow enter the subset S, i.e., the number of visit, must be greater or equal than twice the variable z_i for each nodes $i \in S$. In the connectivity cut procedure, for each subset S,

only the violated cuts with $i = \arg\max_{u \in S}\{z_u\}$ are retained, in order to generate fewer but stronger cuts.

Although connectivity cuts are dominated by the capacity cuts, they are still included in the B&C framework because they can be efficiently separated through a polynomial–time exact max–flow algorithm.

2.5.4 Feasible solutions to the SDVRP

A key component of the proposed branch-and-cut (B&C) algorithm is the verification of feasibility in the original SDVRP for any integer solution generated by the 2IDX model. Since 2IDX is a relaxation of the SDVRP, an integer solution to 2IDX is not necessarily feasible for the SDVRP. In particular, the relaxation does not explicitly model essential operational features such as trip sequencing and refilling decisions.

Therefore, whenever the 2IDX model yields an integer solution, its feasibility is verified by solving the two-edge commodity-flow formulation introduced below. If this feasibility check is successful, the solution is feasible for the SDVRP and is accepted as a valid incumbent. Its objective value is then compared with the best upper bound found so far, and the bound is updated whenever an improvement is obtained. Finally, the no-good cut (2.65) is added to exclude the current 2IDX configuration from further consideration within the B&C framework.

Thus, SDVRP feasibility is assessed by solving a two-edge commodity-flow model defined on a reduced graph. The two-edge formulation was originally introduced in [65] and subsequently employed in [67]. In this work, the formulation is adapted to account for the specific operational setting under consideration, in particular the presence of the binary refilling policy at the depot and the trip concatenation.

Let $\tilde{E} \subseteq E$ denote the set of undirected edges selected by the integer solution of the 2IDX model, and let $\tilde{A}$ be the corresponding set of directed arcs induced by $\tilde{E}$ in the two-edge formulation, i.e., for each edge $(i, j) \in \tilde{E}$, two directed arcs (i, j) and (j, i) belong to $\tilde{A}$. The model introduces *two-edge* binary variables x_{hij}, defined for pairs of consecutive arcs $(h, i) \in \tilde{A}$ and $(i, j) \in \tilde{A}$, which indicate whether a vehicle traverses arc (h, i) immediately followed by arc (i, j).

In addition, a set of integer variables x^k_{0i0} is introduced to indicate whether a vehicle performs a direct trip depot–customer–depot, in trip k. Binary variables x^k_{0ij} and x^k_{hi0} are also defined to represent, respectively, a vehicle traveling from the depot to customer i and then to customer j, and a vehicle traveling from customer h to customer i and then returning to the depot, during trip k. Alongside these routing variables, continuous variables y^k_{0i} and y^k_{i0} represent the load carried by the vehicle on depot-customer arcs $(0, i)$ and $(i, 0)$ during trip k, for arcs belonging to $\tilde{A}$. When both i and j are customers, the load transported on arc (i, j) is instead represented by the continuous variables y_{ij}, that no depend on trip k.

Finally, let $r^k \in \{0, 1\}$ be a binary variable equal to 1 if the vehicle is loaded with the maximum amount $Q = 2R$ at the beginning of trip k, and equal to 0 if the initial load is R. Variable $\alpha^k \in \{0, 1\}$ indicates whether trip k is performed. The travel costs are

redefined as

$$\bar{c}_{hij} = \begin{cases} c_{hi}, & \text{if } j \in C, \\ c_{hi} + c_{ij}, & \text{otherwise,} \end{cases} \qquad \forall (h,i),(i,j) \in \tilde{A}.$$

The feasibility check is performed by solving the following two-edge commodity-flow formulation:

$$\min \sum_{(h,i)\in\tilde{A}} \sum_{(i,j)\in\tilde{A}} \bar{c}_{hij} x_{hij} \tag{2.38}$$

$$\sum_{(h,i)\in\delta^-(i)} x_{hij} \leq 1, \qquad \forall (i,j) \in \tilde{A}, i,j \in C \tag{2.39}$$

$$\sum_{(h,i)\in\delta^-(i)} x_{hij} = \sum_{(j,h)\in\delta^+(j)} x_{ijh}, \qquad \forall (i,j) \in \tilde{A}, i,j \in C \tag{2.40}$$

$$\sum_{k=1}^{K_{max}} x^k_{0ij} = x_{0ij}, \qquad \forall (0,i),(i,j) \in \tilde{A}, i \in C, j \in N \tag{2.41}$$

$$\sum_{k=1}^{K_{max}} x^k_{hi0} = x_{hi0}, \qquad \forall (h,i),(i,0) \in \tilde{A}, h,i \in C \tag{2.42}$$

$$\sum_{(i,j)\in\delta^+(i)} x^k_{0ij} + \sum_{(i,0)\in\delta^+(i)} x^k_{0i0} \leq 1, \qquad \forall (0,i) \in \tilde{A}, k \in K \tag{2.43}$$

$$\sum_{(h,i)\in\delta^-(i)} x^k_{hi0} + \sum_{(0,i)\in\delta^-(i)} x^k_{0i0} \leq 1, \qquad \forall (i,0) \in \tilde{A}, k \in K \tag{2.44}$$

$$y_{i,j} \leq Q \sum_{(h,i)\in\delta^-(i)} x_{hij}, \qquad \forall (i,j) \in \tilde{A}, i,j \in C, \tag{2.45}$$

$$y^k_{0,i} \leq Q \sum_{(i,j)\in\delta^+(i)} x^k_{0ij}, \qquad \forall (0,i) \in \tilde{A}, i \in C, \ \forall k \in K \tag{2.46}$$

$$y^k_{i,0} \leq Q \sum_{(h,i)\in\delta^-(i)} x^k_{hi0}, \qquad \forall (i,0) \in \tilde{A}, i \in C, \ \forall k \in K \tag{2.47}$$

$$y_{i,j} \leq y_{h,i} + Q(1 - x_{hij}), \quad \forall (h,i),(i,j) \in \tilde{A}, \qquad i,j,h \in C \tag{2.48}$$

$$y_{i,j} \leq y^k_{0,i} + Q(1 - x^k_{0ij}), \quad \forall (0,i),(i,j) \in \tilde{A}, \qquad i,j \in C, k \in K \tag{2.49}$$

$$y_{i,j} \geq y^k_{j,0} - Q(1 - x^k_{ij0}), \quad \forall (i,j),(j,0) \in \tilde{A}, \qquad i,j \in C, k \in K \tag{2.50}$$

$$\sum_{i\in C} y^k_{i,0} \leq \sum_{i\in C} y^k_{0,i}, \qquad \forall k \in K \tag{2.51}$$

$$\sum_{k=1}^{K_{max}} (y^k_{0,i} + \sum_{(j,i)\in\delta^-(i)\in A: j\in C} y_{j,i} - y^k_{i,0} - \sum_{(i,j)\in\delta^+(i): j\in C} y_{i,j}) = d_i, \qquad \forall i \in C \tag{2.52}$$

$$\sum_{(0,i)\in\delta^+(0)} y^1_{0,i} = (r^1 + 1)R \tag{2.53}$$

$$\sum_{(0,i)\in\delta^+(0)} y^k_{0,i} \geq \sum_{(0,i)\in\delta^-(0)} y^{k-1}_{i,0} + (r^k + 1)R - Q(1 - \alpha^k), \qquad k = 2,\ldots,K_{max} \tag{2.54}$$

$$\sum_{(0,i)\in\delta^+(0)} y^k_{0,i} \leq \sum_{(i,0)\in\delta^-(0)} y^{k-1}_{i,0} + (r^k + 1)R, \qquad k = 2,\ldots,K_{max} \tag{2.55}$$

$$\sum_{(i,0)\in\delta^-(0)} y^{k-1}_{i,0} + (r^k + 1)R \leq Q, \qquad k = 2,\ldots,K_{max} \tag{2.56}$$

$$\sum_{(0,i)\in\delta^+(0)} y^k_{0,i} \leq Q\alpha^k, \qquad \forall k \in K \tag{2.57}$$

$$\sum_{(0,i)\in\delta^+(0)} y^k_{0,i} \geq R\alpha^k + Rr^k, \qquad \forall k \in K \tag{2.58}$$

$$r^k \leq \alpha^k, \qquad \forall k \in K \tag{2.59}$$

$$\alpha^k \leq \alpha^{k-1}, \qquad k = 2,\ldots,K_{max} \tag{2.60}$$

$$y^k_{i,j} \geq 0, \qquad \forall (i,j) \in A\,, k \in K \tag{2.61}$$
$$x_{0,i,0} \in \mathbb{Z}_+, \qquad \forall i \in C \tag{2.62}$$
$$x_{h,i,j}, x^k_{0,i,j}, x^k_{h,i,0} \in \{0,1\}, \qquad \forall (h,i),(i,j) \in \tilde{A}\,, k \in K \tag{2.63}$$
$$\alpha^k, r^k \in \{0,1\}, \qquad \forall k \in K \tag{2.64}$$

The objective function (2.38) minimizes the total routing cost. Constraint (2.39) ensures that, for each customer-customer arc $(i,j) \in \tilde{A}$, at most one vehicle can traverse the arc, while Constraint (2.40) enforces flow conservation in the two-edge representation. Constraints (2.41)–(2.44) link the trip-indexed routing variables to their aggregated counterparts and regulate trip k dependencies. In particular, Constraints (2.41) and (2.42) aggregate trip into the corresponding arc variables. Constraints (2.43) and (2.44) impose that, for each trip k, at most one departure from the depot and at most one return to the depot are allowed, ensuring that each trip corresponds to a single route segment. Constraint (2.45) enforces the load of the vehicle to be zero whenever the corresponding arc is not selected, in case that all the node involved are customers. Constraints (2.46) and (2.47) link the loads departing from and returning to the depot with the arc customer–customer, in each trip k. Constraints (2.48)–(2.50) ensure consistency of the load across consecutive arcs, including depot-start and depot-return sequences. Finally, Constraint (2.51) guarantees that, for each trip k, the total load returning to the depot does not exceed the total load departing from the depot, thereby enforcing global load consistency. Constraint (2.52) enforces demand satisfaction by balancing, for each customer $i \in C$, the total quantity delivered across all trips and arcs in the reduced graph with the corresponding demand d_i. Constraints (2.53)–(2.59) model the binary refilling policy and the temporal structure of trips, in the same way introduced in the ILP model. Constraint (2.60) enforces trip concatenation, i.e., trip k can be performed only if trip $k-1$ is also performed. Finally, Constraints (2.61)–(2.64) define the domains of the continuous, integer, and binary decision variables.

Whenever a feasible solution is found, it provides a candidate incumbent. As discussed in Section 2.5.2, we compare its objective value with the best upper bound available so far and update UB_{best} if an improvement is obtained. In this case, we also add the optimality cut (2.27) to strengthen the 2IDX relaxation. In both cases, we subsequently introduce a *no-good cut* to prevent the algorithm from revisiting the same integer 2IDX configuration. The cut is the same introduced in [66, 67]. Let E denote the set of undirected edges of the 2IDX formulation, and let $\tilde{E} \subseteq E$ be the set of undirected edges selected by the current integer 2IDX solution. We add the following cut to the 2IDX model:

$$\sum_{(i,j) \in E \setminus \tilde{E}} x_{ij} \geq 1. \tag{2.65}$$

Constraint (2.65) enforces that any subsequent integer solution must include at least one edge that is not present in the incumbent 2IDX configuration. Therefore, the current configuration is excluded, while all other integer solutions remain admissible.

2.6 Model solution deployment for nitrogen application by a Post-processing algorithm

Given the optimal solution and the data of each row (length, nitrogen consumption, time required to traverse it), directly available from VRA information, a tailored Post-processing algorithm (Algorithm 4) has been devised to determine, for each trip k, (1) the rows that must be completed for each field, and (2) the corresponding completion time.

For each trip k, the algorithm takes as input the amount of fertilizer delivered to each customer node i (i.e., field i), equal to:

$$f_i^k = \sum_{(j,i)\in\delta^-(i)} y_{j,i}^k - \sum_{(i,j)\in\delta^+(i)} y_{i,j}^k \geq 0, \quad \forall i \in C$$

as defined by Constraints (2.6). If the amount of fertilizer delivered to a node f_i^k is lower than its demand d_i (as allowed by the SDVRP model), the field is considered only partially treated. In this case, the algorithm computes the rows to treat, the covered length of the last treated row (which can be partially processed), and the completion time for the treatment. It is assumed that the rows s of each field i, $s \in S_i = \{1, 2, \ldots, S_{N_i}\}$, are numbered from 1 to S_{N_i}, starting from the vehicle's entry point into the field. If the vehicle does not complete the treatment of a field during a trip, its next visit to the same field will start from the last row that was partially treated in the previous trip. The operator is assumed to know where the vehicle stopped the fertilizer spreading during the previous visit. The travel distance from the entry point of the field to the first row to treat is neglected and not considered in the algorithm. More precisely, given the (residual) fertilizer n_i^s required by row s of field i, for $i = 1, \ldots, n$, the Post-processing algorithm sequentially selects the rows to treat until the total fertilizer delivered to the field, g_i, reaches f_i^k. If the remaining fertilizer is not sufficient to complete the last row, only a portion of it is treated. In this case, the algorithm computes the portion of the row remaining to process. Then, the amount of fertilizer delivered to field i is subtracted from the node's remaining demand: $d_i \leftarrow d_i - f_i^k$. If the remaining demand becomes equal to 0, the node (i.e., the field) is considered fully treated. The arcs of the graph used in the optimal solution found by the ILP model, representing the roads connecting the depot and the fields, are employed by the Post-processing algorithm to compute the travel times. Given the distances between nodes (represented by c_{ij} in the model) and assuming an average vehicle speed on the road, the travel time for each trip is calculated. For each trip, the total time is obtained by summing up the travel times among the fields visited and the time required to treat the rows. Refilling times are also considered. At the end of each trip, the vehicle may be refilled with either a full load ($Q = 2R$) or a half load (R), depending on the solution. Each refill mode incurs a different fixed time, which is added to the total completion time. Mandatory breaks or work pauses are also included by the Post-processing algorithm to estimate the total duration of each trip.

Hence, the algorithm enables a detailed reconstruction of the operations to be performed during each trip: the required refill, the portion (i.e., the rows) of each field that must be treated, the row in which the treatment must be interrupted, and the total working time. This level of detail helps the operator to better understand and execute the

planned trips.

Furthermore, given the duration of each trip, the Post-processing algorithm groups trips into daily schedules, considering a maximum working time per day of 8 hours. This provides a realistic operational plan and allows for an accurate estimate of the total number of working days required to complete the fertilization process.

The results of this approach of the Post-processing algorithm applied in real data are presented in Section 2.7

Algorithm 4: Post-processing algorithm

Input:
Optimal solution of Model (2.1)–(2.16)
Speed $v_{i,j}$ of tractor on each arc $(i,j) \in A$
For each row $s \in S_i$, where $S_i = \{1, 2, \ldots, s_{N_i}\}$ is the set of rows in node $i \setminus \{0\}$:
- Length l_i^s
- Residual nitrogen required n_i^s
- Traversal time t_i^s

Initialization: Trip index $k = 1$; Trip completion time $t_k = 0$; First uncompleted row of node i, $s_U^i = 1, \forall S_i$

while $k \leq K_{\max}$ **do**
  foreach *node i visited in trip k* **do**
    Retrieve delivered fertilizer: $f_i^k = \sum_{(j,i)\in\delta^-(i)} y_{j,i}^k - \sum_{(i,j)\in\delta^+(i)} y_{i,j}^k \geq 0$ (see Constraint (2.6))
    if $f_i^k < d_i$ **then**
      Mark node i as **partially treated**;
      Initialize cumulative nitrogen applied: $g_i = 0$;
      while $g_i < f_i^k$ **do**
        $g_i \leftarrow g_i + n_i^s$;
        $t_k \leftarrow t_k + t_i^s$;
        $s_U^i \leftarrow s_U^i + 1$;
      Update residual nitrogen consumption of s_U^i: $n_i^{s_U^i} \leftarrow g_i - f_i^k$;
      Compute partial length completion of row s equal to $(1 - ((g_i - f_i^k)/n_i^{s_U^i})) * l_i^s$;
      Update residual demand at node i: $d_i \leftarrow d_i - f_i^k$;
    else
      Mark node i as **fully treated**;
      Compute completion time: $t_k \leftarrow t_k + \sum_{s=s_U^i}^{s_{N_i}} t_i^s$;
  foreach *arc (i,j) visited in trip k* **do**
    Compute road travel time and add to trip completion time: $t_k \leftarrow t_k + c_{ij}/v_{ij}$
  Update trip completion time t_k with the refill time of trip k (which depends on the value of r^k);
  $k \leftarrow k + 1$;

Group trips into working days, ensuring daily limit of 8 hours is respected;

Output:
Number of trips to perform per day
Number of nodes (fields) and related rows treated per trip
Covered length of first uncompleted row of each node i for each trip k
Completion time of each trip

2.7 Computational Experiments

This section reports the computational experiments conducted for the Split Delivery Vehicle Routing Problem (SDVRP). The instance set is described, including both real-

world and synthetic data, together with all parameters required to implement and reproduce the proposed models. Results are then presented for the comparison between the ILP formulation (Section 2.4) and the B&C approach (Section 2.5). Finally, the outcomes of the Post-processing algorithm introduced in Section 2.6 are reported, with the aim of generating executable fertilization plans and daily schedules for the real-world case study.

The computational experiments were conducted on an ASUS ExpertBook laptop equipped with a 1.70 GHz processor and 16 GB of RAM. Both the ILP formulation, the B&C algorithm, and the Post-processing procedure were implemented in the Julia programming language using Gurobi 13.0.0.

2.7.1 Instance Description

The real-world data come from a vineyard estate located in the south of France, near Montpellier (Villeneuve-lès-Maguelone). Figure 2.1 shows the location of the 51 blocks of the estate, grouped into 7 fields (numbered 1 to 7), and of the depot (numbered 0). The different colors of the blocks correspond to different nitrogen demand levels. Each field is modeled as a node of a graph: node 0 represents the depot for refilling the tractor, whereas nodes 1–7 represent the fields to be treated. Specifically, plots that were geographically close and typically treated together by the operator were aggregated into a single field. The total nitrogen demand of a field, denoted as d_i in the model and measured in kilograms (kg), is given by the sum of the total nitrogen requirement of each plot of the field. Meanwhile, using the mathematical models of Variable Nitrogen Application (VRA), the input data required by the Post-processing algorithm can be easily obtained. For each vine row s on field i, the length l_i^s, the nitrogen consumption n_i^s and the traversal time t_i^s, derived from the operational speed within the row, can be easily computed. The distances between each pair of fields, as well as between the depot and each field, were denoted by c_{ij} and based on the actual road paths followed by the operator, as reconstructed from the observed solution. Instead, the distances inside a field were neglected. All the data has been extracted with the QGIS software.

Alongside the real-world data, a set of synthetic instances is generated to compare the ILP formulation with the B&C approach. Starting from the real-world instance (comprising 7 fields) and the depot location, additional instances are obtained by either removing or adding fields. In particular, instances with 4, 5, and 6 fields are created by randomly removing fields from the original set. Moreover, larger synthetic instances are constructed by adding new fields, each characterized by a prescribed demand (d_i) and a distance-cost matrix (c_{ij}) satisfying the triangle inequality. Following this procedure, instances with 8 and 9 fields were generated.

The spreader has a capacity Q equal to 1000 kg. Refilling operations can be carried out using bags of $R = 500$ kg. The fertilizer spreader is a dual-sided device capable of simultaneously treating two adjacent vine rows. When one row is shorter than the other, the driver can independently deactivate one side of the spreader while continuing the operation on the other, ensuring that fertilizer is applied precisely. As already stated, due to operational constraints, each refill must be exactly R or $Q = 2R$; partial refilling is not possible.

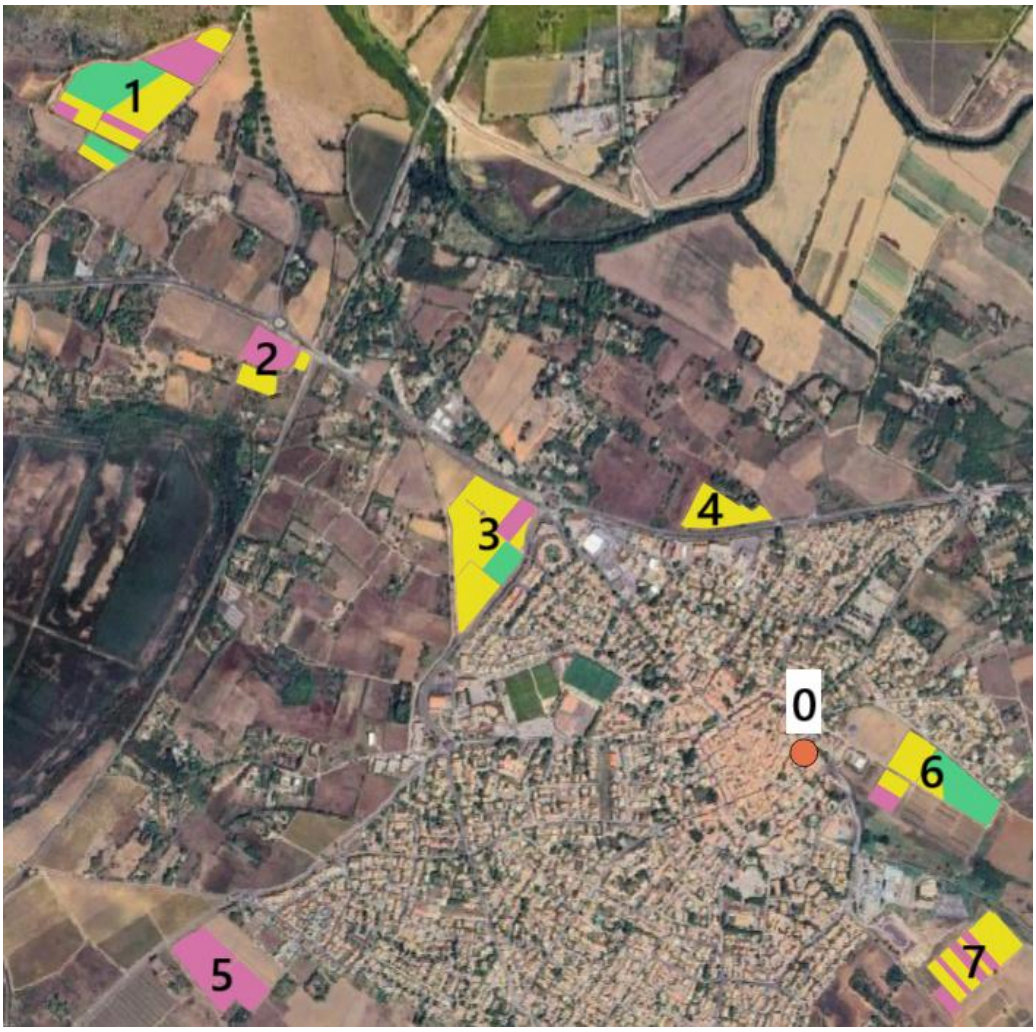

Figure 2.1: Map showing the location of the 7 fields of the estate.

2.7.2 Comparison between ILP formulation and B&C approach

Table 2.1 reports a computational comparison between the proposed ILP formulation and the Branch-and-Cut (B&C) method on a set of synthetic instances, measured in terms of the number of fields. Each row corresponds to one instance. The objective values (Column 2) confirm that, whenever the search is not truncated by the time limit, both methods attain the same optimal solution. For the smaller instances (4 and 5 fields), the ILP formulation exhibits shorter running times than B&C. In contrast, for the instance with 8 and 9 fields the ILP approach reaches the imposed time limit (3600 s) without completing the search, whereas B&C terminates substantially earlier.

The relative difference between the computational times of the ILP and B&C approaches is summarized by the time gap reported in Column 6, defined as the difference

between the ILP and B&C running times normalized by the ILP running time. This indicator highlights a clear shift in performance as the instance size increases: the ILP formulation dominates on smaller instances, whereas the Branch-and-Cut approach becomes more advantageous as the number of fields grows. A crucial driver of the effectiveness of B&C is the strength of the bound available at the root node (as explained in Section 2.5.2). As shown in Column 5, the root node value is already extremely close to the optimum; this is further emphasized by the root gap reported in the last column, which remains essentially negligible across all instances. Such a tight initial bound enhances pruning during the enumeration process, thereby improving the overall efficiency of the B&C scheme.

It is stressed that the present experimental campaign is intentionally limited, both in the number of test instances and in the range of instance sizes; its purpose is to provide an initial validation and a baseline for subsequent, larger computational studies.

		ILP	B&C		Time GAP	Root GAP
# Fields	Value (m)	Time (s)	Time (s)	Root Value (m)	(%)	(%)
4	32276.12	0.57	4.08	32276.12	-615.79	0.00
5	36586.41	6.31	9.80	36417.27	-55.31	-0.46
6	37670.51	60.59	27.08	37595.44	53.41	-0.20
7	40514.73	241.41	175.40	40434.11	27.34	-0.20
8	41491.79	3600	257.2	41410.28	92.86	-0.20
9	44254.50	3600	61.52	44150.99	98.29	-0.23

Table 2.1: ILP vs Branch-and-Cut (B&C) comparison on synthetic instances.

2.7.3 Results for the Post-processing algorithm

Figure 2.2 shows the optimal solution found by both ILP and B&C on real-world data (7 fields) described above. It contains 8 trips that must be sequentially performed by the tractor. Table 2.2 compares the optimal solution with the observed current solution performed by the driver. For both solutions, the number of trips, the total traveled distance, and the manure residue at the end of all the trips are reported. The amount of new fertilizer loaded in the tank at the beginning of each trip is also provided (see Column 3 for both the Tables). Note that, in the observed solution, the spreader is loaded up to its maximum capacity (i.e., $Q = 1000$ kg) at the beginning of each trip (implying that the tank is empty at the end of trips 1-6), while in the solution provided by the model the spreader is only partially loaded with $R = 500$ kg of new fertilizer at the beginning of trips 3 and 6. The solution of the model requires a smaller total traveled distance, satisfies all the field demands, and optimizes the use of the available capacity of the spreader per trip (with a smaller fertilizer residue at the end of the overall fertilization process). Indeed, the solution provided by the model requires a higher number of trips (8 vs 7 trips), but a smaller total traveled distance (12 km shorter) and a smaller residual manure (about 168 kg with respect to 668 kg of the observed solution), demonstrating a significant improvement in operational efficiency.

The deployment of the routings of the model into operational plans is performed by the Post-processing algorithm (introduced in Section 2.6) , which is able to: (i) determine the number of rows to treat for each visited field of each trip, ensuring operational feasibility; (ii) provide a working schedule, obtained by grouping consecutive trips into working days, in such a way that the maximum daily working time of 8 hours is respected. To illustrate the value of this information in helping the operator optimize the site, Figure 2.3 provides an example of a field showing, for each trip in which the field is visited (i.e., Trips 1, 2 and 4) the rows to treat. Rows treated in Trips 1, 2 and 4 are highlighted in blue, red, and green, respectively. For each trip, it is possible to see the last row partially treated, which is the point where the tractor begins fertilizing again in the subsequent trip to the field. Figure 2.4 reports the schedule of the tractor for each working day, as stated by the objective (ii).

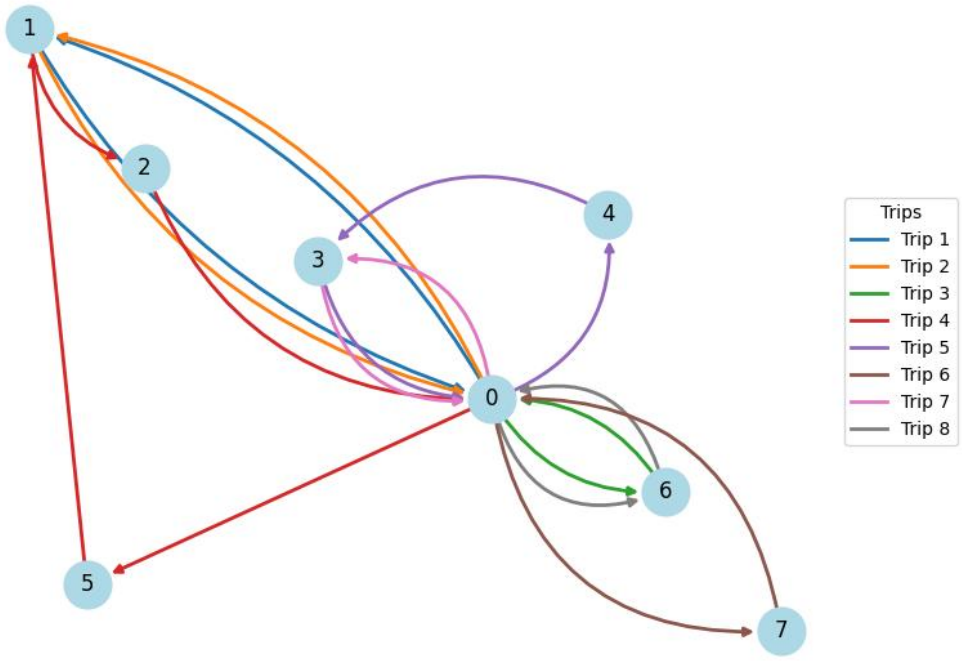

Figure 2.2: Trips of the optimal solution on the graph G. The graph shows the 8 trips of the optimal solution, the nodes represent the 7 fields (node 0 denotes the depot) and the arcs indicate the logical routing sequence.

(a) Optimal Solution

Trips	Distance (m)	Load (kg)
$0 \to 1 \to 0$	8227.18	1000
$0 \to 1 \to 0$	8227.18	1000
$0 \to 6 \to 0$	627.06	500
$0 \to 5 \to 1 \to 2 \to 0$	11142.72	1000
$0 \to 4 \to 3 \to 0$	4517.47	1000
$0 \to 7 \to 0$	2844.22	500
$0 \to 3 \to 0$	4301.84	1000
$0 \to 6 \to 0$	627.06	1000
Total	40514.73	
Total manure residue (kg)		168.02

(b) Observed Solution

Trips	Distance (m)	Load (kg)
$0 \to 4 \to 3 \to 0$	4517.47	1000
$0 \to 3 \to 2 \to 0$	6009.69	1000
$0 \to 2 \to 6 \to 0$	6135.37	1000
$0 \to 6 \to 7 \to 5 \to 0$	8304.28	1000
$0 \to 5 \to 1 \to 0$	11056.61	1000
$0 \to 1 \to 0$	8227.18	1000
$0 \to 1 \to 0$	8227.18	1000
Total	52477.78	
Total manure residue (kg)		668.02

Table 2.2: Comparison between (a) the optimal solution obtained from the MILP model and (b) the observed current solution performed by the driver.

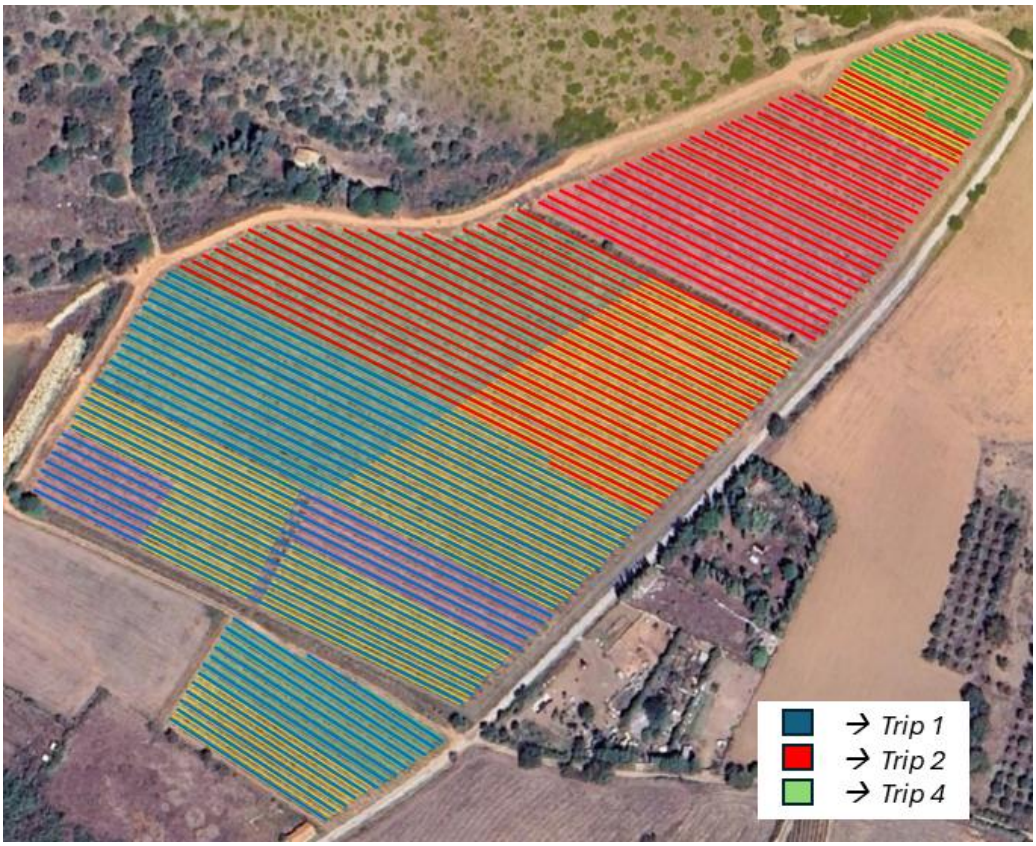

Figure 2.3: Treatment of Field 1 in Trip 1, 2 and 4 as proposed by the Post-processing algorithm. In each trip, where Field 1 is visited, a specific number of rows must be covered, including a final row that is only partially treated. Rows highlighted in blue, red, and green represent Trip 1, Trip 2, and Trip 4, respectively.

Day 1 – Total working time: 7 h 5 min

- **Trip 1:** Load 1000 kg *(Route: 0 → 1 → 0)*
  - Field 1: 37 rows, last row treated for 222.55 m
  - Return to depot with 0.0 kg
- **Trip 2:** Load 1000 kg *(Route: 0 → 1 → 0)*
  - Field 1: 52 rows, last row treated for 67.11 m
  - Return to depot with 46.69 kg
- **Trip 3:** Load 500 kg *(Route: 0 → 6 → 0)*
  - Field 6: 45 rows, last row treated for 174.95 m
  - Return to depot with 0.0 kg

Day 2 – Total working time: 7 h 44 min

- **Trip 4:** Load 1000 kg *(Route: 0 → 5 → 1 → 2 → 0)*
  - Nodes completed: 5, 1, 2
  - Return to depot with 0.0 kg
- **Trip 5:** Load 1000 kg *(Route: 0 → 4 → 3 → 0)*
  - Node completed: 4
  - Field 3: 13 rows, last row treated for 24.39 m
  - Return to depot with 124.32 kg
- **Trip 6:** Load 500 kg *(Route: 0 → 7 → 0)*
  - Node completed: 7
  - Return to depot with 0.0 kg

Day 3 – Total working time: 4 h 2 min

- **Trip 7:** Load 1000 kg *(Route: 0 → 3 → 0)*
  - Node completed: 3
  - Return to depot with 135.37 kg
- **Trip 8:** Load 500 kg *(Route: 0 → 6 → 0)*
  - Node completed: 6
  - Return to depot with 168.02 kg

Figure 2.4: Optimal Daily Tractor Schedule determined by the Post-processing algorithm.

2.8 Conclusion

This chapter focused on a Split Delivery Vehicle Routing Problem (SDVRP), arising in an agricultural logistics context, consisting in planning fertilization operations, under capacity, demand, and operational constraints. The problem setting captures key features of real-world agricultural distribution: field demands vary across locations, refilling at the depot is performed in a discrete (binary) manner, and split deliveries are therefore necessary to guarantee feasibility while preserving efficient routing decisions.

Two exact optimization approaches were investigated and compared. An Integer Linear Programming (ILP) formulation was first introduced as a baseline model, providing a compact and interpretable representation of the problem. A Branch-and-Cut (B&C) algorithm was then developed to strengthen the solution process through the dynamic generation of valid inequalities. Computational experiments on both real-world and synthetic instances show that, on small instances, the ILP formulation is computationally efficient, while the B&C approach exhibits superior performance as instance size increases, benefiting from strong bounds obtained at the root node. It is worth noting that the proposed B&C approach is still at an early stage of development and will be the subject of further research. In particular, future work will focus on designing dedicated heuristics to compute high-quality initial feasible solutions to start the B&C scheme. Overall, the present study should be regarded as preliminary, and the computational results reported here provide a first baseline for subsequent methodological refinements.

To bridge the gap between optimization output and practical implementation, a dedicated Post-processing algorithm was also proposed. This procedure translates the routing solution into executable fertilization plans and day-by-day schedules, ensuring operational feasibility in the real-world case study. As a result, the modelling framework does not remain at a purely theoretical level, but directly supports the deployment of optimization-based decisions in an agricultural setting.

Overall, the chapter demonstrates that exact optimization techniques for SDVRP can effectively support logistics planning in agriculture, even in the presence of operational complexities such as split deliveries, heterogeneous field demands, and binary refilling at the depot. The results highlight both the potential and the current computational limitations of exact methods, motivating further research on alternative formulations and heuristic strategies. In this respect, the proposed models provide a solid methodological foundation for the development of scalable and decision-oriented optimization tools for agricultural logistics.

Bibliography

[1] R. Pérez-Escamilla, "Food security and the 2015–2030 sustainable development goals: From human to planetary health," *Current Developments in Nutrition*, vol. 1, no. 7, p. e000513, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2475299122144720

[2] O. Boyabatlı, J. Nasiry, and Y. H. Zhou, "Crop planning in sustainable agriculture: Dynamic farmland allocation in the presence of crop rotation benefits," *Management Science*, vol. 65, no. 5, pp. 2060–2076, 2019. [Online]. Available: https://doi.org/10.1287/mnsc.2018.3044

[3] L. M. Butkevičienė, L. Skinulienė, I. Auželienė, V. Bogužas, R. Pupalienė, and V. Steponavičienė, "The influence of long-term different crop rotations and monoculture on weed prevalence and weed seed content in the soil," *Agronomy*, vol. 11, no. 7, 2021. [Online]. Available: https://www.mdpi.com/2073-4395/11/7/1367

[4] J. Schöning, P. Wachter, and D. Trautz, "Crop rotation and management tools for every farmer?: The current status on crop rotation and management tools for enabling sustainable agriculture worldwide," *Smart Agricultural Technology*, vol. 3, p. 100086, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S277237552200051X

[5] European Union, "Regulation (EU) No 1307/2013 of the European Parliament and of the Council," 2013, [Accessed July 5, 2024]. [Online]. Available: http://data.europa.eu/eli/reg/2013/1307/2023-01-01

[6] K. G. Grunert, S. Hieke, and J. Wills, "Sustainability labels on food products: Consumer motivation, understanding and use," *Food Policy*, vol. 44, pp. 177–189, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306919213001796

[7] European Commission, "Approved 28 cap strategic plans 2023–2027," 2023, accessed 10 November 2024. [Online]. Available: https://agriculture.ec.europa.eu/system/files/2023-06/approved-28-cap-strategic-plans-2023-27.pdf

[8] G. Pe'er, A. Bonn, H. Bruelheide, P. Dieker, N. Eisenhauer, P. H. Feindt, G. Hagedorn, B. Hansjürgens, I. Herzon, Lomba, E. Marquard, F. Moreira, H. Nitsch, R. Oppermann, A. Perino, N. Röder, C. Schleyer, S. Schindler, C. Wolf, Y. Zinngrebe, and S. Lakner, "Action needed for the eu common agricultural policy to address sustainability challenges," *People and Nature*, vol. 2, no. 2, pp. 305–316, 2020. [Online]. Available: https://besjournals.onlinelibrary.wiley.com/doi/abs/10.1002/pan3.10080

[9] R. Finger, A. Fabry, M. Kammer, J. Candel, T. Dalhaus, and E. M. Meemken, "Farmer protests in europe 2023–2024," *EuroChoices*, vol. 23, no. 3, pp. 59–63, 2024. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/1746-692X.12452

[10] M. Benini, E. Blasi, P. Detti, and L. Fosci, "Solving crop planning and rotation problems in a sustainable agriculture perspective," *Computers Operations Research*, vol. 159, p. 106316, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0305054823001806

[11] W. Britz, M. van Ittersum, A. Oude Lansink, and T. Heckelei, "Tools for integrated assessment in agriculture. state of the art and challenges," *Bio-Based and Applied Economics*, vol. 1, no. 2, pp. 125–150, 2012. [Online]. Available: https://doi.org/10.13128/BAE-11232

[12] J. D. van der Ploeg, D. Barjolle, J. Bruil, G. Brunori, L. M. Costa Madureira, J. Dessein, Z. Drag, A. Fink-Kessler, P. Gasselin, M. Gonzalez de Molina, K. Gorlach, K. Jürgens, J. Kinsella, J. Kirwan, K. Knickel, V. Lucas, T. Marsden, D. Maye, P. Migliorini, P. Milone, E. Noe, P. Nowak, N. Parrott, A. Peeters, A. Rossi, M. Schermer, F. Ventura, M. Visser, and A. Wezel, "The economic potential of agroecology: Empirical evidence from europe," *Journal of Rural Studies*, vol. 71, pp. 46–61, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0743016718314608

[13] A. Wezel, M. Casagrande, F. Celette, J.-F. Vian, A. Ferrer, and J. Peigné, "Agroecological practices for sustainable agriculture. a review," *Agronomy for Sustainable Development*, vol. 34, no. 1, pp. 1–20, 2014. [Online]. Available: https://doi.org/10.1007/s13593-013-0180-7

[14] R. Baldoni and L. Giardini, *Coltivazioni Erbacee*. Bologna: Pàtron, 2002, ('Herbaceous Cultivations').

[15] Y. A. Yigezu, T. El-Shater, M. Boughlala, Z. Bishaw, A. A. Niane, F. Maalouf, W. T. Degu, J. Wery, M. Boutfiras, and A. Aw-Hassan, "Legume-based rotations have clear economic advantages over cereal monocropping in dry areas," *Agronomy for Sustainable Development*, vol. 39, no. 6, p. 58, 2019. [Online]. Available: https://doi.org/10.1007/s13593-019-0602-2

[16] Anon, "Decreto ministeriale 18 luglio 2018, n. 6793 modificato da dm 9 aprile 2020, n. 3775," 2022. [Online]. Available: https://www.gazzettaufficiale.it/eli/id/2020/06/22/20A03222/sg

[17] Ministerial Decree, "Ministerial decree n. 6793, 18 july 2018, modified by ministerial decree n. 3775, 9 april 2020," 2020, accessed July 5, 2024. [Online]. Available: https://www.gazzettaufficiale.it/eli/id/2020/06/22/20A03222/sg

[18] U. Bonciarelli, A. Onofri, P. Benincasa, M. Farneselli, M. Guiducci, E. Pannacci, G. Tosti, and F. Tei, "Long-term evaluation of productivity, stability and sustainability for cropping systems in mediterranean rainfed conditions," *European Journal of Agronomy*, vol. 77, pp. 146–155, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1161030116300387

[19] Barilla Group, "Carta del mulino - regole," https://www.mulinobianco.it/lacartadelmulino/Carta-del-mulino-regole.pdf, 2024, [Accessed July 5, 2024].

[20] O. Ahumada and J. R. Villalobos, "Application of planning models in the agri-food supply chain: A review," *European Journal of Operational Research*, vol. 196, no. 1, pp. 1–20, 2009. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0377221708001987

[21] G. Behzadi, M. J. O'Sullivan, T. L. Olsen, and A. Zhang, "Agribusiness supply chain risk management: A review of quantitative decision models," *Omega*, vol. 79, pp. 21–42, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0305048316302742

[22] J. J. Glen, "Feature article—mathematical models in farm planning: A survey," *Operations Research*, vol. 35, no. 5, pp. 641–666, 1987. [Online]. Available: https://doi.org/10.1287/opre.35.5.641

[23] T. J. Lowe and P. V. Preckel, "Decision technologies for agribusiness problems: A brief review of selected literature and a call for research," *Manufacturing & Service Operations Management*, vol. 6, no. 3, pp. 201–208, 2004. [Online]. Available: https://doi.org/10.1287/msom.1040.0051

[24] L. Alfandari, J. L. Lemalade, A. Nagih, and G. Plateau, "A mip flow model for crop-rotation planning in a context of forest sustainable development," *Annals of Operations Research*, vol. 190, no. 1, pp. 149–164, 2011. [Online]. Available: https://doi.org/10.1007/s10479-009-0553-0

[25] S. Dogliotti, W. Rossing, and M. van Ittersum, "rotat, a tool for systematically generating crop rotations," *European Journal of Agronomy*, vol. 19, no. 2, pp. 239–250, 2003. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1161030102000473

[26] L. Alfandari, A. Plateau, and X. Schepler, "A branch-and-price-and-cut approach for sustainable crop rotation planning," *European Journal of Operational Research*, vol. 241, no. 3, pp. 872–879, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0377221714008558

[27] L. M. Santos, P. Munari, A. M. Costa, and R. H. Santos, "A branch-price-and-cut method for the vegetable crop rotation scheduling problem with minimal plot sizes," *European Journal of Operational Research*, vol. 245, no. 2, pp. 581–590, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0377221715002428

[28] L. M. R. dos Santos, P. Michelon, M. N. Arenales, and R. H. S. Santos, "Crop rotation scheduling with adjacency constraints," *Annals of Operations Research*, vol. 190, no. 1, pp. 165–180, 2011. [Online]. Available: https://doi.org/10.1007/s10479-008-0478-z

[29] D. Forrester and J. Rodríguez, "An integer programming approach to crop rotation planning at an organic farm," *The UMAP Journal*, vol. 39, no. 3, pp. 213–232, 2018. [Online]. Available: https://doi.org/10.1007/S10100-021-00751-8

[30] T. Nordblom and E. Thomson, "A whole-farm model based on experimental flocks and crop rotations in northwest syria," ICARDA, Aleppo, Syria, Tech. Rep. ICARDA-102En, January 1987. [Online]. Available: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2147573

[31] N. K. Detlefsen and A. L. Jensen, "Modelling optimal crop sequences using network flows," *Agricultural Systems*, vol. 94, no. 2, pp. 566–572, 2007. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0308521X07000108

[32] M. E. Salassi, M. A. Deliberto, and K. M. Guidry, "Economically optimal crop sequences using risk-adjusted network flows: Modeling cotton crop rotations in the southeastern United States," *Agricultural Systems*, vol. 118, no. C, pp. 33–40, 2013. [Online]. Available: https://ideas.repec.org/a/eee/agisys/v118y2013icp33-40.html

[33] A. Galán-Martín, C. Pozo, G. Guillén-Gosálbez, A. Antón Vallejo, and L. Jiménez Esteller, "Multi-stage linear programming model for optimizing cropping plan decisions under the new common agricultural policy," *Land Use Policy*, vol. 48, pp. 515–524, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0264837715002008

[34] L. M. R. dos Santos, A. M. Costa, M. N. Arenales, and R. H. S. Santos, "Sustainable vegetable crop supply problem," *European Journal of Operational Research*, vol. 204, no. 3, pp. 639–647, 2010. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0377221709008996

[35] G. Regis Mauri, "Improved mathematical model and bounds for the crop rotation scheduling problem with adjacency constraints," *European Journal of Operational Research*,

vol. 278, no. 1, pp. 120–135, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0377221719303364

[36] I. Fikry, A. Eltawil, and M. Gheith, "A robust crop rotation optimization model with water scarcity and net return uncertainty considerations," *IEEE Access*, vol. 9, pp. 128 938–128 950, 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9539176

[37] W. Haneveld and A. Stegeman, "Crop succession requirements in agricultural production planning," *European Journal of Operational Research*, vol. 166, no. 2, pp. 406–429, 2005. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0377221704002358

[38] J. Yang, W. Zhang, M. Li, H. Chen, and Y. Wang, "Optimization of crop cultivation based on integer programming methods," *Agriculture*, vol. 15, no. 2, p. 250, 2025. [Online]. Available: https://hsetdata.com/index.php/ojs/article/view/586

[39] V. M. Albornoz, M. I. Véliz, R. Ortega, and V. Ortíz-Araya, "Integrated versus hierarchical approach for zone delineation and crop planning under uncertainty," *Annals of Operations Research*, vol. 286, no. 1, pp. 617–634, 2020. [Online]. Available: https://doi.org/10.1007/s10479-019-03198-y

[40] A. Biswas and B. B. Pal, "Application of fuzzy goal programming technique to land use planning in agricultural system," *Omega*, vol. 33, no. 5, pp. 391–398, 2005, oR and its Applications. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0305048304000970

[41] C. Filippi, R. Mansini, and E. Stevanato, "Mixed integer linear programming models for optimal crop selection," *Computers Operations Research*, vol. 81, pp. 26–39, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0305054816303033

[42] J. Li *et al.*, "Crop rotation model for contract farming with constraints on similar profits," *Computers and Electronics in Agriculture*, vol. 119, pp. 12–18, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0168169915003038?via%3Dihub

[43] F. Capitanescu *et al.*, "Multi-stage farm management optimization under environmental and crop rotation constraints," *Journal of Cleaner Production*, vol. 147, pp. 197–205, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0959652617300835?via%3Dihub

[44] R. Cortignani and G. Dono, "Greening and legume-supported crop rotations: An impacts assessment on italian arable farms," *Science of the Total Environment*, vol. 734, p. 139464, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0048969720329818?via%3Dihub

[45] European Commission, "Key policy objectives of the cap 2023–2027," https://agriculture.ec.europa.eu/common-agricultural-policy/cap-overview/cap-2023-27/key-policy-objectives-cap-2023-27_en#briefs, 2023, accessed 6 November 2024.

[46] EU CAP Network, "Policy insights: Cap green architecture components," http://eu-cap-network.ec.europa.eu/thematic-group-green-architecture-designing-green-strategies_en, 2023, accessed 6 November 2024.

[47] European Parliament and Council of the European Union, "Regulation (eu) 2021/2115 of the european parliament and of the council of 2 december 2021," https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32021R2115, 2021, accessed 6 November 2024.

[48] Italian Ministry of Agriculture, Food and Forestry Policies, "Strategic plan for the cap (psp) italy 2023–2027," https://www.reterurale.it/downloads/PSP_Italia_151120221_Def.pdf, 2022, accessed 6 November 2024.

[49] European Commission, "European farmers exempted from rules on land lying fallow," https://ec.europa.eu/commission/presscorner/api/files/document/print/en/ip_24_781/IP_24_781_EN.pdf, 2024, accessed 6 November 2024.

[50] European Parliament and Council of the European Union, "Regulation (eu) 2024/1468 amending regulations (eu) 2021/2115 and (eu) 2021/2116 on the common agricultural policy (cap)," https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=OJ%3AL_202401468, 2024, accessed 6 November 2024.

[51] European Parliament and Council of the European Union, "Regulation (eu) 2024/1468 of the european parliament and of the council of 14 may 2024," https://eur-lex.europa.eu/eli/reg/2024/1468/oj/eng, 2024, accessed 6 November 2024.

[52] Ministero dell'Agricoltura, della Sovranità Alimentare e delle Foreste, "Decreto ministeriale 28 giugno 2024, prot.n.289235 – attuazione del regolamento (ue) 2024/1468," https://www.arpea.piemonte.it/sites/default/files/documentazione/Normative/Nazionale/MASAF_2024_0289235_DMsemplificazionePAC2024postCSR28giu2024_signed.pdf, Jun. 2024, accessed 6 November 2024.

[53] L. Fosci, "Main role of agro-industries in the agroecological transition: public policies and private initiatives," Ph.D. dissertation, Università degli Studi della Tuscia, Italy, 2022.

[54] V. L. de Lima, C. Alves, F. Clautiaux, M. Iori, and J. M. de Carvalho, "Arc flow formulations based on dynamic programming: Theoretical foundations and applications," *European Journal of Operational Research*, vol. 296, no. 1, pp. 3–21, 2022. [Online]. Available: https://doi.org/10.1016/j.ejor.2021.04.024

[55] M. Gatti, C. Squeri, A. Garavani, T. Frioni, P. Dosso, I. Diti, and S. Poni, "Effects of variable-rate nitrogen application on cv. Barbera performance: I. vegetative growth and leaf nutritional status," *Journal of Viticulture and Enology*, vol. 22, no. 3, pp. 15–24, 2018.

[56] M. Gatti, C. Squeri, A. Garavani, T. Frioni, P. Dosso, I. Diti, and S. Poni, "Effects of variable-rate nitrogen application on cv. Barbera performance: Yield and grape composition," *American Journal of Enology and Viticulture*, vol. 70, no. 2, pp. 188–200, 2019. [Online]. Available: https://doi.org/10.5344/ajev.2019.18072

[57] A. Guerrero and A. M. Mouazen, "Evaluation of variable rate nitrogen fertilization scenarios in cereal crops from economic, environmental and technical perspective," *Soil and Tillage Research*, vol. 213, p. 105110, 2021. [Online]. Available: https://doi.org/10.1016/j.still.2021.105110

[58] M. Dror and P. Trudeau, "Savings by split delivery routing," *Transportation Science*, vol. 23, no. 2, pp. 141–145, 1989. [Online]. Available: https://doi.org/10.1287/trsc.23.2.141

[59] C. Archetti, N. Bianchessi, and M. G. Speranza, "A column generation approach for the split delivery vehicle routing problem," *Networks*, vol. 58, no. 3, pp. 241–254, 2011. [Online]. Available: https://doi.org/10.1002/net.20467

[60] M. Dror and P. Trudeau, "Split delivery routing," *Naval Research Logistics*, vol. 37, no. 3, pp. 383–402, 1990. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1002/nav.3800370304

[61] C. Archetti, M. W. P. Savelsbergh, and M. G. Speranza, "The split delivery vehicle routing problem: A survey," *Discrete Applied Mathematics*, vol. 232, pp. 275–292, 2017. [Online]. Available: https://link.springer.com/chapter/10.1007/978-0-387-77778-8_5

[62] M. Dror, P. Trudeau, and J. D. Laporte, "Vehicle routing with split deliveries," *Discrete Applied Mathematics*, vol. 50, no. 3, pp. 239–254, 1994. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0166218X9200172I?via%3Dihub

[63] L. Gouveia, M. Leitner, and M. Ruthmair, "A branch-and-cut algorithm for the multi-depot split delivery vehicle routing problem," *Transportation Research Part B: Methodological*, vol. 170, pp. 1–25, 2023. [Online]. Available: https://optimization-online.org/wp-content/uploads/2021/10/mdsdvrp_TR.pdf

[64] N. Bianchessi, G. Righini, and R. W. Calvo, "A branch-and-cut-and-price algorithm for the split delivery vehicle routing problem with time windows," *Transportation Science*, vol. 50, no. 2, pp. 620–636, 2016.

[65] P. Munari and M. Savelsbergh, "Compact formulations for split delivery routing problems," *Transportation Science*, vol. 56, no. 4, pp. 1022–1043, 2022. [Online]. Available: https://doi.org/10.1287/trsc.2021.1106

[66] C. Archetti, N. Bianchessi, and M. G. Speranza, "Branch-and-cut algorithms for the split delivery vehicle routing problem," *European Journal of Operational Research*, vol. 238, no. 3, pp. 685–698, 2014. [Online]. Available: https://doi.org/10.1016/j.ejor.2014.04.026

[67] M. Gamst, R. Lusby, and S. Ropke, "Exact and heuristic methods for the split delivery vehicle routing problem," *Transportation Science*, vol. 58, 2024. [Online]. Available: https://doi.org/10.1287/trsc.2022.0353

[68] J. M. Belenguer, M. C. Martínez, and E. Mota, "A lower bound for the split delivery vehicle routing problem," *Operations Research*, vol. 48, no. 5, pp. 801–810, 2000. [Online]. Available: https://doi.org/10.1287/opre.48.5.801.12407

[69] D. Naddef and G. Rinaldi, "Branch-and-cut algorithms for the capacitated vrp," pp. 53–84. [Online]. Available: https://epubs.siam.org/doi/abs/10.1137/1.9780898718515.ch3