UNIVERSIDADE ESTADUAL DE CAMPINAS
Faculdade de Engenharia Elétrica e de Computação

Bruno Santos de Miranda

Optimization Techniques in Agriculture: The Crop Rotation Problem

Técnicas de Otimização na Agricultura: O Problema de Rotação de Culturas

Campinas

2020

Bruno Santos de Miranda

Optimization Techniques in Agriculture: The Crop Rotation Problem

Técnicas de Otimização na Agricultura: O Problema de Rotação de Culturas

Dissertation presented to the School of Electrical and Computer Engineering of the University of Campinas in partial fulfillment of the requirements for the degree of Master in Electrical Engineering, in Automation area.

Dissertação apresentada à Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas como parte dos requisitos exigidos para a obtenção do título de Mestre em Engenharia Elétrica, na Área de Automação.

Supervisor: Prof. Dr. Akebo Yamakami

Co-supervisor: Profa. Dra. Priscila Cristina Berbert Rampazzo

Este trabalho corresponde à versão final da dissertação defendida pelo aluno Bruno Santos de Miranda, orientada pelo Prof. Dr. Akebo Yamakami.

Campinas

2020

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca da Área de Engenharia e Arquitetura
Luciana Pietrosanto Milla - CRB 8/8129

M672o

Miranda, Bruno Santos de, 1994-
Optimization techniques in agriculture : the crop rotation problem / Bruno Santos de Miranda. – Campinas, SP : [s.n.], 2020.

Orientador: Akebo Yamakami.
Coorientador: Priscila Cristina Berbert Rampazzo.
Dissertação (mestrado) – Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Otimização multiobjetivo. 2. Rotação de cultivos. 3. Agricultura. 4. Algoritmos evolutivos. 5. Algoritmos genéticos. I. Yamakami, Akebo, 1947-. II. Rampazzo, Priscila Cristina Berbert, 1984-. III. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. IV. Título.

Informações para Biblioteca Digital

Título em outro idioma: Técnicas de otimização na agricultura : o problema de rotação de culturas
Palavras-chave em inglês:
Multiobjective optimization
Crop rotation
Agriculture
Evolutionary algorithms
Genetic algorithms
Área de concentração: Automação
Titulação: Mestre em Engenharia Elétrica
Banca examinadora:
Priscila Cristina Berbert Rampazzo [Coorientador]
Anibal Tavares de Azevedo
Kelly Cristina Poldi
Data de defesa: 13-03-2020
Programa de Pós-Graduação: Engenharia Elétrica

Identificação e informações acadêmicas do(a) aluno(a)
- ORCID do autor: https://orcid.org/0000-0003-0889-606X
- Currículo Lattes do autor: http://lattes.cnpq.br/1295317100536487

COMISSÃO JULGADORA - DISSERTAÇÃO DE MESTRADO

Candidato: Bruno Santos de Miranda RA: 211491
Data de Defesa: 13 de Março de 2020
Título da Tese: "Optimization Techniques in Agriculture: The Crop Rotation Problem (Técnicas de Otimização na Agricultura: O Problema de Rotação de Culturas)"

Profa. Dra. Priscila Cristina Berbert Rampazzo (Presidente)
Profa. Dr. Anibal Tavares de Azevedo
Profa. Dra. Kelly Cristina Poldi

A Ata de Defesa, com as respectivas assinaturas dos membros da Comissão Julgadora, encontra-se no SIGA (Sistema de Fluxo de Dissertação/Tese) e na Secretaria de Pós-Graduação da Faculdade de Engenharia Elétrica e de Computação.

Acknowledgements

My deep gratitude goes first to Professor Akebo and Professor Priscila, whose skillful assistance and guidance have been essential.

Above ground, I would like to thank my dear mother, who, although no longer with us, continuous to inspire by her example and kindness. My appreciation also extends to all my family; especially my dear father, my brother and loving sister.

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

Finally, I would like to thank God Almighty for giving me the opportunity and strength to undertake this challenging journey.

"The people that walked in darkness have seen a great light:
they that dwell in the land of the shadow of death,
upon them hath the light shined."
(Holy Bible, Isaiah 9, 2)

Abstract

Crop rotation is the future of sustainable agriculture. Diversity in the cropping sequence can improve soil physical and chemical properties without demanding all the conventional tillage practices or large amounts of agricultural chemicals. Growing cover crops along the rotation also plays a fundamental role in controlling pests and weeds, improving soil fertility and reducing erosion. Although we have focused on bringing about more sustainable agrarian practices, farms ought to be profitable and resilient to thrive in an uncertain future. Therefore, planning crop rotations needs to balance the potential economic scenarios and the environmental conservation, which optimization techniques can manage this balance naturally. Our main effort in this research is to develop the crop rotation's concepts in the optimization perspective. After carefully considering the nutrient flow in agricultural fields and many advantages of seeding cover crops, we have proposed new models for the Crop Rotation Problem (CRP). Our research proceeds with evaluating optimization techniques for the CRP and proposing new alternatives. From classical methodologies, we have analyzed multiobjective optimization methods such as the weighted sum and the achievement scalarizing function technique. Looking for more efficient methods, evolutionary algorithms (EAs), which are based on biological evolution, such as genetic inheritance and mutation, are interesting alternatives. We have developed a mono-objective genetic algorithm and a multiobjective one. After running several tests using real data of the CRP, the achieved results confirm that the proposed algorithms have satisfactory performance. This research contributed to the fields of Agriculture, with the proposed models of CRP and Optimization, with the development of evolutionary algorithms.

Keywords: Multiobjective Optimization; Crop Rotation; Agriculture; Evolutionary Algorithms; Genetic Algorithms.

Resumo

Rotação de culturas é o futuro da agricultura sustentável. Diversidade na sequência de rotação melhora as propriedades físicas e químicas do solo sem demandar todas as exaustivas práticas convencionais de manejo do solo ou grandes quantidades de insumos agrícolas. Cultivar plantas de cobertura ao longo da rotação também desempenha um papel fundamental no controle de pestes e ervas daninhas, melhora a fertilidade do solo e reduz os processos erosivos. Embora esta pesquisa concentre-se na promoção de práticas agrícolas mais sustentáveis, as propriedades rurais precisam ser lucrativas e resilientes para prosperar num futuro incerto. Então, o planejamento das rotações de culturas precisa equilibrar os cenários econômicos potenciais e a conservação ambiental, sendo que as técnicas de otimização conseguem realizar este balanço naturalmente. Após considerar o fluxo de nutrientes nos campos cultiváveis e muitas vantagens do cultivo das plantas de rotação, foram propostos novos modelos para o Problema de Rotação de Culturas (PRC). A pesquisa prosseguiu com a avaliação das técnicas de otimização disponíveis para o PRC e com a proposta de novos métodos. Das abordagens clássicas, foram analisados métodos de otimização multiobjetivo, tais como o método da soma ponderada e as técnicas de escalarização. Em busca de métodos mais eficientes, os algoritmos evolutivos (AE), que são baseados na evolução biológica, tais como herança genética e mutação, são alternativas interessantes. Foram desenvolvidos algoritmos genéticos para otimização mono-objetivo e para otimização multiobjetivo. Após a realização de diversos testes utilizando dados reais do PRC, os resultados encontrados confirmam que os algoritmos propostos têm desempenho satisfatório. Esta pesquisa contribuiu para os campos da Agricultura, com os modelos propostos para o PRC, e da Otimização, com o desenvolvimento de algoritmos evolutivos.

Palavras-chaves: Otimização Multiobjetivo; Rotação de Cultivos; Agricultura; Algoritmos Evolutivos; Algoritmos Genéticos.

List of Figures

List of Tables

List of abbreviations and acronyms

CRP	Crop Rotation Problem
GA	Genetic Algorithm
CRQR	Crop Rotation Quality Rating
PLGA	Priority List Genetic Algorithm
PLMGA	Priority List Multiobjective Genetic Algorithm
CRQR	Crop Rotation Quality Rating
USDA	United States Department of Agriculture
NASS	National Agricultural Statistics Service
UNICAMP	Universidade Estadual de Campinas
DM	Decision Maker
ASFM	Achievement Scalarizing Function Method

Contents

Introduction

Crop rotation is an agrarian technique based on switching grown crops on a piece of land, breaking simultaneous crop scheduling from a certain family. There are cyclical rotations, which repeat the same sequence indefinitely and non-cyclical ones that allow changes in the crop sequence, adapting to reach management decisions and evolving as market opportunities appear (MOHLER; JOHNSON, 2009).

The main concepts of the crop rotation technique are presented in Figure 1. Splitting the farm's available area into several plots is the first step. Which plot would have its own crop's history and particular soil practices. Figure 1 exhibits a general plot distribution that is rich on crop diversity. Considering a group of distinct crops, at a certain period, some of them might be close to harvest when others would have just been seeded.

The plot distribution defines a neighborhood, where neighbors (adjacent cultivable areas) are called *adjacent plots*. In general, it is not desirable to have crops from the same family side-by-side without cycle disruption. It might increase migration of arthropods and other pests from one field to another. Adjacency among plots is a key component in this research.

Figure 1: Crop diversity and rotation. (Source: Plodozmian [1])

Although conventional farming has powerful mechanisms such as pesticides and synthetic fertilizers that ensure high production and control crop diseases, these are unsustainable practices since the demand for resources, such as fertilizers and chemical compounds, has increased gradually. In addition, some agents have turned out resistant,

[1] Available at: <https://pt.wikibooks.org/wiki/A_evolu%C3%A7%C3%A3o_tecnol%C3%B3gica/Navega%C3%A7%C3%A3o,_implementos_e_o_grande_desenvolvimento_da_agricultura#/media/Ficheiro:Plodozmian.jpg>. Access date: January 15th, 2020.

reducing drastically the efficient of regulated pesticides and weed killers. Crop rotation is a critical tool and provides alternatives for controlling pests, building healthy soils, and reaching a sustainable production chain (FORRESTER; RODRIGUEZ, 2018).

Caring about soil properties is a commitment to future generations. Besides being profitable, agribusiness should consider the long-term effects of the agrarian practices on the soil (FINCH; SAMUEL; LANE, 2014). Both soil's physical and chemical properties can be improved with the proper selection of crops and their arrangement along the cropping planning (OUDA; ZOHRY; NORELDIN, 2018).

Diversity in the cropping plan could enhance financial stability in the agrarian sector. Farm's profits would not rely on just one main cash crop, but a diverse set of commercial crops, well-distributed in several periods, which might eventually improve cash flow by introducing regular incomes in the agribusiness. From labor aspect, workers would be less exposed to chemical compounds threatening job quality and safety, once the crop rotation was well-established (SANTOS et al., 2011).

Many farmers have already understood the importance of crop rotations. Some classic strategies such as the cereal-legume rotation are widely spread. Evaluating the whole potential of the crop rotation technique requires long-term data from the soil parcels and about their relationship with the grown crops (FRANCIS, 2005). Advanced technological devices would help monitoring soil characteristics, as they have provided large improvements in agriculture and changed the human interaction with the environment (FAR; REZAEI-MOGHADDAM, 2018).

Site-specific farming has been made attainable by technologies combining global positioning system (GPS) and geographic information systems (GIS). Connecting real-time data collection with accurate position information leads to the efficient manipulation and analysis of large amounts of geospatial data. Without the current technological level, implementing production techniques that consider land variability was impressive hard and also used to limit farmer's ability to develop the best soil practices and boost yields (GPS, 2018). Mapping the fields in detail and real-time monitoring crops are essential elements to develop a trend referred to as *Precision Agriculture* (KAMILARIS; KARTAKOULLIS; PRENAFETA-BOLDú, 2017).

Although we have accounted many benefits from the agrarian innovations, like positioning systems for machine control and digital image processing from the fields, there are still many opportunities to value creation along the farm's supply chain (BRAUN; COLANGELO; STECKEL, 2018).

As we have more information about the plot state and crop features, we are able to extract powerful insights farmers need to make smarter management decisions. Optimization techniques can support the decision maker (DM) and provide the best alternatives. Our daily lives are driven by looking for the best, using all available resources. Although strict academic approach for optimization might be a great deal to take daily decisions, the general concept of optimization is deeply connected with the human thought (VANDERPLAATS, 2007).

Researching about crop rotation sounds highly acclaimed from the agrarian viewpoint due to the desirable effects in the soil properties. From the Operational Research aspect, however, it might be an underrated problem. Surely, logistics and industrial applications would be more usual. Figure 2 tries to broaden the perspective of crop rotations by introducing a farm elevation. We notice several plots that indicate the complexity of farm management. Planning crop rotations in many plots over several years is a great deal harder than in conventional practices.

Figure 2: Contour farming; crop rotations in large farms. (Source: Britannica [2])

The Crop Rotation Problem (CRP) can be classified as a *Scheduling Problem*, which represents a class of decision-making problems critical in optimization. Single-machine scheduling, multi-machine scheduling (identical, uniform and not related machines), flow shop scheduling (machines are arranged in series and jobs cannot be performed in parallel), and job shop (machines are arranged in series and jobs can be performed in parallel) are some samples of Scheduling Problems. Scheduling jobs is critical in many enterprises. Delivering products on time and reducing the consumption of resources are common milestones. Optimization can improve job flow by developing consistent planning schemes.

Metaheuristics have been extensively applied in optimization of Scheduling Problems. Although they do not assure optimal solutions, these algorithms are quite efficient in searching quality alternatives without consuming exhaustive computational

[2] Available at: <https://www.britannica.com/technology/agricultural-technology/images-videos#/media/1/9620/149126>. Access date: January 16th, 2020.

resources. Evolutionary algorithms are recognized by their efficiency and performance, applicability (reaching a large set of optimization problems), and also their capability of solving problems without much previous knowledge in the problem features (KNOWLES; CORNE; DEB, 2008).

Objectives and contributions

Analyzing the CRP from optimization standpoint sets the multidisciplinary nature of this research. The developed subjects and proposed models aim to enhance the crop rotation features by providing a reliable optimization tool-chain for the CRP. Although we are deeply committed to CRP, the discussed optimization methods have a wide range of applications.

We account contributions in the agriculture and optimization fields by developing CRP models that connect many agrarian concepts. The proposed multiobjective model introduced a new perspective in the crop rotation planning. We have discussed and tested classical optimization methods with the CRP. In evolutionary computation, we have proposed novel metaheuristic procedures for mono-objective and multiobjective optimization. A combination of deterministic and stochastic procedures creates a powerful optimization package, reaching a large set of CRP instances and generating reliable solutions.

Research outlines

Chapters ahead follow the developments of this research, oriented by the methodology itself. At the beginning, we have reviewed the literature and introduced the state-of-art related to the CRP. After carefully considering the aspects of CRP, we have proposed new optimization models and detailed their properties, which resumes the first part of this dissertation, mostly engaged in the CRP's attributes. The second part of this research is *almost* entirely practical; although we have mainly presented tests and results in these further chapters, there are plenty of references about the optimization methods along. The last part summarizes our achievements, suggesting future works in research line. A brief description of the chapters and their contents follows:

- Chapter 1 introduces the CRP and presents a survey of related researches.
- Chapter 2 presents the proposed mono-objective model of the CRP.
- Chapter 3 explores the novel multiobjective model of the CRP.
- Chapter 4 details nature-inspired algorithms. In this chapter, we present a new mono-objective algorithm called Priority List Genetic Algorithm (PLGA).

- Chapter 5 analyzes the multiobjective optimization methods. We also present a novel multiobjective genetic algorithm, named Priority List Multiobjective Algorithm (PLMGA).
- Chapter 6 details computational results from PLGA's tests and the proposed mono-objective CRP model.
- Chapter 7 reports computational tests with classical optimization methods in the proposed multiobjective CRP model.
- Chapter 8 exhibits several scenarios and analyzes PLMGA's performance in the proposed multiobjective CRP model.
- Chapter 9 summarizes contributions and presents our final observations and further steps.
- Appendix A, Appendix B and Appendix C demonstrate the proposed CRP databases, which have been fundamental on developing the proposed algorithms.

1 LITERATURE REVIEW ABOUT THE CRP

We have selected some important references to develop our research and we have presented a review in this chapter. In addition, Section 1.1 analyzes some available CRP's models. The researches about the CRP have not been restricted to the agrarian field, but they are multidisciplinary and connected with environmental trends. There are many researches about the CRP and a sustainable agrarian future.

Many types of research have been produced about CRP. They look forward to establish a sustainable production system without compromising profits on the farms. The CRP is also connected to Farming 4.0 in many aspects. As farmers have already brought Industry 4.0 ideas into the field, combining the physical and the virtual world and transforming the value creation in farm management, new terms have emerged such as Farming 4.0 or Agriculture 4.0 defining this contemporaneous trend (BRAUN; COLANGELO; STECKEL, 2018).

Crop rotation is one of the most valuable contemporaneous agrarian practices. It is able to enhance soil health without demanding more external resources. The crop rotation technique consists of developing a crop sequence in a cultivable area that alternates crops from different families (SANTOS et al., 2011) (BEHNKE et al., 2018).

Besides many types of research about the CRP in the agrarian field, proposals from the Operational Research field are more usual nowadays as optimization of the CRP models has turned out quite challenging. Although deterministic methods have provided great results and succeed in many applications, they may exceed reasonable computational time in the large instances of the CRP. Dynamic and interactive algorithms are difficult to develop using deterministic concepts. As exact search algorithms have strict procedures to generate solutions, they lack of flexible, creating opportunities for the stochastic procedures (MEMMAH et al., 2015). It does not mean stochastic procedures are easy or simple to develop computational applications, but, as they are less dependent on the problem characteristics, applications would be more versatile and generic than using deterministic methods.

Dury et al. (2012) presented a complete survey of the models supporting crop rotation strategies and introduced some terminology related to the CRP. Their review indicated that crop rotation models are usually static, and their main goal is to increase farm's profitability. Multiobjective models in the literature are still uncommon. Dury et al. (2012) supported that the CRP should be dynamic, which would enable continuous improvements and changes in the long-term crop sequences, instead of providing a static response and dismissing the evolving scenarios in agriculture.

Establishing crop rotations sustains soil fertility and good tillage. Although expressive results would take longer in degraded soils, the rotation technique applies for any soil type. Bold e Nijloveanu (2016) chose an evolutionary algorithm for generating crop rotations. The objective function in Bold e Nijloveanu (2016) represents the profit maximization. The proposed algorithm encodes data in a two dimensional integer arrays. Crossover operator generates new individuals by interchanging information from a random crossover point, and the mutation operator exchanges positions in the chromosome.

Lin e Hsieh (2013) developed a grain crop rotation model, combining price fluctuations and natural climate changes into the model. The decision-making was based on a stationery Markov process. They discussed a dynamic assessment of grain crop rotation, considering how grain prices are affected by natural disasters such as heavy rains or long droughts.

Technological advances can integrate field management and the crop rotation system even further. Variations in nutrient levels are usual, even in small areas, and conventional farmers rely on tillage procedures and chemical compounds to enhance soil conditions. The proposed model in Atchatha e Jayakumar (2016) presents the crop selection based on the nutrient availability from each cultivable field. There are also market considerations and local demand.

Forrester e Rodriguez (2018) developed an integer 0-1 linear optimization model for organic vegetable crop production. They proposed a set of seeding principles to be achieved. The proposed model in Forrester e Rodriguez (2018) considers soil nutrient depletion, weed control, and the irrigation mechanism available. The objective function and the set of constraints aim to meet the projected market demand.

Imadi et al. (2016) considered the sustainable aspects of the agrarian production chain and discussed methods to reach more sustainable practices in agriculture. Inter-cropping and trap cropping are pointed out as good alternatives for controlling arthropod populations and pests. Integrated pest management with inter-cropping could largely reduce pesticide requirements. By seeding resistant crops, the crop sequence could rise the pest mortality rate due to the lack of resources for its population (GHORBANI et al., 2008).

As food demand is expected to increase worldwide, agrarian intensification will not be enough to reach the equilibrium due to the environmental impact assessments for certain farming activities and the related biodiversity loss. Finding alternative agricultural practices that reduce the dependence on external chemical compounds, such as nitrogen-based fertilizers and pesticides, without compromise production rates challenges farms decision-making and their stakeholders. Land use management could support more sustainable decisions. Memmah et al. (2015) presented a great review of land use optimization, which they have characterized as a complex and hard task, based on many spatial factors, constraints and conflicting goals. Their research about optimization algorithms that generates alternative land use solutions indicated that stochastic approaches could be more efficient and flexible to deal with large and complex optimization problems. They also noticed that deterministic approaches are often inefficient to tackle large-scale combinatorial and nonlinear problems.

Concerned about a large number of consumed resources and the environmental impact of mono-culture farming, Santos et al. (2008) discussed a novel optimization model of the CRP, based on organic farming. They also developed a new column generation algorithm, combined with a greedy heuristic procedure.

The research presented in Aliano, Florentino e Pato (2014) analyzed hybrid metaheuristics algorithms in search of quality solutions for the CRP. Attaining feasibility presented to be a hard task on the proposed model, to overcome this hurdle, the initial population was generated by a heuristic procedure. The hybrid algorithms with local search and with Simulating Annealing presented good results in this related work. Aliano, Florentino e Pato (2018) have presented a bi-objective approach, which acknowledges that profitability and diversity of crop rotation are conflicting goals. The research in Aliano, Florentino e Pato (2018) explored the bi-objective optimization model.

Pavón, Brunelli e Lücken (2009) proposed a multiobjective evolutionary algorithm for the CRP. As long as the CRP is connected to contradictory objectives, they tried to provide a way into the trade-off solutions. Their formulation considers many objectives, such as: minimize the total investment, maximize nutrient reserves in the soil, maximize returns, and increase crop diversity in the crop sequence. They have run tests based on cultivable areas in Paraguay.

The ecosystem and the agribusiness viability are threatened by the production practices that rely only on the external chemical inputs. Rosenzweig, Stromberger e Schipanski (2018) studied the impact of the intensified cropping system in no-till agrarian ecosystems. Crop diversity is often related to the cropping system intensification, which increases plant competition with weeds and may suppress weed production and seed germination. The lack of diversity may result in herbicide dependency. No-till and cropping rotations might reduce weed management costs drastically due to the disrupted weed life cycles and high competition. Rosenzweig, Stromberger e Schipanski, 2018 also pointed out that the cropping system intensification might stimulate nitrogen and phosphorus uptake due to the greater soil microbial activity.

Lukowiak, Grzebisz e Sassenrath (2016) researched phosphorus management in a soil-plant system in long-term cropping sequences with oil-seed rape. Phosphorus availability in the soil system is deeply related to crop production. During cropping stages of development, there are critical uptake intervals, and phosphorus applications should be synchronized with the actual crop demand. In general, flowering is the maximum phosphorus uptake period. Although phosphorus fertilizers can supply cropping requirements, soil reservoirs are the primary source for growing crops.

In farm production planning, farmers are usually worried about succession. Sometimes, considering some crops as successors is not recommended in the same field or, at least, should be avoided. Establishing an inadvisable sequence might deplete soil fertility or turn the following crop susceptible to spread diseases and harsh plagues. Haneveld e Stegeman (2005) discussed the crop rotation requirements from a mathematical standpoint. They have modeled inadmissible sequences in order to avoid their side-effects in the crop scheduling. For instance, neglecting crop's family rotations by succeeding potatoes after potatoes increases nematodes in the soil, and so, this sub-sequence is highly prejudicial. Haneveld e Stegeman (2005) also developed a stationary linear programming model of CRP.

Whole-farm models ought to consider crop rotations due to the significant impact of rotations in the environment and, mainly, in the production. Detlefsen e Jensen (2007) presented a mathematical model based on network strategies. In this formulation, precedent crops are considered as nodes in the graph, and each possible successor is connected with arrows. The size of the graph is proportional to the number of precedent crops. Using network modeling, Detlefsen e Jensen (2007) provided a valuable insight into the CRP.

Zuber et al. (2017) developed a method to evaluate the influence of the long-term intensified cropping system and tillage into the soil environment. They identified and analyzed soil parameters that could define well-established soils and health soil functions.

Franchini et al. (2012) evaluated different agrarian practices in southern Brazil. They observed that crop rotation minimizes soil disturbance, and crop yields were profoundly affected by the cropping system.

1.1 Crop rotation models in the literature

Alfandari et al. (2011) proposed a mixed-integer linear programming model for the CRP. Each region has its own characteristics, and the farm's goals from each area vary to meet their evolving business. Madagascan farmers used to clear fields using fire, which provides fertile soils fast. As clearing forests with fire in order to safe productivity is essentially unsustainable. Their research is an effort against deforestation and they aim to minimize the maximum surface required during a certain planning horizon.

Capitanescu et al. (2017) developed a mixed-integer linear programming model for farm management, which produces a crop rotation sequence. It also aims to maximize profitability while reaching specified environmental constraints.

There are many acclaimed crop rotation approaches in the literature, and they represent an outright perspective of the CRP. In the following subsections, we reviewed in detail some of them, looking forward to compose a solid understanding about the CRP modeling.

1.1.1 Crop Rotation's Model A: Santos

The first model in this review was proposed in Santos et al. (2008). Input parameters are a empty plot set, a total number of periods and a crop set. The list ahead presents the parameters in the model:

- M: number of periods in each rotation (time unit);
- L: number of plots (the total cultivable area split in several fields);
- C: set of growing crops, except green manure crops;
- G: set of green manure crops;
- N: the total number of crops in the set C ∪ G;
- N_f: total of plant families;
- $F(p)$: set of crops from plant family p ($p = 1, \cdots, N_f$);
- S_k: set of adjacent plots to plot k;
- t_i: cropping cycle of crop i, from seeding to harvest;
- I_i: period $[e_i, b_i]$ during which seeding crop i is recommended, where e_i is the earliest period, and b_i, is the latest one.

As a 0-1 linear optimization model, Equations (1.1) and (1.2) define the decision variables. Indexes are $i = 1, \cdots, n$; $j = 1 \in I_i$ and $k = 1, \cdots, L$, which represent that any growing crop could be planted on their proper seeding interval ($j \in I_i$) in any plot k from 1 to L. Index n represents a fallow period ($n = N + 1$).

$$x_{ijk} = \begin{cases} 1, & \text{if crop } i \text{ is planted during period } j \text{ in plot } k; \\ 0, & \text{otherwise.} \end{cases} \tag{1.1}$$

$$x_{ijk} \in \{0,1\}, \quad i = 1, \cdots, n, \quad j \in I_i, \quad k = 1, \cdots, L \tag{1.2}$$

Objective function in Equation (1.3) represents the total occupation of the plot set in the crop rotation planning. The objective function skips green manure crops in the total occupation, considering only crops from the C set. If $x_{ijk} = 1$, crop i will hold plot k from period j to $j + t_i$, leading to $t_i \cdot x_{ijk}$ filled periods in plot k.

$$\max \quad \sum_{k=1}^{L} \sum_{i \in C} \sum_{j \in I_i} t_i \cdot x_{ijk} \tag{1.3}$$

Constraint set in Equation (1.4) assures spatial and temporal limitation in the model, preventing two crops from occupying the same place at the same time. For instance, if $x_{ijk} = 1$, $x_{i(j-r)k} = 1$ when $r = 0$, and so, it will be feasible only if there is no other scheduled crop from period $j - (t_i - 1)$ to j.

$$\sum_{i=1}^{n} \sum_{r=0}^{t_i - 1} x_{i(j-r)k} \leq 1, \quad j = 1, \cdots, M, \quad k = 1, \cdots, L \tag{1.4}$$

Constraint set in Equation (1.5) avoids seeding crops from the same family in the same plot without disruption in the sequence. Using almost the same backtracking trick from Equation (1.4), which is looking back in the schedule t_i periods, if there is more than one scheduled crop from the same family, it is a unfeasible solution.

$$\sum_{i \in F(p)} \sum_{r=0}^{t_i} x_{i(j-r)k} \leq 1, \quad p = 1, \cdots, N_f, \quad j = 1, \cdots, M, k = 1, \cdots, L \tag{1.5}$$

Although the total occupation in the objective function does not rely in green manure crops, they must be in the crop rotation. Equation (1.6) establishes that one green manure crop should be in the solution, otherwise it is an unfeasible crop sequence. A fallow period is also required, according to Equation (1.7). Let record that index n represents a fallow period.

$$\sum_{j=1}^{M} \sum_{i \in G} x_{ijk} = 1, \quad k = 1, \cdots, L \tag{1.6}$$

$$\sum_{j=1}^{M} x_{njk} = 1, \quad k = 1, \cdots, L \tag{1.7}$$

Crops from the same family growing on the same period in adjacent plots are not allowed to satisfy the constraint set in Equation (1.8). Parameters u and k represent adjacent plots to plot k.

$$\sum_{i \in F_p} \sum_{r=0}^{t_i} [x_{i(j-r)u} + x_{i(j-r)v}] \leq 1, \quad p = 1, \cdots, N_f, \quad j = 1, \cdots, M, \quad (u, v) \in S_k \tag{1.8}$$

Crop demand, profit evaluation and nutrient requirements are not considered in the optimization model proposed by Santos et al. (2008). The design of the decision variables x_{ijk} and the constraint definition of plot's adjacency are very strong attributes from this approach.

1.1.2 Crop Rotation's Model B: Aliano

Aliano, Florentino e Pato (2014) developed a crop rotation model adapted from the concepts presented in Santos et al. (2008). Indexes in the model are $i = 1, \cdots, n$ where $n = N + 1$, $j = 1, \cdots, M$ (or considering only crop's seeding interval $j \in I_i$) and $k = 1, \cdots, L$. The list ahead describes the model parameters:

- M: planning horizon divided into M periods;
- L: cropping area with L plots;
- C: set of commercial crops;
- G: set of green manure crops;
- N: the total number of growing crops ($C \cup G$);
- N_f: number of plant families;
- $F(p)$: set of crops from the family p ($p = 1, \cdots, N_f$);
- S_k: set of adjacent plots to plot k;
- $area_k$: cultivable area of plot k ($area$);
- t_i: crop i cycle, including the time estimated for preparing the soil and harvesting;
- I_i: period $[e_i, b_i]$ during which it can seed crop i, where e_i is the earliest period, and b_i, is the latest recommended one;
- l_{ij}: profitability of crop i in period j per unit of area ($\$/area$);
- p_{ij}: production of crop i in period j ($unit/area$);

- D_i: crop i's demand (*yield unit*).

The binary linear model in Aliano, Florentino e Pato (2014) uses the decision variables in Equation (1.9), which their indexes are detailed in Equation (1.10).

$$x_{ijk} = \begin{cases} 1, & \text{if crop } i \text{ is planted during period } j \text{ in plot } k; \\ 0, & \text{otherwise.} \end{cases} \tag{1.9}$$

$$x_{ijk} \in \{0, 1\}, \quad i = 1, \cdots, n, \quad j \in I_i, \quad k = 1, \cdots, L \tag{1.10}$$

Although maximizing occupation (as shown in Equation (1.3)) could control weeds and reduce erosion over the crop rotation, considering profits on the crop sequence is fundamental. The objective function in Equation (1.11) defines profit maximization from the crop sequence. As farmers usually grow green manure crops for increasing soil properties, crops in the set G have no commercial value.

$$\max \quad \sum_{i \in C} \sum_{j \in I_i} \sum_{k=1}^{L} area_k \cdot l_{ij} \cdot x_{ijk} \tag{1.11}$$

Constraint set in Equation (1.12) avoids scheduling crops from the same family in adjacent plots. Parameter v represents plot that is one of plot k's neighbors.

$$\begin{gathered} \sum_{i \in F_p} \sum_{r=0}^{t_i - 1} \sum_{v \in S_k} x_{i(j-r)v} \leq L \left((1 - \sum_{i \in F_p} \sum_{r=0}^{t_i - 1} x_{i(j-r)k} \right), \\ p = 1, \cdots, N_f, \quad j = 1, \cdots, M, \quad k = 1, \cdots, L \end{gathered} \tag{1.12}$$

Constraint set in Equation (1.13) prevents crops from the same family from being consecutively seeded in any plot. Spatial and temporal limitation, which prevents crops from occupying the same plot simultaneously, are defined by Equation (1.14).

$$\sum_{i \in F_p} \sum_{r=0}^{t_i} x_{i(j-r)k} \leq 1, \quad p = 1, \cdots, N_f, \quad j = 1, \cdots, M, \quad k = 1, \cdots, L \tag{1.13}$$

$$\sum_{i=1}^{N+1} \sum_{r=0}^{t_i - 1} x_{i(j-r)k} \leq 1, \quad j = 1, \cdots, M, \quad k = 1, \cdots, L \tag{1.14}$$

Constraints related to the allocation of green manure crop and fallow intervals are presented in Equations (1.16) and (1.17). They are not limited to only one allocation as in the model proposed by Santos et al. (2008), which has similar constraints (Equations (1.6) and (1.7)).

$$\sum_{i \in G} \sum_{j=1}^{M} x_{ijk} \geq 1, \quad k = 1, \cdots, L \tag{1.15}$$

$$\sum_{j=1}^{M} x_{njk} \geq 1, \quad k = 1, \cdots, L \tag{1.16}$$

The model proposed by Aliano, Florentino e Pato (2014) also includes demand constraint. Yields must be equal or greater than crop demand (D_i). Constraint set in Equation (1.17) imposes demand parameters.

$$\sum_{j \in I_i^D} \sum_{k=1}^{L} area_k \cdot p_{ij} \cdot x_{ijk} \geq D_i, \quad i \in C \tag{1.17}$$

1.1.3 Crop Rotation's Model C: Forrester

Forrester e Rodriguez (2018) proposed a mixed-integer programming model of CRP. They focused on an organic farm and developed a mixed-integer program which provides a four-year crop rotation schedule. They have addressed market demand, weed control, soil nutrient depletion, and the type of irrigation utilized. Rather than maximize profit, they focus on reaching the projected market demand without violating any rotation's principle included in the model. There are four essential indexes in the model: (1) $i \in Crops$; (2) $j \in Fields$; (3) $k \in Years$; and (4) $l \in Months$. The main parameters are:

- ABF_j: available bed feet in Field j;
- DEM_{ikl}: demand of Crop i in Year k and Month l.

Some variables are described ahead:

- $x_{ijkl} = \begin{cases} 1, & \text{if Crop } i \text{ is planted in Plot } j \text{ in Year } k \text{ during Month } l; \\ 0, & \text{otherwise.} \end{cases}$
- y_{ijkl}: proportion of Field j to Crop i in Year k during Month l;
- u_{ikl}: unmet demand of Crop i in Year k during Month l.

The objective function presents in Equation (1.18). By minimizing the unmet demand (u_{ikl}), they try to reach the projected market demand.

$$\min \sum_{i \in Crops} \sum_{k \in Years} \sum_{l \in Months} u_{ikl} \tag{1.18}$$

In the model proposed by Forrester e Rodriguez (2018), associated constraints in Equation (1.19) state that $y_{ijkl} \geq 0$ only if $x_{ijkl} = 1$. Constraints in Equation (1.20) ensure a crop group can only be seeded if it uses at least 10% of the field. Constraints in Equations (1.21) and (1.22) are related to the projected market demand.

$$y_{ijkl} \leq x_{ijkl} \tag{1.19}$$

$$y_{ijkl} \geq 0.10 \ . \ x_{ijkl} \tag{1.20}$$

$$\sum_{j \in Fields} ABF_j \ . \ y_{ijkl} \geq DEM_{ikl} - u_{ikl} \tag{1.21}$$

$$u_{ikl} \geq 0 \tag{1.22}$$

Forrester e Rodriguez (2018) also considered soil nutrient depletion, weed control and the type of irrigation utilized.

2 A NEW MONO-OBJECTIVE MODEL OF THE CRP

Crop nutrient demand and soil fertility are fundamental in the crop rotation schedule. By developing a new mono-objective model of the CRP, we would like to strength the relationship between the nutrient flow on cultivable fields and the crop sequence. This chapter introduces the proposed mono-objective model and its parameters.

2.1 The proposed mono-objective model of the CRP

According to Deep et al. (2009), a mixed-integer linear programming (MILP) problem is an optimization problem in which some or all decision variables are restricted to integer values. They are linear or nonlinear and subject to a constraint set or not subject to. We would like to detail our proposed MILP model of the CRP in this section. The main indexes in the proposed optimization model are:

- i: crop index ($i = 1, \cdots, N$);
- j: period index ($j = 1, \cdots, M$);
- k: plot index ($k = 1, \cdots, L$);
- p: crop family index ($p = 1, \cdots, N_f$);
- α: fertilization interval index, $\alpha \in \Omega$, $\Omega = \{\alpha \in \mathbf{N}^* \mid \alpha \cdot \theta \leq M, \quad \theta \in \mathbf{N}^*\}$.

Fertilization interval (θ) and planning horizon (M), which are model parameters, define the fertilization index (α). For instance, if the fertilization interval is 12 periods ($\theta = 12$) and the planning horizon equals 24 periods ($M = 24$), then the Ω set is $\{1, 2\}$ due to $\alpha = 1$ and $\alpha = 2$ satisfy the definition $\alpha \cdot \theta \leq M$.

Fundamental parameters are as follows:

- N: the total number of crops;
- N_f: number of crop families;
- M: planning horizon divided in M periods (*time unit*);
- L: cultivable field divided in L plots;

- F_p: the set of crops from the Family p;
- $area_k$: cultivable area of Plot k (*area unit*);
- l_i: profitability of Crop i (\$);
- t_i: Crop i production cycle, from seed to the harvest (*time unit*);
- p_i: Crop i typical yield (*Crop i yield*);
- I_i: the set of Crop i seeding periods $\{I_{i1}, \cdots, I_{in}\}$;
- D_i: demand for Crop i (*Crop i yield in M periods*);
- S_k: the set of adjacent plots of the Plot k;
- R_{N_i}: the projected nitrogen removal of Crop i (*kg N / area unit*);
- R_{P_i}: the projected phosphorus removal of Crop i (*kg P / area unit*);
- R_{K_i}: the projected potassium removal of Crop i (*kg K / area unit*);
- c_N: typical nitrogen fertilizer cost (\$);
- c_P: typical phosphorus fertilizer cost (\$);
- c_K: typical potassium fertilizer cost (\$);
- β: sequence safety interval *time unit*;
- θ: fertilization interval (*time unit*);
- F_{min}: minimum fertilizer supply in θ periods;
- F_{max}: maximum fertilizer supply in θ periods.

The proposed model is formulated as a MILP problem, there are binary and continuous variables. Profits (P) represent the objective function output. Fertilization amendments ($F_{N_{\alpha k}}$, $F_{P_{\alpha k}}$ and $F_{K_{\alpha k}}$) are continuous variables. The crop sequence variables x_{ijk} are binary and they keep track of when and where crops are seeded. Variables in the proposed model are as follows:

- P: net profits from the crop scheme;
- x_{ijk}: crop allocation variables, where:

$$x_{ijk} = \begin{cases} 1, & \text{if Crop } i \text{ is seeded in the Period } j \text{ in Plot } k. \\ 0, & \text{otherwise.} \end{cases}$$

- $F_{N_{\alpha k}}$: nitrogen fertilizer amount required in Interval α in Plot k;
- $F_{P_{\alpha k}}$: phosphorus fertilizer amount required in Interval α in Plot k;
- $F_{K_{\alpha k}}$: potassium fertilizer amount required in Interval α in Plot k.

Our objective function and associated constraints are as follows:

$$\max \quad P = \sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{k=1}^{L} area_k \,.\, l_i \,.\, p_i \,.\, x_{ijk} - \sum_{\alpha\in\Omega}\sum_{k=1}^{L} F_{N_{\alpha k}} \,.\, c_N + F_{P_{\alpha k}} \,.\, c_P + F_{K_{\alpha k}} \,.\, c_K \tag{2.1}$$

$$\text{subject to} \quad \sum_{i\in F_p}\sum_{r=0}^{t_i-1}\sum_{v\in S_k} x_{i(j-r)v} \leq L\left(1-\sum_{i\in F_p}\sum_{r=0}^{t_i-1} x_{i(j-r)k}\right), \quad p=1,\cdots,Nf, \quad j=1,\cdots,M, \quad k=1,\cdots,L \tag{2.2}$$

$$\sum_{i\in F_p}\sum_{r=0}^{t_i+\beta} x_{i(j-r)k} \leq 1, \quad p=1,\cdots,Nf, \quad j=1,\cdots,M, \quad k=1,\cdots,L \tag{2.3}$$

$$\sum_{i=1}^{N}\sum_{r=0}^{t_i-1} x_{i(j-r)k} \leq 1, \quad j=1,\cdots,M, \quad k=1,\cdots,L \tag{2.4}$$

$$F_{N_{\alpha k}} - \sum_{i=1}^{N}\sum_{j=1+(\alpha-1)\,.\,\theta}^{\alpha\,.\,\theta} x_{ijk} \,.\, area_k \,.\, R_{N_i} \geq 0, \quad k=1,\cdots,L, \quad \alpha\in\Omega \tag{2.5}$$

$$F_{P_{\alpha k}} - \sum_{i=1}^{N}\sum_{j=1+(\alpha-1)\,.\,\theta}^{\alpha\,.\,\theta} x_{ijk} \,.\, area_k \,.\, R_{P_i} \geq 0, \quad k=1,\cdots,L, \quad \alpha\in\Omega \tag{2.6}$$

$$F_{K_{\alpha k}} - \sum_{i=1}^{N}\sum_{j=1+(\alpha-1)\,.\,\theta}^{\alpha\,.\,\theta} x_{ijk} \,.\, area_k \,.\, R_{K_i} \geq 0, \quad k=1,\cdots,L, \quad \alpha\in\Omega \tag{2.7}$$

$$\sum_{j=1}^{M}\sum_{k=1}^{L} area_k \,.\, p_{ij} \,.\, x_{ijk} \geq D_i, \quad i=1,\cdots,N \tag{2.8}$$

$$\sum_{k=1}^{L}\sum_{j\notin I_i} x_{ijk} \leq 0, \quad i=1,\cdots,N \tag{2.9}$$

$$x_{ijk}\in\{0,1\}, \quad i=1,\cdots,N, \quad j=1,\cdots,M, \quad k=1,\cdots,L \tag{2.10}$$

$$F_{N_{\alpha k}} = \{F_{N_{\alpha k}}\in\mathbf{R}^+ \mid F_{max}\geq F_{N_{\alpha k}}\geq F_{min}\}, \quad \alpha\in\Omega, \quad k=1,\cdots,L \tag{2.11}$$

$$F_{P_{\alpha k}} = \{F_{P_{\alpha k}}\in\mathbf{R}^+ \mid F_{max}\geq F_{P_{\alpha k}}\geq F_{min}\}, \quad \alpha\in\Omega, \quad k=1,\cdots,L \tag{2.12}$$

$$F_{K_{\alpha k}} = \{F_{K_{\alpha k}}\in\mathbf{R}^+ \mid F_{max}\geq F_{K_{\alpha k}}\geq F_{min}\}, \quad \alpha\in\Omega, \quad k=1,\cdots,L \tag{2.13}$$

$$\Omega = \{\alpha\in\mathbf{N}^* \mid \alpha \,.\, \theta \leq M, \quad \theta\in\mathbf{N}^*\} \tag{2.14}$$

The objective function in Equation (2.1) is based in Aliano, Florentino e Pato (2014). It represents the profit maximization in the crop rotation during the planning horizon. Fertilization costs are also included, looking forward to reduce the dependency on external chemical fertilizers.

Constraints in Equations (2.2), (2.3) and (2.4) are adapted from Santos et al. (2011). They hold the main requirements in the crop rotation scheme. Constraint set in Equation (2.2) ensures that crops from the same family will not occupy adjacent plots simultaneously. Constraint set in Equation (2.3) prevents scheduling crops from the same family in the same area without rotating crops or assigning a fallow interval. The sequence safety interval β increases the distance . The third one represents the spatial limitation of the plots, which assures there will not be more than one scheduled crop in a plot at a specific period.

The constraint set in Equation (2.8) aims to meet market demand from each crop. Constraints in Equation (2.9) prevent any scheduling outside the proper seed time for each crop.

The nutrient budget concepts presented in Section 2.2 led to the constraint set in Equations (2.5), (2.6) and (2.7). They state that the nutrient amendments for each plot have to reach at least the crop requirements.

2.1.1 A sample solution: understanding the proposed model of the CRP

Although a mathematical model summarizes the CRP attributes, its construction bases on practical details that might be unnoticed in the model presentation. Hence, we selected the sample solution presented in Table 1 and entrusted the current section with the task of lightning up the features of the model. This sample solution was generated with the optimization solver IBM ILOG CPLEX, more details about the software are presented in Appendix D.

There are some required parameters from the fields (plots) that are presented in Figure 3. Each plot has its own cultivable area, and they can be adjacent to each other, which means they share a common border. Plot drawings are just illustrative; the adjacent concept does not regard any particular shape. By analyzing the cultivable area from each plot in Figure 3, we should expect the highest profitable crops allocated in the large areas, and so, generating the maximum income. As the plots are adjacent to each other, the largest area should match a very profitable crop.

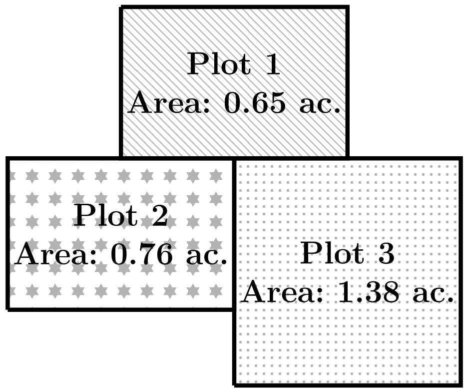

Figure 3: Describing a plot set and its features.

The periods in Table 1 are 15-day intervals, and so, the sample solution is a one-year-long sequence. There is no overlapping in the sample solution, which means the constraint set in Equation (2.4) is fulfilled. Cells filled with six-pointed stars represent fallow periods in the solution. Plant families are also displayed, so we could check whether crops from the same family are simultaneous scheduled in the neighborhood, which is limited by Constraints in Equation (2.2). The constraint set in Equation (2.3) is supposed to restrict subsequent schedules, but it does not block from scheduling crops in the same family after rotating the area or fallow periods, as we can see in the sample solution.

Table 1: A sample solution for the proposed model.

<table>
<tr><th>Period</th><th>Plot 1</th><th>Plot 2</th><th>Plot 3</th></tr>
<tr><td>1</td><td rowspan="4">Leaf Lettuce for Fresh Market, Winter (Lettuce)</td><td></td><td rowspan="4">Cabbage for Fresh Market, Spring (Mustard)</td></tr>
<tr><td>2</td><td rowspan="8">Tomato for Fresh Market, Spring (Nightshade)</td></tr>
<tr><td>3</td></tr>
<tr><td>4</td></tr>
<tr><td>5</td><td></td><td></td></tr>
<tr><td>6</td><td rowspan="6">Spinach for Fresh market, Summer (Beet)</td><td></td></tr>
<tr><td>7</td><td rowspan="4">Leaf Lettuce for Fresh Market, Summer (Lettuce)</td></tr>
<tr><td>8</td></tr>
<tr><td>9</td></tr>
<tr><td>10</td><td rowspan="4">Cabbage for Fresh Market, Summer (Mustard)</td></tr>
<tr><td>11</td><td rowspan="4">Tomato for Fresh Market, Summer (Nightshade)</td></tr>
<tr><td>12</td><td></td></tr>
<tr><td>13</td><td rowspan="4">Summer Squash for Fresh Market (Cucurbit)</td></tr>
<tr><td>14</td><td rowspan="4">Leaf Lettuce for Fresh Market, Summer (Lettuce)</td></tr>
<tr><td>15</td><td rowspan="4">Spinach for Fresh market, Fall (Beet)</td></tr>
<tr><td>16</td></tr>
<tr><td>17</td><td></td></tr>
<tr><td>18</td><td></td><td rowspan="2"></td></tr>
<tr><td>19</td><td rowspan="4">Leaf Lettuce for Fresh Market, Winter (Lettuce)</td><td rowspan="4">Cabbage for Fresh Market, Fall (Mustard)</td></tr>
<tr><td>20</td><td rowspan="4">Spinach for Fresh market, Winter (Beet)</td></tr>
<tr><td>21</td></tr>
<tr><td>22</td></tr>
<tr><td>23</td><td rowspan="2">Broccoli for Fresh Market, Winter (Mustard)</td><td></td></tr>
<tr><td>24</td><td>Bell Peppers for Fresh Market and Processing, Spring (Nightshade)</td><td>Leaf Lettuce for Fresh Market, Winter (Lettuce)</td></tr>
</table>

From the sample solution in Table 1, we have selected Plot 1's rotation sequence and evaluated the projected profits in this plot. Table 2 exhibits the expected returns from the partial sequence. In order to evaluate the full objective function in Equation (2.1), we have to account fertilization costs that are detailed in Table 3.

Table 2: Evaluating profits from the proposed crop sequence.

Period	Plot 1 (Area: 0.65 ac.)	Average Yield	Profit per Unit	Profit per Crop • Area
1	Leaf Lettuce for Fresh Market, Winter (Lettuce)	569 cartons / ac.	US$ 5.71 / carton	US$ 2,112.58
2				
3				
4				
5				
6	Spinach for Fresh market, Summer (Beet)	8.53 T / ac.	US$ 227.77 / T	US$ 1,262.89
7				
8				
9				
10				
11				
12				
13	Summer Squash for Fresh Market (Cucurbit)	176 cwt / ac.	US$ 9.79 / cwt	US$ 1,120.66
14				
15				
16				
17				
18				
19	Leaf Lettuce for Fresh Market, Winter (Lettuce)	569 cartons / ac.	US$ 5.71 / carton	US$ 2,112.58
20				
21				
22				
23	Broccoli for Fresh Market, Winter (Mustard)	6.1 T / ac.	US$ 478.20 / T	US$ 1,896.13
24				
Total				US$ 8,504.80

Many researches about the CRP consider fertilizer costs only as budget parameters, but the proposed model places in evidence the required fertilizer amounts during the crop rotation planning horizon, which could be essential to make sustainable decisions right in the planning. The nutrient removal from the partial sequence presents in Table 3.

Table 3: Nutrient removal according to the crop sequence.

Period	Plot 1 (Area: 0.65 ac.)	N removal (kg)	P removal (kg)	K removal (kg)
1	Leaf Lettuce for Fresh Market, Winter (Lettuce)	17.69	8.49	35.38
2				
3				
4				
5				
6	Spinach for Fresh market, Summer (Beet)	11.8	2.35	11.8
7				
8				
9				
10				
11				
12				
13	Summer Squash for Fresh Market (Cucurbit)	9.02	2.00	-
14				
15				
16				
17				
18				
19	Leaf Lettuce for Fresh Market, Winter (Lettuce)	17.69	8.49	35.38
20				
21				
22				
23	Broccoli for Fresh Market, Winter (Mustard)	8.25	2.35	11.8
24				
Total		64.45	23.68	94.36
Costs		US$ 105.55	US$ 38.36	US$ 113.23

2.1.2 Computational tests and model validation

We have performance several computational test in order to validate the proposed optimization model. We have also developed a genetic algorithm to produce crop sequences using the proposed model. Chapter 4 presents the genetic algorithm in detail. Our results and final observations presents in Chapter 6.

2.2 Nutrient budget and farm management

We have modeled fertilization parameters in Equations (2.10), (2.11) and (2.12). Evaluating fertilization costs develops in the objective function from Equation (2.1). Using a partial sequence from Table 1, we have demonstrated the fertilization cost evaluation in Table 3. Nutrient budget is a long topic, which we discuss some fundamental elements over this section.

Nutrient budget aims to reach the complexity of the nutritional cycle in agriculture by acknowledging the nutrient flow as nutrients enter the soil system or leave it (MEISINGER; CALDERON; JENKINSON, 2008). From a mathematical perception, nutrient budget and cash flow are quite similar, both represent the balance between inputs and outputs, but only the first one relays most on empirical methodologies and in a wide variables set that might not generate a very precise model due to complex scenario. In most cases, a fair estimation is enough for managing crops, while the second one based on precisely countable information and the current state can be accessed anytime. Nutrient budgets have some tolerance, but a cash flow will be complete useless if it lacks accuracy.

In other to provide a better dimension of the nutrient cycle complexity related to the agricultural practices, let us partially analyze the nitrogen cycle. Nitrification could happen whether ammonia was added to the soil system through applications of synthetic nitrogen fertilizer. This increases the nitrous oxide production by providing large amounts of soil substrate for microbial nitrogen conversion (nitrification and denitrification), or other fertilizer forms, such as nitrogen fixation by legumes and mineralized soil organic matter. Microbial activity results in the transformation of ammonia to nitrite and nitrate, although nitrous oxide losses are still significant (BEHNKE et al., 2018).

Saturating rains might drive the soil waterlogged, which depletes the oxygen supply to the root system and inhibits respiration, reducing the energy status of cells related to the crop metabolic process. Low oxygen level leads to a series of metabolic changes for crops, and anaerobic metabolism is activated, decreasing energy production dramatically since glycolysis (anaerobic metabolic pathway) is the primary source supply. Under these stress conditions, the plant's survival relays most on the anaerobic activities. Low oxygen also speeds up denitrification. Denitrifiers use nitrate as a terminal electron acceptor; the complete denitrification from nitrate to nitrogen releases nitrous oxide in an intermediate step. As a result, denitrification might release a large proportion of annual nitrous oxide during relatively short periods (SIGNOR; CERRI, 2013).

In general, fertilized crops consume less than 50% of the total nitrogen applied, exceeding applied nitrogen is mostly wasted and subject to leaching, volatilization, and denitrification. A large amount of land reserved for growing highly fertilized corn and nitrogen-fixing soybeans supplies the substrate need to emit significant quantities of nitrous oxide. Nitrous oxide emissions resulting from conventional agriculture contribute mainly to the greenhouse effect (SAHA et al., 2018).

Synchronizing fertilizer applications with crop requirements decreases the surplus nitrogen's availability during long periods, and then emissions of nitrous oxide from agricultural practices might be minimized. Besides volatilization and leaching, fertilizer applications on the soil surface are subject to run-off and surface loss, which could be reduced using covered fertilizer and applications near the root system (RÜTTING; ARONSSON; DELIN, 2018).

Summarizing the nitrogen cycle from the agricultural interests, we have nitrification, denitrification, volatilization, leaching, run-off, and crop uptake. These processes are time-dependent and relay on microbial agents. Weather conditions also have a significant influence on the nitrogen flow.

A nutrient budget measures nutrients imported to and exported from a defined system, and it represents the accounting of removed nutrients with the harvested crop and nutrient inputs from natural sources or inorganic fertilizer applications (BASSANINO et al., 2011). There are three conventional nutrient budget techniques: farm-gate, soil surface, and soil system. Agronomic ecosystems have distinct characteristics and complexity, which reflect on the accuracy of the nutrient budget technique (WATSON; ATKINSON, 1999) (SHOBER; HOCHMUTH; WIESE, 2011). The weather can also affect the nutrient cycle and increase losses during specific periods. Soil nutrient surplus might be considered an index of sustainability in the crop field management (SHOBER; HOCHMUTH; WIESE, 2011) (OENEMA; KROS; VRIES, 2003).

Establishing the required amount of nutrients for each crop involves plenty of data, which should be precisely acquired in the crop field. The appealing for precision agriculture regards this concern, without regular assessment of the nutrient availability and crop requirements, any detached data relating nutrient demand seems unreasonable. Once the availability of nutrients in the soil and the proper crop requirements have been settled, establishing a soil surface nutrient budget sounds entirely appropriate. A soil surface nutrient budget accounts for all nutrients that enter the soil surface and leave the soil through crop uptake. The total amount of manure or nitrogen amendment applied would be adjusted to account for ammonia volatilizing since this amount would not enter accumulated in the soil surface and biological nitrogen fixation. The nutrient surplus or deficit presents the total nutrient loss from the soil or soil storage variations (BASSANINO et al., 2011) (OENEMA; KROS; VRIES, 2003) (SAINJU, 2017).

3 A NOVEL MULTIOBJECTIVE APPROACH FOR THE CRP

Crop rotation can be seen as an agrarian practice that could control pests and their damage (SANTOS et al., 2011), or a management tool, which improves agribusiness performance while establishing sustainable thoughts in the farm (MOHLER; JOHNSON, 2009). Decisions about the crop sequence consider many aspects, such as market opportunities, soil characteristics, and farm resources (labor and machinery). Considering the long-term consequences of well-established crop rotations and their relationship with soil properties, we proposed a new multiobjective model of the CRP. Section 3.1 describes essential characteristics of the novel formulation. Section 3.2 presents the proposed multiobjective model. Section 3.3 details the Crop Rotation Quality Ratiing (CRQR) using a sample solution of the proposed multiobjective model.

3.1 Crop cultural traits and the quality of the crop rotation

Crop schemes interchanging crops from grass and legume families gradually increase soil fertility and tillage. Legume crops have outstanding performance as green manure, capturing atmospheric nitrogen and making it available for further plants. Deep taproots in some of these crops can recycle leached nutrients in the subsoil to the upper layer. Grass crops have a fine root system, which can unleash nutrients and aggregate them into soil crumbs (MOHLER; JOHNSON, 2009) (LOSS et al., 2015).

Tilth usually describes the soil condition for proper seed germination and root growth. Soil tilth in the good state might benefit crops from improving water infiltration and aeration. Crop rotation can improve soil tilth and reduce the requirements of mechanical interventions in the crop field. Soil aggregates and the space between their structure (called pores) characterize the tilth state. On rainy and wet periods, large pores prevent oxygen deficiency, and the small ones increase water storage in the soil, which is essential during dry seasons (MOHLER; JOHNSON, 2009).

Cover cropping has long been related to soil improvements, such as physical properties (providing ground cover and slowing down soil erosion), nitrogen availability (fixing atmospheric nitrogen into the soil), and increasing organic matter content. In general, soil quality improvements are attained through cover crop decomposition, which enriches organic matter content in the surface layer. Finally, biomass mineralization releases nutrients into absorbed ways, increasing soil fertility and reducing the consumption of amendments in the following crop (EDWARDS; BURNEY, 2005).

In the proposed model, evaluating the potential benefits of the crop scheme defines rotation quality. Based on Clark (2012), the following parameter set characterize the role of cover crops and their benefits in the soil system. Combining these parameters and the cover crop allocation in the plot set results in the Crop Rotation Quality Rating (CRQR).

- Total Nitrogen Fixation (fixed by legumes);
- Dry Matter;
- Soil Builder;
- Erosion Fighter;
- Weed Fighter;
- Good Grazing;
- Quick Growth;
- Lasting Residue;
- Duration;
- Harvest Value;
- Cash Crop Inter-seed.

Some non-trivial parameters are: (1) *soil builder* which rates the capacity of producing organic matter and improving tillage; (2) *erosion fighter* that ranks how extensive and how quickly a root system develops, it also accounts performance of sustaining the soil against erosion; (3) *weed fighter* represents how well crops out-competes weeds during their life cycle; (4) *food grazing* indicates the potential value as a forage; and (5) *lasting residue* that ranks the effectiveness of the cover crop in providing a long-lasting mulch (CLARK, 2012).

3.2 The proposed multiobjective model of the CRP

The proposed CRP model embraces crop rotation quality, cover crop traits, beyond the nutrient budget already developed in Chapter 2. The main indexes in the proposed model are as follows:

- i: crop index ($i = 1, \cdots, N$);
- j: period index ($j = 1, \cdots, M$);
- k: plot index ($k = 1, \cdots, L$);
- p: crop family index ($p = 1, \cdots, N_f$);
- α: fertilization interval index, $\alpha \in \Omega$, $\Omega = \{\alpha \in \mathbf{N}^* \mid \alpha \cdot \theta \leq M, \quad \theta \in \mathbf{N}^*\}$;
- b: crop cultural trait index ($b = 1, \cdots, C_T$), $C_T = 11$ as we selected 11 cultural traits.

Fundamental parameters presents along the list ahead:

- N: the total number of crops, including cover crops;
- C: the set of cover crops;
- N_f: number of crop families;
- M: planning horizon divided in M periods (*time unit*);
- L: cultivable field divided in L plots;
- F_p: the set of crops from the Family p;
- $area_k$: cultivable area of Plot k (*area unit*);
- l_i: profitability of Crop i (\$);
- t_i: Crop i production cycle, from seed to the harvest (*time unit*);
- p_i: Crop i typical yield (*Crop i yield*);
- I_i: the set of Crop i seeding periods $\{I_{i1}, \cdots, I_{in}\}$;
- D_i: demand for Crop i (*Crop i yield in M periods*);
- S_k: the set of adjacent plots of the Plot k;
- R_{N_i}: the projected nitrogen removal of Crop i (*kg N / area unit*);
- R_{P_i}: the projected phosphorus removal of Crop i (*kg P / area unit*);
- R_{K_i}: the projected potassium removal of Crop i (*kg K / area unit*);
- c_N: typical nitrogen fertilizer cost (\$);
- c_P: typical phosphorus fertilizer cost (\$);
- c_K: typical potassium fertilizer cost (\$);
- β: sequence safety interval *time unit*;
- θ: fertilization interval (*time unit*);
- F_{min}: minimum fertilizer supply in θ periods;
- F_{max}: maximum fertilizer supply in θ periods;
- T_{bi}: rated crop cultural Trait b of Crop i.

Decision variables have the same structure from the proposed mono-objective model in Chapter 2. Crop allocation is defined by $x_{ijk} = 1$, if the Crop i is seeded in Period j in Plot k or $x_{ijk} = 0$, otherwise. Fertilization decision variables $F_{N_{\alpha k}}$, $F_{P_{\alpha k}}$ and $F_{K_{\alpha k}}$ are restricted to the interval $[F_{min}, F_{max}]$. Fertilization interval index α, which is defined by Ω, establishes how many continuous variables $F_{N,P,K}$ are required. The variable set in the proposed multiobjective model presents ahead:

- P: net profits from the crop scheme;
- x_{ijk}: crop allocation variables, where:

$$x_{ijk} = \begin{cases} 1, & \text{if Crop } i \text{ is seeded in the Period } j \text{ in Plot } k. \\ 0, & \text{otherwise.} \end{cases}$$

- $F_{N_{\alpha k}}$: nitrogen fertilizer amount required in Interval α in Plot k;
- $F_{P_{\alpha k}}$: phosphorus fertilizer amount required in Interval α in Plot k;
- $F_{K_{\alpha k}}$: potassium fertilizer amount required in Interval α in Plot k;
- C_{RQR}: Crop Rotation Quality Rating of the crop rotation sequence;
- Q_k: rotation quality rating of plot k;
- Q_{avg}: average plot rotation quality rating.

Constraints in Equation (3.3) prevent the allocation of crops from the same family on adjacent plots at the same period. Constraints in Equation (3.4) avoid simultaneous allocations of crops from the same family without rotation or a proper fallow period. Equation (3.5) represents the spatial limitation in the plot set. Nutrient budgets, which are discussed in Section 2.2, are presented in Equations (3.6), (3.7) and (3.8). Equation (3.9) defines crop demand in the optimization problem and Equation (3.10) restricts allocation in the proper seeding time of each crop. Equations from (3.11) to (3.15) define the decision variables in the model.

Objective function in Equation (3.1) evaluates profitability from the crop scheme. Incomes are based on production, returns per unit and plot area; while it determines fertilization costs individually considering the variables $F_{N,P,K_{\alpha k}}$ and costs per unit $c_{N,P,K}$.

The second objective function in Equation (3.2) is the CRQR, developed from the crop cultural traits presented in Subsection 3.1. Getting the rotation quality rate from each plot k is the first step. Q_k is the sum of each rated parameter T_{bi} from each cover crop ($i \in C$) allocated in plot k. Q_{avg} represents the average value of Q_k. CRQR is the sum of the rotation quality rating from each plot (Q_k) minus the deviation of Q_k. The sum

$\sum_{k=1}^{L}|(Q_k - Q_{avg})|$ ensures that cover crops will be fairly distributed in the crop sequence, avoiding too much deviation from Q_k.

$$\max \quad P = \sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{k=1}^{L} area_k \;.\; l_i \;.\; p_i \;.\; x_{ijk} - \sum_{\alpha\in\Omega}\sum_{k=1}^{L} F_{N_{\alpha k}} \;.\; c_N + F_{P_{\alpha k}} \;.\; c_P + F_{K_{\alpha k}} \;.\; c_K \tag{3.1}$$

$$\max \quad C_{RQR} = \sum_{k=1}^{L} Q_k - \sum_{k=1}^{L}|(Q_k - Q_{avg})|, \tag{3.2}$$

$$Q_k = \sum_{j=1}^{M}\sum_{i\in C}\sum_{b=1}^{C_T} T_{bi} \;.\; x_{ijk}, \quad k = 1,\cdots,L$$

subject to

$$\sum_{i\in F_p}\sum_{r=0}^{t_i-1}\sum_{v\in S_k} x_{i(j-r)v} \leq L\left(1 - \sum_{i\in F_p}\sum_{r=0}^{t_i-1} x_{i(j-r)k}\right), \quad p = 1,\cdots,Nf, \quad j = 1,\cdots,M, \quad k = 1,\cdots,L \tag{3.3}$$

$$\sum_{i\in F_p}\sum_{r=0}^{t_i+\beta} x_{i(j-r)k} \leq 1, \quad p = 1,\cdots,Nf, \quad j = 1,\cdots,M, \quad k = 1,\cdots,L \tag{3.4}$$

$$\sum_{i=1}^{N}\sum_{r=0}^{t_i-1} x_{i(j-r)k} \leq 1, \quad j = 1,\cdots,M, \quad k = 1,\cdots,L \tag{3.5}$$

$$F_{N_{\alpha k}} - \sum_{i=1}^{N}\sum_{j=1+(\alpha-1)\,.\,\theta}^{\alpha\,.\,\theta} x_{ijk} \;.\; area_k \;.\; R_{N_i} \geq 0, \quad k = 1,\cdots,L, \quad \alpha\in\Omega \tag{3.6}$$

$$F_{P_{\alpha k}} - \sum_{i=1}^{N}\sum_{j=1+(\alpha-1)\,.\,\theta}^{\alpha\,.\,\theta} x_{ijk} \;.\; area_k \;.\; R_{P_i} \geq 0, \quad k = 1,\cdots,L, \quad \alpha\in\Omega \tag{3.7}$$

$$F_{K_{\alpha k}} - \sum_{i=1}^{N}\sum_{j=1+(\alpha-1)\,.\,\theta}^{\alpha\,.\,\theta} x_{ijk} \;.\; area_k \;.\; R_{K_i} \geq 0, \quad k = 1,\cdots,L, \quad \alpha\in\Omega \tag{3.8}$$

$$\sum_{j=1}^{M}\sum_{k=1}^{L} area_k \;.\; p_{ij} \;.\; x_{ijk} \geq D_i, \quad i = 1,\cdots,N \tag{3.9}$$

$$\sum_{k=1}^{L}\sum_{j\notin I_i} x_{ijk} \leq 0, \quad i = 1,\cdots,N \tag{3.10}$$

$$x_{ijk} \in \{0,1\}, \quad i = 1,\cdots,N, \quad j = 1,\cdots,M, \quad k = 1,\cdots,L \tag{3.11}$$

$$F_{N_{\alpha k}} = \{F_{N_{\alpha k}} \in \mathbf{R}^+ \mid F_{max} \geq F_{N_{\alpha k}} \geq F_{min}\}, \quad \alpha\in\Omega, \quad k = 1,\cdots,L \tag{3.12}$$

$$F_{P_{\alpha k}} = \{F_{P_{\alpha k}} \in \mathbf{R}^+ \mid F_{max} \geq F_{P_{\alpha k}} \geq F_{min}\}, \quad \alpha\in\Omega, \quad k = 1,\cdots,L \tag{3.13}$$

$$F_{K_{\alpha k}} = \{F_{K_{\alpha k}} \in \mathbf{R}^+ \mid F_{max} \geq F_{K_{\alpha k}} \geq F_{min}\}, \quad \alpha\in\Omega, \quad k = 1,\cdots,L \tag{3.14}$$

$$\Omega = \{\alpha \in \mathbf{N}^* \mid \alpha \;.\; \theta \leq M, \quad \theta \in \mathbf{N}^*\} \tag{3.15}$$

3.3 Sample solution of the proposed multiobjective model of the CRP

Using IBM ILOG CPLEX, which supports Optimization Programming Language (OPL), we generated the crop scheme shown in Table 6. Plot set characteristics exhibits in Figure 3. Our main purpose in this section is to demonstrate how to evaluate objective function in Equation (3.2). Before proceeding to the solution, we would like to introduce a rating of cover crops attributes. Each column from Tables 4 and 5 represents one ranked feature b. The best performance has the highest classification (upper bound is 4 and lower bound is 0). Restricting CRQR evaluation to cover crops leaves cash crops outside the index composition. Cover crops are designed to improve cultivable fields, while cash crops are profit oriented. Cultural practices developed in cover crop would not compromise soil fertility; hereby, cover crops would only enhance soil properties and are more appropriate in the rotation quality analysis.

Table 4: Rating the qualities of the cover crops in terms of soil benefits.

Cover Crops	Family	Dry matter	N Scav-enger	Soil Builder	Erosion Fighter	Weed Fighter
Barley	Grass	2.7	3	3	4	4
Oats	Grass	2.7	3	2	3	3
Rye	Grass	2.9	4	4	4	4
Buckwheat	Buckwheat	1.3	0	2	1	1
Sorghum-sudangrass	Grass	4.0	4	4	4	4
Mustard	Mustard	2.7	2	3	3	3
Field peas	Legume	2.0	1	2	3	3

Table 5: Rating the qualities of cover crops in terms of soil benefits.

Cover crops	Good Grazing	Quick Growth	Lasting Residue	Duration	Harvest - Forage	Harvest - Seed	Cash Crop Inter-seed
Barley	3	3	4	2	3	2	3
Oats	2	4	2	1	2	2	4
Rye	2	4	4	3	1	1	3
Buckwheat	3	4	0	1	0	1	3
Sorghum-sudangrass	2	4	3	4	4	0	0
Mustard	2	3	1	2	0	1	0
Field peas	2	3	1	2	4	3	4

Table 6 presents a sample solution of the proposed CRP multiobjective model. Crop families from corresponding crops are exhibited in parenthesis. Skipping cash crops leaves tomato, spinach, bell peppers, oats and leaf lettuce outside the CRQR evaluation (unfilled cells represent cash crops). Cover crops in the solution scheme (dot filled cells) present their total contribution in the quality rotation. The row *Sum* shows the Q_k from each plot. The row *Deviation* represents $|Q_k - Q_{avg}|$. By subtracting the total deviation from the total score, we have the CRQR of the crop sequence. Other details of the proposed model, such as constraint evaluation, fertilization and profitability, are equivalent to the mono-objective model presented in Chapter 2.

Table 6: A sample solution of the proposed multiobjective CRP model.

Period	Plot 1	Plot 2	Plot 3
1	Barley (Grass) 36.7	Bell Peppers for Fresh Market and Processing, Spring (Nightshade)	Leaf Lettuce for Fresh Market, Winter (Lettuce)
2			
3			
4			
5			
6			Mustard, Spring (Mustard) 22.7
7			
8			
9	Tomato for Fresh Market, (Nightshade)	Summer Squash for Fresh Market (Cucurbit)	
10			
11			
12			Sorghum-sudangrass, Late Spring, Summer (Grass) 37.0
13	Summer Squash for Fresh Market (Cucurbit)		
14		Buckwheat (Buckwheat) 17.3	
15			
16			
17			
18		Oats, Winter Cover (Grass)	Leaf Lettuce for Fresh Market, Fall (Lettuce)
19	Spinach for Fresh Market, Winter (Beet)		
20			
21			
22			Field peas, Fall (Legume) 30.0
23	Rye (Grass) 36.9		
24		Bell Peppers for Fresh Market and Processing, Spring (Nightshade)	
Sum	73.6	48.0	89.7
Deviation	3.1	22.4	19.3
Average	70.4	Total Score	211.2
Total Deviation	44.8	CRQR	166.4

3.3.1 Computational tests and model validation

We have performance several computational test in order to validate the proposed optimization model. Chapter 5 presents several classical optimization techniques along with the proposed multiobjective genetic algorithm. Results and computational tests present in Chapter 7, optimizing the proposed multiobjective model using classic techniques, and in Chapter 8, developing several tests with the proposed model and multiobjective genetic algorithm.

4 UNCOVERING NATURE INSPIRED ALGORITHMS

Evolutionary metaheuristics are computational procedures designed to solve problems. In general, they are iterative algorithms and limited to a number of executions. They based on operators inspired in nature genetic heritage, random variables, competitive and selective pressure, and this combination results in powerful and flexible algorithms able to explore sparse search spaces and provide very good solutions (KNOWLES; CORNE; DEB, 2008).

Genetic algorithms (GAs) were developed using mechanisms of natural selection and genetic, capturing the human essence of searching solutions (GOLDBERG, 1989). Before we move forward into the structure of the GA, we could explore some concepts of the GA using a simple mind experiment. Let us think about solving a 5000-piece puzzle and imagine that for each piece into the right place on our board; we will score some points. At the beginning, we do not have any historical information about the board. We start with an aleatory piece and try to figure out a proper match. Once we found the first pair of pieces, we would have reached a better solution than the previous one. Based on this pair of pieces, our next step is to figure out another one to place together. We could also try to complete another region on our puzzle, and then, try to connect the dots later, as long as we are solving the puzzle and placing pieces together, our imaginary score system will go up. Our first observation is that solving a puzzle is iterative, a piece by piece work. Into the GA's world, a generation represents an iterative process.

Suppose we made significant progress at solving the puzzle, but there are still over a thousand pieces left. Considering an unfilled spot in our puzzle, without deploying any sophisticated technique, we could test all the pieces left, one by one, and ruled out every possibility. This exhaustive procedure describes a random search algorithm. GAs do not get solutions by ruling out all the alternatives, they actually use random numbers inside their operators, but they exploit the search space in an oriented way. We certainly do not want to test all the pieces, but we can look closely at the colors and shapes, set many pieces apart, and suddenly, we would found the missing piece. GAs do not recognize features and patterns as good as human beings, which is not something to worry about. Sometimes the colors and shapes are so blurred or impossible to figure out, but, even though GAs are able to provide solutions.

Understanding how information flows in the GA is essential. When we are solving our puzzle, we look at the board and see the solution composed by pieces and locations. If we would like to share our partial solution, taking pictures will be the easiest way. By looking at the picture, someone far away could try to solve the puzzle. Folks could say try this piece on that place, using a fair description of the piece and its proper location. GAs have to be able to handle data in an equivalent way. Inside GAs, encoding and decoding techniques are deployed to transcript information from the real problem into the GA universe and, once the iterations are over, return back to the real world. And so, a puzzle's solution would not be a picture, but a string inside the GA. The pieces and their locations in the board are represented in the string. It does not have to be a exact match between the real elements of the problem and the string data, but there must be a straightforward procedure to translate their relationship. Once we established this communication channel in the GA, we would ask other folks for help, which are crossover and mutation operators in the GAs.

GAs do not manage a single board at the time, but a collection of boards; let us remember a *board* represents any puzzle solution. Now, think about two boards: one has just the top right corner complete, and the other has only the bottom left filled. Combining these solutions into a third one would create a more complete puzzle, which describes the function of the crossover operator. Using the proper terminology, each solution into the GA is an individual, and the collection of boards is a population and we can create as many individuals as we wish using the crossover operator. Back to our puzzle, we could generate many boards sorting some pieces in aleatory positions. A similar procedure generates the first individuals and initializes the GA population.

If the initial population is composed by poor individuals, the crossover operator will produce few improvements. Thinking about the puzzle, combining highly unfilled regions from two distinct boards could produce worse solutions than the original boards, replacing pieces in the right place with empty spots. Then, lets try to add at least one new piece to each board in the collection before generating more boards. This effort might improve even the worst initial population. Trying to develop small changes in the boards represents the mutation operator. It could also decrease potential stagnation in the GA population. Iterations on the puzzle are close to generations, but that does not mean generations are just small movements. Actually, creating a new generation uses selection, crossover and mutation operators.

In the GA, results are represented by fitness: the better is the solution, the higher is the fitness. About the set of boards or the population, we might have almost filled puzzles and, conversely, almost empty. As we are trying to fill the whole board, keeping the lowest score solution is pointless. Then, we could sort out the population and place the highest at the top of the stack, leaving behind the lower score solutions.

4.1 A new proposal of mono-objective GA

In this section, we present the proposed GA for the CRP, which is called *Priority List Genetic Algorithm* (PLGA). GAs work with the coding of the variable set, not the variables themselves. The term *Priority List* comes from the coding technique developed in this research.

A simple and robust GA, which scores good results in many optimization problems, is developed using three basic concepts: (1) selection, (2) crossover, and (3) mutation. The simplicity of operation is one of the main reasons that makes GAs appealing (GOLDBERG; DEB, 1991).

4.1.1 PLGA's coding and decoding functions

Binary encoding technique is one of the simplest ways to develop crossover and mutation operators in the GA. Binary encoded GAs are very successful in many optimization problems, but in the CRP, the population average solution has improved very little from generation to generation. Search operators using binary encoding are not very efficient in the CRP and fail to generate significant results. Leaving binary encoding behind, the GA performance in the CRP increases using the developed encoding technique called *Priority List*, which is a combination of real encoding parameters and lists.

Let us explain how the priority encoding technique works. Each population member (p_i) holds two data strings (chromosomes): C_1 and C_2. The decision variables x_{ijk} are related to the chromosome C_1 (it represents the crop allocation). A definition of C_1 is $C_1 = \{\{c_1, c_2, \cdots, c_m, \cdots, c_D\} \mid c_m \in \mathbb{R}, 0 \leq c_m \leq 1\}$, where D is the number of rows in the reference matrix H.

In the CRP optimization model, the variable set x_{ijk} represents crop allocations, which is a combination of the type of crop (i), the selected period (j) and the plot (k). Spanning the variable set x_{ijk} results in the reference matrix H, where each row of H holds three cells: (1) crop, (2) period and (3) plot. For instance, $H_1 = [i \quad j \quad k]$ means that allocating crop i at period j in plot k is *possible* (seasonality and production cycle restrictions are respected). The dimension of H matrix is $D \times 3$.

As the total number of rows H matrix is equivalent to the dimension of C_1, the row index of H is related to the C_1 index. In fact, the c_m represents the priority of the allocation in the m row of the reference matrix H. The higher is c_m, $0 \leq c_m \leq 1$, the sooner is allocation H_m processed. Each individual also holds a matrix Q, which keeps the decoded crop sequence. The dimension of Q is $M \times L$. The following pseudo-code describes the decoding function of C_1:

```
/* O Matrix is temporary matrix                                  */
foreach c_m in C_1 do
    O_i := [m | c_m];
/* Sort O rows in descending order: Criterion is the second column of O
   */
O = [{m* | c_m*}, ··· , {D | c_D}, ··· , {m | c_m}], c_m* >= c_m, m* ≠ m;
/* Get a possible allocation in row m of H                       */
foreach m in O^1 (1^st column of O) do
    i = H_[m,1], j = H_[m,2], k = H_[m,3]; if plot k is available then
        if there is none adjacent plot growing crops from the same family of i then
            if there is none predecessor or successor from the same family that is
              already allocated in plot k then
                for n := j to j + t_i do
                    Q_[n,k] = i;
```

For instance, a reference matrix H is similar to the left side in Table 7. Spanning crop i combinations starts with the first seeding period to the last one. According to the matrix H in Table 7, farmers could seed crop *1* from period 17 to 20. Generating combinations with all the plots k completes the reference matrix H. In this example, the plot set is $\{1, 2\}$. The right side of Table 7 is a example of chromosome C_1. The real parameter c_m corresponds to the *Chromosome* column, which is the priority component in the PLGA. The greatest c_m also holds the highest priority in the decoding evaluation.

Table 7: Priority list encoding technique: a real encoding development.

Reference Matrix					Individual string	
Index	**Crop**	**Period**	**Plot**		**Chromosome**	**Index**
1	**1**	**17**	**1**		**0.3527983**	**1**
2	**1**	**18**	**1**		**0.8572196**	**2**
3	**1**	**19**	**1**		**0.0049846**	**3**
4	**1**	**20**	**1**		**0.4896225**	**4**
5	**1**	**17**	**2**		**0.4130837**	**5**
6	**1**	**18**	**2**		**0.3991828**	**6**
7	**1**	**19**	**2**		**0.1883861**	**7**
8	**1**	**20**	**2**		**0.3622583**	**8**

Once a C_1 chromosome of any population member p_n is fully sorted in terms of priority, we will find out something like Table 8. Decoding functions are designed to follow the priority list all the way down, trying to allocate all the combinations in the priority list. Evidently, it is not possible to place all the combinations in the final cropping sequence due to the spatial and temporal constraints. From the sample in Table 8 and 9, once we have placed the first crop, we must analyze if we can place the next one from the list without violating the constraint set. If it is not possible to schedule the next, we will proceed to the following row until we have crossed the full list. Considering Table 9, as the first combination is in place, we could not schedule more crops on the interval displayed in Table 9.

Table 8: Decoding process: sorting out the chromosome from each individual.

Chromosome	Index		Index	Crop	Period	Plot
0.8572196	2		2	1	18	1
0.4896225	4		4	1	20	1
0.4130837	5		5	1	17	2
0.3991828	6		6	1	18	2
0.3622583	8		8	1	20	2
0.3527983	1		1	1	17	1
0.1883861	7		7	1	19	2
0.0049846	3		3	1	19	1

Table 9: Cropping planning: scheduling the highest priority crop.

Period	Plot
17	
18	1
19	1
20	1
21	1
22	
23	

Chromosome C_2 corresponds to the continuous decision variables $F_{N_{\alpha k}}$, $F_{P_{\alpha k}}$ and $F_{K_{\alpha k}}$. A general definition of C_2 is $C_2 = [c_1 \mid c_2 \mid c_3 \mid c_4 \mid c_5 \mid c_6 \mid \cdots] = [F_{N_{\alpha=1\ k=1}} \mid F_{P_{\alpha=1\ k=1}} \mid F_{P_{\alpha=1\ k=1}} \mid F_{N_{\alpha=1\ k=2}} \mid F_{P_{\alpha=1\ k=2}} \mid F_{P_{\alpha=1\ k=2}} \mid \cdots]$, $k \in L$, $\alpha \in \Omega$. There is an index of C_2 for each fertilization decision variable $F_{N_{\alpha k}}$, $F_{P_{\alpha k}}$ and $F_{K_{\alpha k}}$. Hence, matching the index with the proper variable and assigned the value c_m to the decision variable is the decode process of C_2.

Developing an one-year crop scheme for the plot set in Figure 3 would require the set of fertilization variables in Table 10. In the PLGA, each population member p_n would have a string similar to the sample of C_2 chromosome in Table 11. Fertilization constraints in the mono-objective model are described in Equations (2.5), (2.6) and (2.7).

Table 10: Introducing the fertilization variables in the PLGA.

Plot | Fert.	N	P	K
Plot 1	$F_{N_{Year1\|Plot1}}$	$F_{P_{Year1\|Plot1}}$	$F_{P_{Year1\|Plot1}}$
Plot 2	$F_{N_{Year1\|Plot2}}$	$F_{P_{Year1\|Plot2}}$	$F_{P_{Year1\|Plot2}}$
Plot 3	$F_{N_{Year1\|Plot3}}$	$F_{P_{Year1\|Plot3}}$	$F_{P_{Year1\|Plot3}}$

Table 11: Fertilization variables assemble in each member of the population in the PLGA.

Individual String II	
Index	**Chromosome**
1	$F_{N_{Year1\|Plot1}}$
2	$F_{N_{Year1\|Plot2}}$
3	$F_{N_{Year1\|Plot3}}$
4	$F_{P_{Year1\|Plot1}}$
5	$F_{P_{Year1\|Plot2}}$
6	$F_{P_{Year1\|Plot3}}$
7	$F_{K_{Year1\|Plot1}}$
8	$F_{K_{Year1\|Plot2}}$
9	$F_{K_{Year1\|Plot3}}$

4.1.2 Initializing the first population in the PLGA

Once we introduced coding and decoding in the PLGA, it is easy to understand how we can generate an initial population. Each individual is a priority list itself. As the reference matrix follows from the combinations in the database (number of crops, plots, and periods), it is composed at the beginning of the execution and does not change afterward. Since we have a fixed number of indexes, we need to generate random numbers to fill the chromosome column. As small changes in the priority list could alter the output solution impressively, there are no worries about diversity in the initial population using this random initialization.

PLGA's population in generation g (P_g) is composed by individuals $\{p_1, p_2, ..., p_i, ..., p_T\}$, where T is the population size. Each individual has two chromosome structures $C_1\{p_i\}$ and $C_2\{p_i\}$. The first chromosome ($C_1\{p_i\}$) is the encoded priority list, as we have described in the previous subsection. Each position of $C_1\{p_i\}$ has a continuous parameter c_m that belongs to the closed set $[0, 1]$. The total number of positions in $C_1\{p_i\}$ is D, which is the size of the reference matrix H. The second chromosome ($C_2\{p_i\}$) is the group of fertilization variables and its dimension depends on the number of plots (L) and the number of periods (M). In the PLGA, the initial population $P_0 = \{p_1, p_2, \cdots, p_i, \cdots, p_{PS}\}$ is randomly created according to the next steps:

```
foreach p_i in P_0 do
    foreach c_m in C_1{p_i} do
        Generate a random number v ∈ V, V = {v ∈ R | 0 ≤ v ≤ 1};
        c_m = v;
    foreach c_m in C_2{p_i} do
        Generate a random number s ∈ S, S = {s ∈ R | F_min ≤ s ≤ F_max};
        c_m = s;
```

4.1.3 PLGA's fitness function

After completing the decoding process, each member of the population would have established its crop sequence. Individuals storage crop sequences in a $M \times L$ matrix (number of periods $\times$ number of plots), let us called it *allocation matrix*. Scheduled crops fill their corresponding cells, while empty spots represent fallow periods. For instance, if $m_1 \times l_1$ cells are filled with crop n_1, it means that crop n_1 is allocated in this period. If the cycle of crop n_1 was 4 periods long, the crop sequence would have 4 sequential positions filled.

Evaluating the crop sequence is a step-by-step procedure. Each position of the allocation matrix has to be verified. For instance, suppose that the reading mechanism finds crop n_3 in $m_2 \times l_4$, it will register the fertilizer requirements and profits that this allocation produces. If the cycle of crop n_3 was 6 periods long, the reading mechanism would skip cells from $m_3 \times l_4$ to $m_7 \times l_4$ because they are related to the same crop allocation.

Once all the positions have been verified, PLGA's evaluating procedure would compare the fertilization variables with the fertilizer requirements of the crop sequence, creating a nutrient balance on each plot. All constraint sets are already verified on decoding, but the fertilization restrictions. If any fertilizer balance is negative (input resources are not enough to supply all the allocated crops), the solution is unfeasible. In the PLGA, constraint violations are penalized. If all the fertilizer balance is positive, it means we have a feasible solution. Using a pseudo-code, we describe the evaluation function in the PLGA ahead:

```
/* Initialization                                              */
P := 0; V := 0; B_T := 0; C_F := 0;
foreach k in L do
    foreach k in L do
        B_N[k,α] := 0; B_P[k,α] := 0; B_K[k,α] := 0;
foreach i in N do
    W_i = 0;
```

```
/* Navigate allocation matrix Q */
foreach k in L do
    foreach j in M do
        /* Check if Q_[j,k] holds any crop index i */
        if Q_[j,k] is not empty then
            /* Extract the crop i */
            i := Q_[j,k];
            /* Adjust fertilization balance */
            B_N[k,α] := B_N[k,α] + area_k . R^i_N; B_P[k,α] := B_P[k,α] + area_k . R^i_P; B_K[k,α] :=
              B_K[k,α] + area_k . R^i_K;
            /* Update crop i production */
            W_i := W_i + area_k . p_i;
            /* Estimate gross profit from crop i */
            P := P + area_k . l_i . p_i;
            /* Skip the next positions of matrix Q, they are related to
               crop i */
            j := j + t_i;
        else
            j := j + 1;
/* Evaluating demand constraints */
foreach i in N do
    /* If production is smaller than demand of crop i, then increase
       unfeasible variable V */
    if W_i < D_i then
        V =: V + |W_i − D_i|;
/* Check fertilization constraints */
foreach α in Ω do
    foreach k in L do
        C_F =: C_F + F_Nα k . c_N; C_F =: C_F + F_Pα k . c_P; C_F =: C_F + F_Kα k . c_K;
         b_1 =: F_Nα k − B_N[k,α]; b_2 =: F_Pα k − B_P[k,α]; b_3 =: F_Kα k − B_K[k,α]; if b_1 < 0
         then
            B_T =: B_T + b_1;
        if b_2 < 0 then
            B_T =: B_T + b_2;
        if b_3 < 0 then
            B_T =: B_T + b_3;
/* If the nutrient supply is smaller than crop nutrient demand, then
   increase unfeasible variable V */
V := V + B_T;
P := P − C_F ;          /* Estimate net profit: profit objective function */
return V ;              /* If V > 0, the crop sequence in Q is unfeasible */
```

4.1.4 PLGA's selection operator: Tournament Selection

GAs can improve the fitness of each succeeding generation using selection pressure. Low pressure may slow down the convergence rate, and too high pressure might increase the probability of premature convergence and drive the GA to sub-optimal solutions. Tuning the selection pressure is essential to improve the algorithm's performance in distinct domains. Miller e Goldberg (1995) defines selection pressure as the degree to which the better individuals are preferred. In general, the selection pressure is comprised entirely of the selection scheme, although there are sorting schemes among individuals that can also increase the selection pressure in the GA.

We have already featured the importance of selection pressure, but we have not placed selection mechanisms into the GA's structure. A selection mechanism is expected to be simple coded and highly efficient, whether in parallel or nonparallel architectures. Although the crossover operator generates new off-springs from a pair of parents, parents' selection is fundamental in the process. The new offspring inherit parents' genes, a poor selection might deplete the fitness over succeeding generations.

The tournament selection bases on a simple idea: select some number of individuals randomly from a population and choose the best individual from this group. As forming a new generation requires many off-springs, the tournament is repeated as many times as need to fill the mating pool. Although tournaments are usually held between pairs of individuals, large tournaments are also efficient (GOLDBERG; DEB, 1991).

4.1.5 PLGA's crossover operator: Laplace Crossover

Deep et al. (2007) developed a real encoded crossover operator called Laplace Crossover. It bases on the Laplace distribution and it is also a centrist parent operator. The density function of the Laplace distribution is:

$$f(x) = \frac{1}{2b} e^{-\frac{|x-a|}{b}}, \quad -\infty < x < \infty \tag{4.1}$$

The Laplace distribution is defined by Equation (4.2), where a is a location parameter and b is a scale parameter, which $b > 0$. Off-springs will be close to the parents when the scale parameter is small ($b \leq 0.5$) and quite distant when it is large ($b = 1$).

$$F(x) = \begin{cases} \frac{1}{2} e^{\frac{|x-a|}{b}}, & x \leq a \\ 1 - \frac{1}{2} e^{-\frac{|x-a|}{b}}, & x > a \end{cases} \tag{4.2}$$

The Laplace Crossover generates two off-springs as follows:

1. Define the location parameter (a) and the scale parameter (b);
2. Select a pair of parents $x^1 = [x_1^1, x_2^1, \cdots, x_n^1]$ and $x^2 = [x_1^2, x_2^2, \cdots, x_n^2]$
3. Initialize the random parameters u_i and r_i, which are in the close interval $[0, 1]$:
4. Evaluate the β parameter, which is given by:

$$\beta = \begin{cases} a - b \ . \ log(u_i), & \text{if } r_i < 0.5 \\ a + b \ . \ log(u_i), & \text{if } r_i \geq 0.5 \end{cases} \tag{4.3}$$

5. Generate two off-springs as follows:

$$y_i^1 = x_i^1 + \beta \ . \ |x_i^1 - x_i^2| \tag{4.4}$$

$$y_i^2 = x_i^2 + \beta \ . \ |x_i^1 - x_i^2| \tag{4.5}$$

In the PLGA, Laplace crossover has been adapted according to the characteristics of the individuals and their chromosomes. Figure 4 describes the adapted crossover operator.

Each individual $(p_1, p_2, \cdots)$ has its own chromosomes C_1 and C_2:

- C_1: the encoded priority list chromosome, where $C_1 = \{c_1, c_2, \cdots, c_m, \cdots, c_D | \{c_1, c_2, \cdots, c_m, \cdots, c_D\} \in [0, 1]\}$;
- C_2: the chromosome of fertilization variables, where $C_2 = \{c_1, c_2, \cdots, c_m, \cdots, c_E\} | \{c_1, c_2, \cdots, c_m, \cdots, c_E\} \in [F_{min}, F_{max}]$.

There are populations P_g and PN_g:

- T: the size of population P_g;
- P_g: the current population of generation g, where $P_g = \{p_1, p_2, \cdots, p_t, p_w, \cdots, p_T\}$;
- N: the size of population PN_g;
- PN_g: the population of new individuals of generation g, where $PN_g = \{p_1, p_2, \cdots, p_n, p_{n+1}, \cdots, p_N\}$.

Other parameters from the adapted Laplace crossover operator are as follows:

- p_t and p_w: selected individuals from population P_g (parents);
- p_n and p_{n+1}: off-springs, individuals from population PN_g (sons);

- *exc*: probability of generating alleles using regular crossover or the adapted Laplace crossover;
- K: the chromosome size.

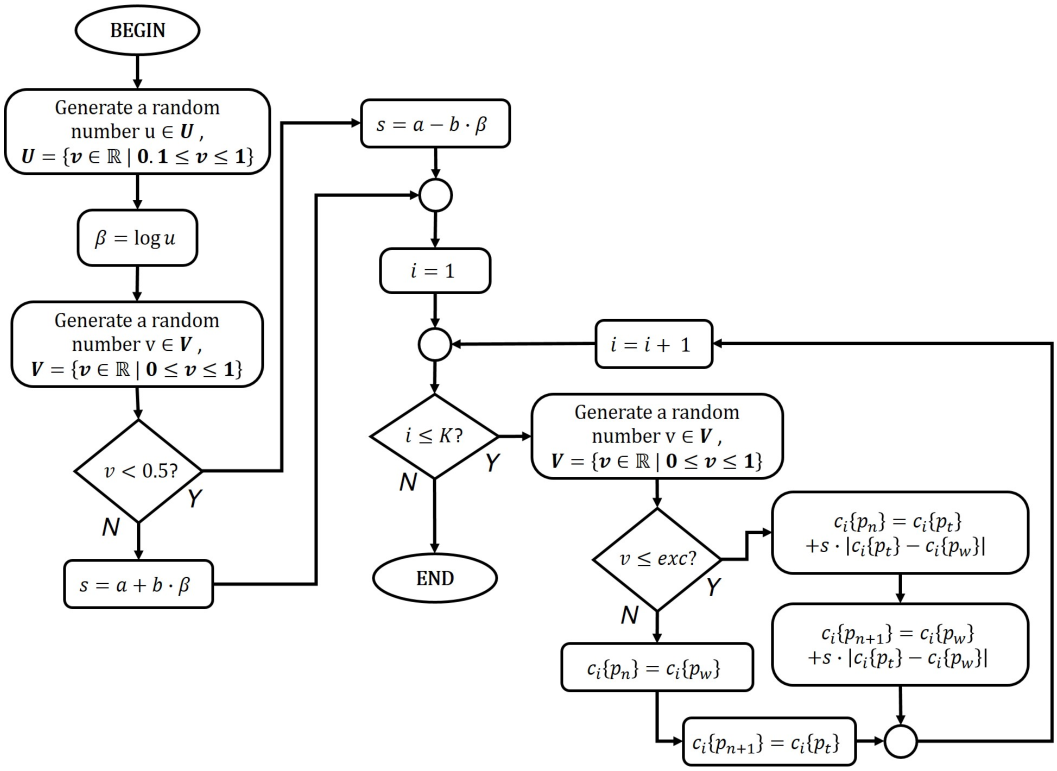

Figure 4: PLGA's adapted Laplace crossover operator.

4.1.6 PLGA's mutation operator: Power Mutation

Deep e Thakur (2007) developed a mutation operator for real coded GAs called Power Mutation. The parameter p is the mutation index and controls the strength of the mutation; the smaller is p, the less perturbed is the muted solution. The following procedures describe this mutation operator:

1. Initialize the random numbers $s_i \in [0, 1]$ and $r \in [0, 1]$;
2. Evaluate the parameters s and t:

$$s = s_i^p \quad , \quad t = \frac{\hat{x} - x^L}{x^U - \hat{x}} \tag{4.6}$$

3. Proceed the mutation and create the muted solution as follows:

$$x = \begin{cases} \hat{x} - s \ . \ (\hat{x} - x^L), & if \ t < r \\ \hat{x} + s \ . \ (x^U - \hat{x}), & if \ t \geq r \end{cases} \tag{4.7}$$

The mutation operator proposed by (DEEP; THAKUR, 2007) have been adapted in PLGA, following the characteristics of the CRP. Figure 5 describes the PLGA's adapted mutation operator.

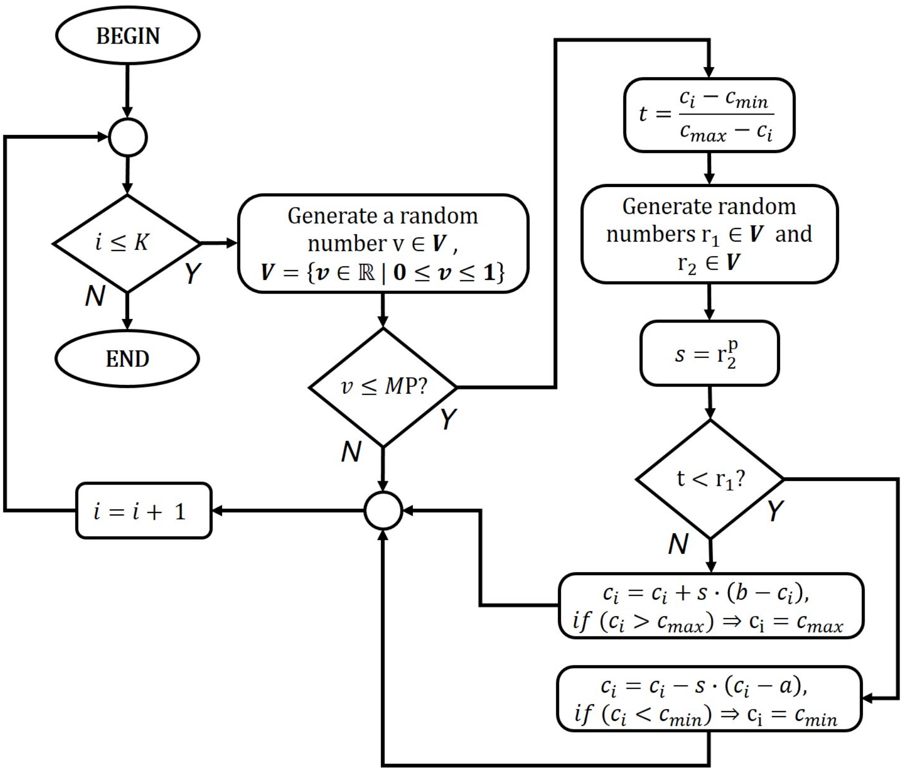

Figure 5: PLGA's mutation operator.

There are two chromosome from each individual (C_1 and C_2). The total number of alleles in the C_1 and C_2 is represented by D and E, respectively. Mutation operator may introduce mutations in any allele. In Figure 5, K represents the number of alleles in the chromosome under evaluation. If a random number v is lesser or equal the mutation probability (MP), mutation will alter gene value c_i from its initial state; otherwise, c_1 remains unchanged and the algorithm proceeds to the next allele. c_{min} and c_{max} are based on the mathematical model. Mutating the gene value follows the definition in Equation (4.7).

It is unlike to generate a mutated gene larger than its upper bound or smaller than its lower bound. Anyway, if the mutated c_i was larger than c_{max}, it would take c_{max} value. In the opposite, if c_i is lower than c_{min}, it would be equal to c_{min}. x^U is an upper bound of allele x and x^L is a lower bound of allele x.

4.1.7 Describing the PLGA's flowchart

In this subsection, we would like to connect all the elements of the proposed GA. The flowchart in Figure 6 projects a holistic perspective of PLGA and it also connects all the previous subsections about the genetic operators in the PLGA.

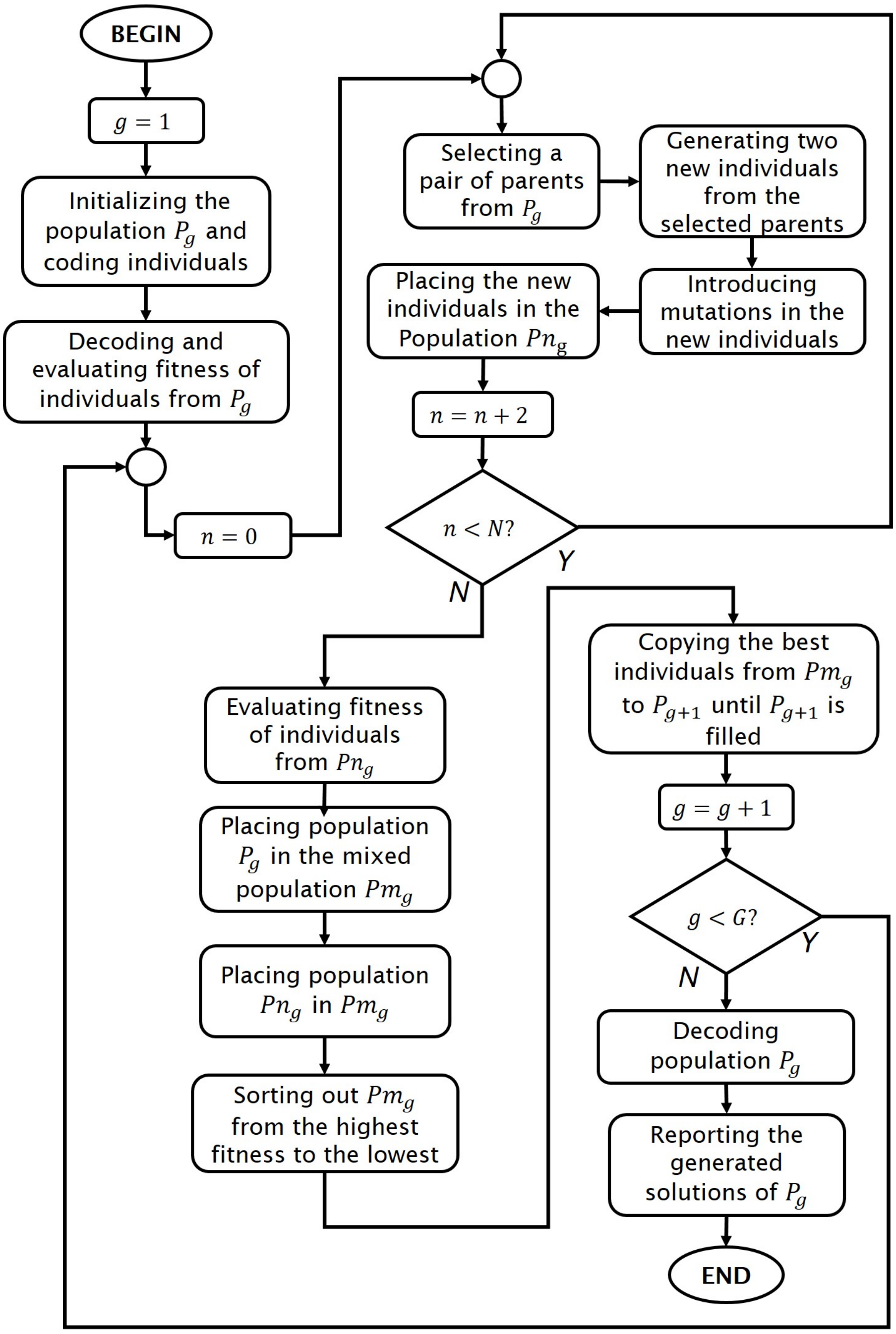

Figure 6: PLGA's flowchart.

The initialization is basically the creation of the initial population, which is done by a random procedure, as we have already described previously. Random initialization seems a good option due to the rich diversity introduced in the population.

The optimization process stops when the generation counter g reaches the limit of G. At this point, all the individuals from the final population P_g are decoded and evaluated. Reporting functions export the generated solution in a comprehensive output.

Generation flow produces off-springs (new individuals). Selecting prospective parents is the first step, then the crossover operator combines partial strings from each parent to produce new individuals. The mutation operator introduces some new elements in the generated population. Developing mutations in a small proportion should keep evolution in place on the execution. In the PLGA, the selection operator picks up two parents, and a crossover operator produces two sons each time, repeating until we have a new population of N individuals.

The mixed population Pm_g combines all the individuals from P_g and Pn_g. The mixed population is a copy of both populations. After the combination of individuals, a sorting function organized them from the highest fitness to the lowest. As Pm_g is bigger than P_g, just the best individuals of Pm_g are selected to compose P_{g+1}.

5 OPTIMIZATION TECHNIQUES IN MULTIOBJECTIVE PROBLEMS

Although we have already proposed a multiobjective model for the CRP in Chapter 3, we still need to present a powerful optimization techniques to achieve our main goal that is to provide reliable tools for the agribusiness management. Multiobjective definitions and concepts are presented in this chapter, as well as some multiobjective optimization techniques. We are not trying to produce a wide survey of the multiobjective optimization methods, but we would like to explore essential concepts that are deeply connected to the overall research.

Besides the general problems found in the mono-objective optimization process, multiobjective optimization ought to overcome a few barriers. Conflicting objective functions are not unusual, which means that improve one objective may deplete the other. So there is not a solution that gets the best of all the objectives simultaneously. Immeasurable objective functions represent another challenge to the optimization method and also, balancing the uncertain of the decision-maker may require broad diversity in the solution set. Although multiobjective modeling increases the problem complexity, it must be noted that real-world applications are much better represented in the multiobjective model (CLíMACO; ANTUNES; ALVES, 2003).

5.1 Essential definitions

Modeling a multiobjective problem allows the optimization of all the possible goals simultaneously (FERREIRA, 1999). A multiobjective problem is stated as follows in the space of decision variables:

$$\textit{minimize} \quad F(x) = [f_1(x),\ f_2(x),\ \cdots,\ f_m(x)] \tag{5.1}$$

$$\textit{subject to} \quad x \in \mathcal{S} \tag{5.2}$$

Although maximization problems are the main course of this research, concepts, and definitions presented in this chapter are based on the standard minimization model. $F := [f_1,\ f_2,\ \cdots,\ f_r]$ ($r \geq 2$) is a vector of objective functions and $S \subset \mathbb{R}$ is a feasible set of the problem.

In general, solving a multiobjective problem is the effort to reach a decision vector that fulfills the problem constraints and is a distinguished solution in terms of the decision-maker goals. In multiobjective optimization, local and global minimums rarely appear in the solution set. In opposite of the $\mathbb{R} = \mathbb{R}^1$, the multidimensional space $\mathbb{R}^r$ is a partially ordered space, and then, not all the parameters can be compared ($y^1 := f_1(x)$ and $y^2 := f_2(x)$ are incomparable).

- **Definition 1**: (Pareto Optimal) A solution vector $x' \in \mathcal{S}$ is a Pareto-optimum if there are none other vector solution $x \in \mathcal{S}$ such that $f_i(x') \preceq f_i(x)$ for all $i = 1, \cdots, k$ and $f_j(x') < f_j(x)$ for at least one index j. Hence, a solution vector x' is Pareto optimum only if it is a non-dominated solution by $\mathcal{S}$.

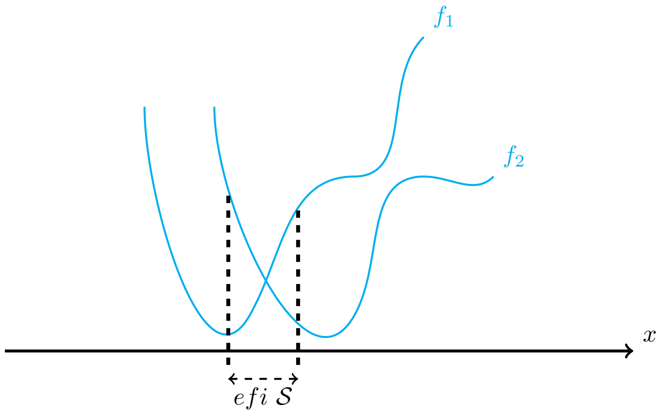

Figure 7: General concept of efficient solutions.

- **Definition 2**: (Pareto Dominated) Let $x_K \in \mathcal{S}$ and $x_L \in \mathcal{S}$. In the minimization problem, x_L is a dominated solution by x_L, $F(x_L) \succeq F(x_K)$, if and only if:

$$\forall i \in \{1,\ 2,\ \cdots,\ m\}, f_i(x_K) \preceq f_i(x_L) \ and\ \exists j \in \{1,\ 2,\ \cdots,\ m\}, f_j(x_K) < f_j(x_L) \tag{5.3}$$

The search space $\mathcal{S}$ is partially ordered in the sense that two arbitrary solutions are related in two possible ways: either one dominates the other or none dominates. Let be x_1 and $x_2 \in \mathbf{S}$. x_1 dominates x_2, $F(x_1) \preceq F(x_2)$ (minimization problem), if and only if: $\forall\ i \in \{1, 2, ..., m\}, f_i(x_1) \leqslant f_i(x_2)$ and $\exists\ j \in \{1, 2, ..., m\}, f_j(x_1) < f_j(x_2)$. That is, x_1 is not worse than x_2 in any of the objectives and is better in at least one objective (ABRAHAM; JAIN, 2005). And x is a Pareto-optimal solution if and only if x is non-dominated in relation to $\mathbf{S}$, that is, no vector in the search space dominates x. In multiobjective optimization, expected solutions are composed by a set of equilibrium points, that is, a family of solutions considered equivalent and higher than the rest of the solutions.

The range of $F(x)$ is $\mathbf{Y} = F(\mathcal{S}) = \{y \in \mathbb{R}^r \ :\ y = f(x), x \in \mathcal{S}\}$. In the objective space, the problem is represented as follows:

$$\begin{aligned} &minimize && y = [y_1,\ y_2,\ \cdots,\ y_r] && (5.4)\\ &subject\ to && y \in \mathbf{Y} \subset \mathbb{R}^r && (5.5) \end{aligned}$$

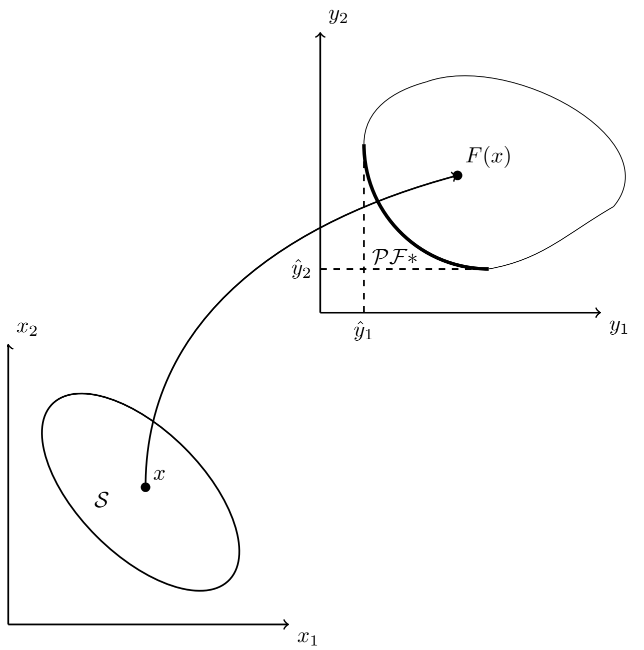

Figure 8: Representing a general solution in the objective space.

- **Definition 3**: A Pareto-optimum set is stated as:

$$\mathcal{P}* = \{x \in \mathcal{S} \mid \nexists \gamma \in \mathcal{S},\ F(\gamma) \preceq F(x)\} \tag{5.6}$$

- **Definition 4**: A Pareto efficient frontier can be defined as follows:

$$\mathcal{PF}* = \{F(x) = [f_1(x),\ f_2(x),\ \cdots, f_m(x)]\ :\ x \in \mathcal{P}*\} \tag{5.7}$$

- **Definition 5**: If for all $x \in \mathcal{P}*$, there is no solution γ, where $\|\gamma - x\|_\infty \leq \epsilon$, that dominates any solutions in the Pareto-optimum set, then they are a set of local Pareto-optimum solutions. $\epsilon \geq 0,\ \epsilon \in \mathbb{R}$ and the vector γ is selected from a small perturbed neighborhood of the Pareto solution x.

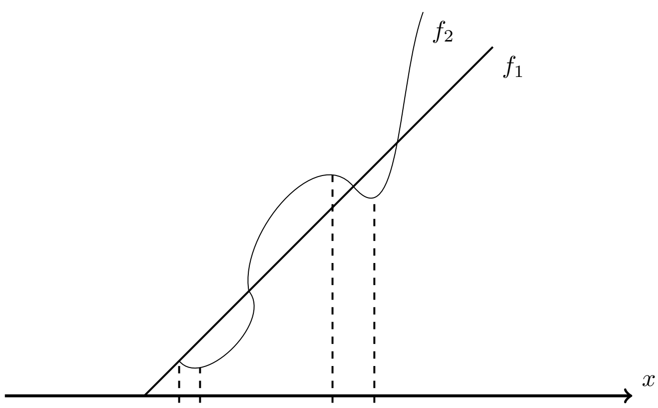

Figure 9: Local efficient solutions.

- **Definition 6**: If it does not exist any solution in the search space that dominates any other element in the set $\mathcal{P}*$, hence all the solutions in $\mathcal{P}*$ establish a global Pareto-optimum set.

- **Definition 7**: A decision vector $x^* \in \mathcal{S}$ is weakly Pareto optimal if there does not exist another decision vector $x \in \mathcal{S}$ such that $f_i(x) < f_i(x^*)$ for all $i = 1, \cdots, r$. An objective vector $y^* \in \mathcal{Y}$ is weakly Pareto optimal if there does not exist another objective vector $y \in \mathcal{Y}$ such that $y_i < y_i{}^*$ for all $i = 1, \cdots, r$; or equivalently, if the decision vector corresponding to it is weakly Pareto Optimal (MIETTINEN, 1998) . Moving from a weakly Pareto optimal point in any direction and improving all the objective functions simultaneously is impossible (MARLER; ARORA, 2010).

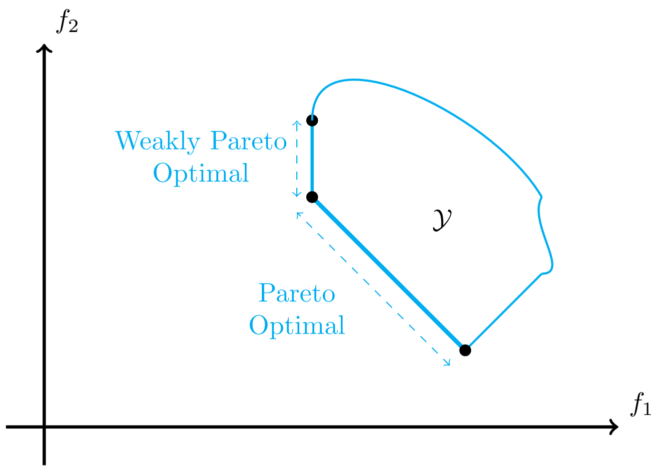

Figure 10: Pareto optimal and weakly Pareto optimal frontiers.

5.2 Classical optimization methods

Decisions on multiobjective problems are based on a set of equally reasonable solutions. A collection in which the decision-maker (DM) can select the desired solution that matches its aspiration levels (BRANKE et al., 2008). There are several deterministic and heuristic methods for solving multiobjective optimization problems. Some of them have been analyzed in this research, where we are looking for characteristics that could contribute to the CRP and better achieve the DM's aspirations. In *a priori* methods, DM's preferences are defined before the optimization process takes place. The original multiobjective problem is reformulated to solve using *a priori* methods.

A posteriori methods initially generate Pareto-optimal frontiers without considering any general preference. Then the DM would have to choose a compromise solution from the solution set, according to some particular criterion or preference.

In this research, DM represents the farmer's standpoint. Trying to generate efficient frontiers with high resolution would enhance the final decision. We describe classical multiobjective optimization techniques ahead. Statements and formulations in this subsection are based on a general minimization problem in Equations (5.8) and (5.9), where m is the total number of objective functions, x represents the decision variable and $\mathbf{S}$ defines the constraint set which x is subjected to.

$$\min \sum_{i=1,\cdots,m} f_i(x) \tag{5.8}$$

$$\text{subject to} \quad x \in \mathbf{S} \tag{5.9}$$

- The Weighted Sum: It has been detailed in (ZADEH, 1963) and (GASS; SAATY, 1955). This approach is quite intuitive and, in general, many multiobjective applications have based on this technique without being fully aware. A consistent weight selection could produce multiple solution points or generate a particular solution based on the preferences incorporated into the weight configuration. A definition of the Weighted Sum follows ahead, where w_i represents the weight of objective i.

$$\min \sum_{i=1,\cdots,m} w_i f_i(x) \tag{5.10}$$

$$\text{subject to} \quad x \in \mathbf{S} \tag{5.11}$$

$$w_i \geq 0 \tag{5.12}$$

  A weight set represents the importance of each objective function in the optimization. Selecting weights that prioritize one objective instead of others can be challenging; in general, objective functions are indistinct. Even with a satisfactory selection of weights, the final solution might not reflect precisely the intended preferences (MARLER; ARORA, 2010). In fact, setting weights to get differences in the objective-function magnitudes is a fundamental deficiency of the Weighted Sum Method.

- Goal programming: Charnes, Cooper e Ferguson (1955) introduced one of the earliest goal programming formulations. Further developments and the proper terminology is given in (CHARNES; COOPER, 1961). A classical approach for goal programming presents in (CHARNES; COOPER, 1977). Goal programming quickly became popular in the field of multiobjective optimization methods because it is a straightforward procedure and easy to formulate. Although goal programming is a popular technique, some basic errors are usually due to improper implementations and lack of practice, such as the generation of Pareto-inefficient solutions, lack of weight sensitivity analysis, direct comparison of incommensurable goals and ineffectual representation of decision-maker preferences (JONES; TAMIZ, 2010). A classical definition is presented ahead. Consider t_i as the objective i target; d_i^+ indicates how much $f_i(x)$ exceeds t_i and d_i^- evaluates the amount $f_i(x)$ lacks to reach t_i.

$$\begin{aligned}
min_{x,d^+,d^-} \quad & (\sum_{i=1}^{m}(d_i^+ + d_i^-)^p)^1/p, \quad p \geq 1 && (5.13)\\
\text{subject to} \quad & f_i(x) - d_i^+ + d_i^- = t_i, \quad i = 1, \cdots, m && (5.14)\\
& d_i^+ \geq 0, \quad d_i^- \geq 0, \quad d_i^+ \,.\, d_i^- = 0, \quad i = 1, \cdots, m && (5.15)\\
& x \in \Omega && (5.16)
\end{aligned}$$

- The Achievement Scalarizing Function Method: Wierzbicki (1986) introduced scalarizing functions in the multiobjective optimization problems. An arbitrary vector y_{asp} holds the aspiration levels of the decision-maker, which is projected on the set of Pareto Optimal (P.O.) solutions. In this approach, there is no requirement for an idealistic reference point, and it is not based on metric distances.

$$\begin{aligned}
\min \quad & \max_{i=1,\cdots,m} \; \frac{f_i(x) - y^{asp}}{{y_i}^{nadir} - {y_i}^{ideal}} && (5.17)\\
\text{subject to} \quad & x \in \mathcal{S} && (5.18)\\
& {y_i}^{ideal} = \min_{x \in \mathcal{S} | x \text{ is P.O.}} \; f_i(x), \quad \forall i = 1, \cdots, m && (5.19)\\
& {y_i}^{nadir} = \max_{x \in \mathcal{S} | x \text{ is P.O.}} \; f_i(x), \quad \forall i = 1, \cdots, m && (5.20)
\end{aligned}$$

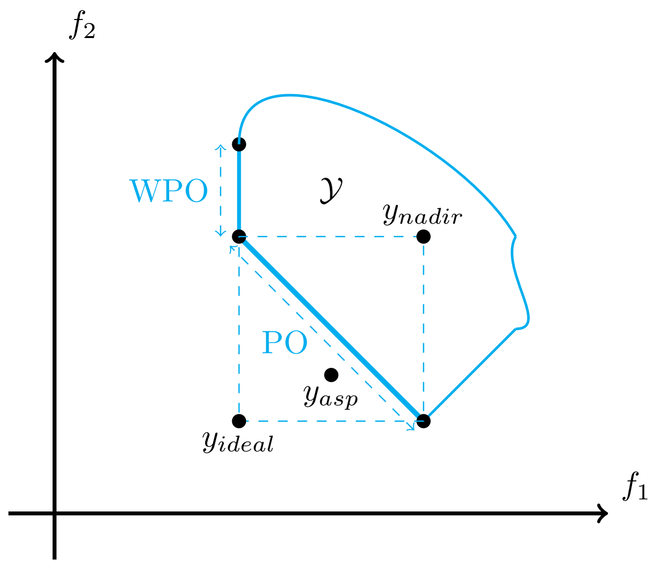

Figure 11: Describing the Achievement Scalarizing Function Method.

Figure 11 describes the parameter set in the Achievement Scalarizing Function Method. Considering the bi-objective minimization problem, y_{ideal} is the combination of the best solution of f_1 and f_2, which is unfeasible unfortunately. In the opposite, there is y_{nadir} which is the worst combination of f_1 and f_2. y_{asp} would indicate the direction of search, introducing the DM's preferences. PO stands as Pareto-optimal frontier, WPO as weakly Pareto-optimal frontier, and $\mathcal{Y}$ represents the objective space.

- Optimization based on boundaries: This method requires previous information from the DM, it has to select one particular objective and establish boundaries for others. A regardless choice of lower bound (l_j) and upper bound (L_j) could turn the problem unfeasible.

$$\min \quad f_i(x) \tag{5.21}$$
$$\text{subject to} \quad l_j(x) \leq f_j(x) \leq L_j, \; \forall \; j = 1, \cdots, m \mid j \neq i \tag{5.22}$$
$$x \in \mathbf{S} \tag{5.23}$$

- ϵ-Constraint Method: This method has been introduced in (CHANKONG; HAIMES, 1983) and (HAIMES YV; LASDON; WISMER DA, 1971). The ϵ-constraint method is very intuitive and simple to apply in multiobjective problems. However, without scalarizing the objective functions properly, it is a great deal to achieve a well-distributed spread of the Pareto frontier. A definition follows in the equations ahead, where upper bounds (ϵ_j) are input parameters, and $f_i(x)$ is the selected objective function to be minimized.

$$\min \quad f_i(x) \tag{5.24}$$
$$\text{subject to} \quad f_j(x) \leq \epsilon_j, \; \forall \; j = 1, \cdots, m \mid j \neq i \tag{5.25}$$
$$x \in \mathbf{S} \tag{5.26}$$

5.3 A novel multiobjective genetic algorithm

Generating a unique solution could not be the best outcome we expect from multiobjective optimization algorithms. Multiobjective models may require a solution set, which should offer enough options for comparison. In classical optimization methods, multiobjective problems are converted to equivalent single-objective models, which considers some user preferences leading to a distinctive solution. Then, the DM has to alter input parameters and rerun the algorithm to get other characteristic solution. In opposite, multiobjective evolutionary algorithms can find multiple efficient solutions in a single execution (DEB et al., 2002).

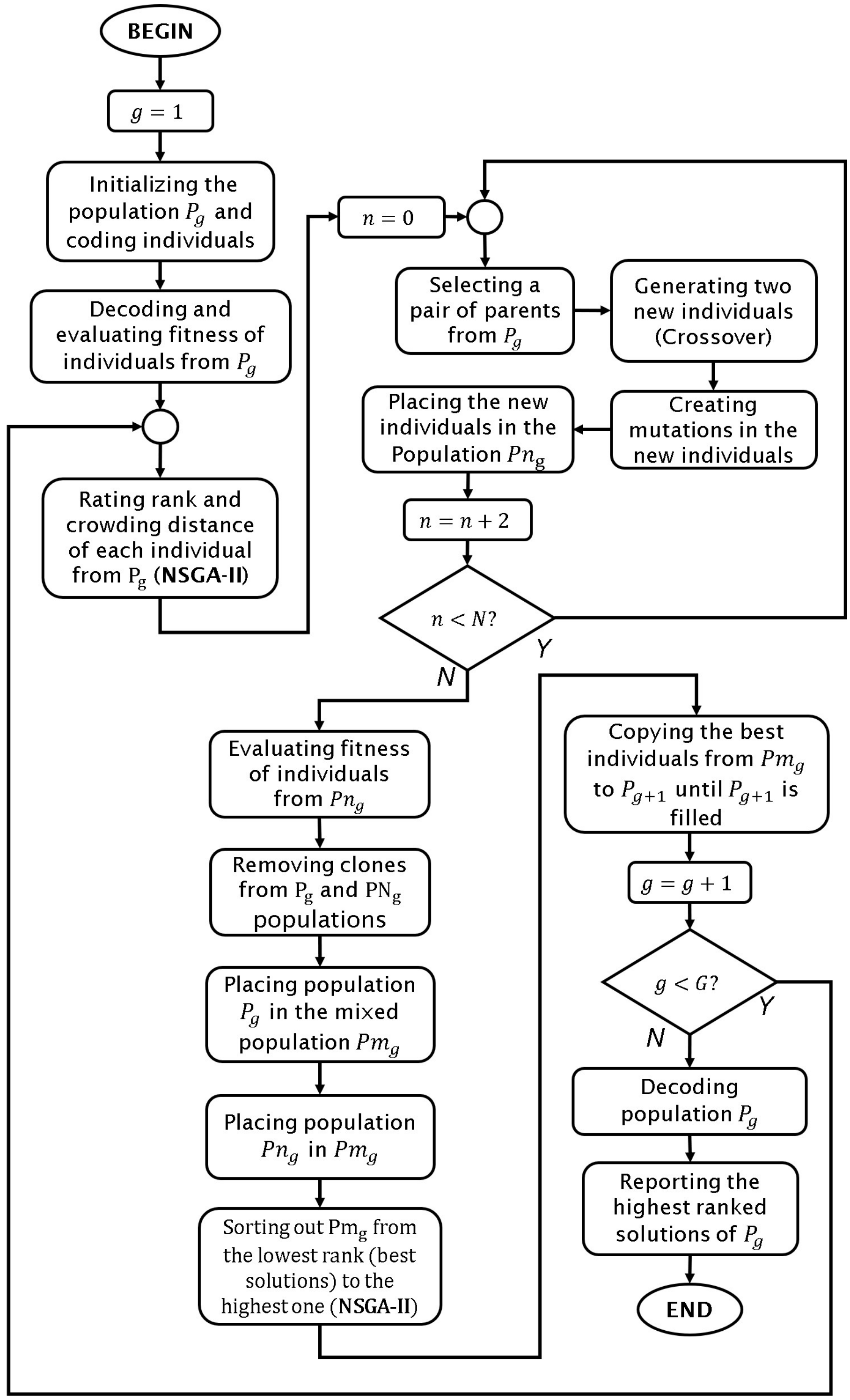

Figure 12: PLMGA's flowchart A.

In these sections, we described the *Priority List multiobjective Genetic Algorithm* (PLMGA). The proposed algorithm has the same operators of the PLGA combined with the concepts in the *Non-dominated Sorting Genetic Algorithm II* (NSGA-II). Figure 12 provides a general description of the proposed algorithm PLMGA using a flowchart.

In the PLMGA, initialization, coding, decoding, and evaluating function remain the same from the mono-objective PLGA. A random procedure generates the initial population. Coding and decoding follow the same principles described in Subsection 4.1.1.

5.3.1 The role of NSGA-II modules in the PLMGA

Multiobjective evolutionary algorithms (MOEAs) can produce multiple Pareto-optimal solutions in one single simulation run. Deb et al. (2002) developed an improved version of the non-dominated sorting genetic algorithm (NSGA), which they have called NSGA-II. The algorithm's release aims to reduce the high computational complexity of non-dominated sorting and lack of elitism, which could speed up performance. Any evolutionary algorithm (EA) can incorporate NSGA-II modules since they work with a population of solutions.

All population members are assigned a classification. It begins with the first frontier (the best individuals). Using a flowchart, we describe the first classification in Figure 13. In the PLMGA, we handle over the mixed population (P_{m_g}) in order to get back a sorted population P_{g+1}.

- F_1: first frontier;
- Pm_g: mixed population;
- K: quantity of individuals in the mixed population;
- S_{p_k}: set of solutions that the solution p_k dominates;
- n_{p_k}: domination counter;

At the beginning, first frontier is empty. Then, we compare each member of the mixed population (p_k) with each other (p_q) as long as they are not the same individual. If the solution p_k dominates p_q (considering a minimization problem, a solution p_k that dominates p_q is represented as $p_k \preceq p_q$), then $p_q \in S_{p_k}$. Else, if the solution p_q dominates p_k, we increment the domination counter n_{p_k}. Once we ruled out all the solutions of the population, if $n_{p_k} = 0$, then the solution p_k is non-dominated and $rank_{p_k} = 1$.

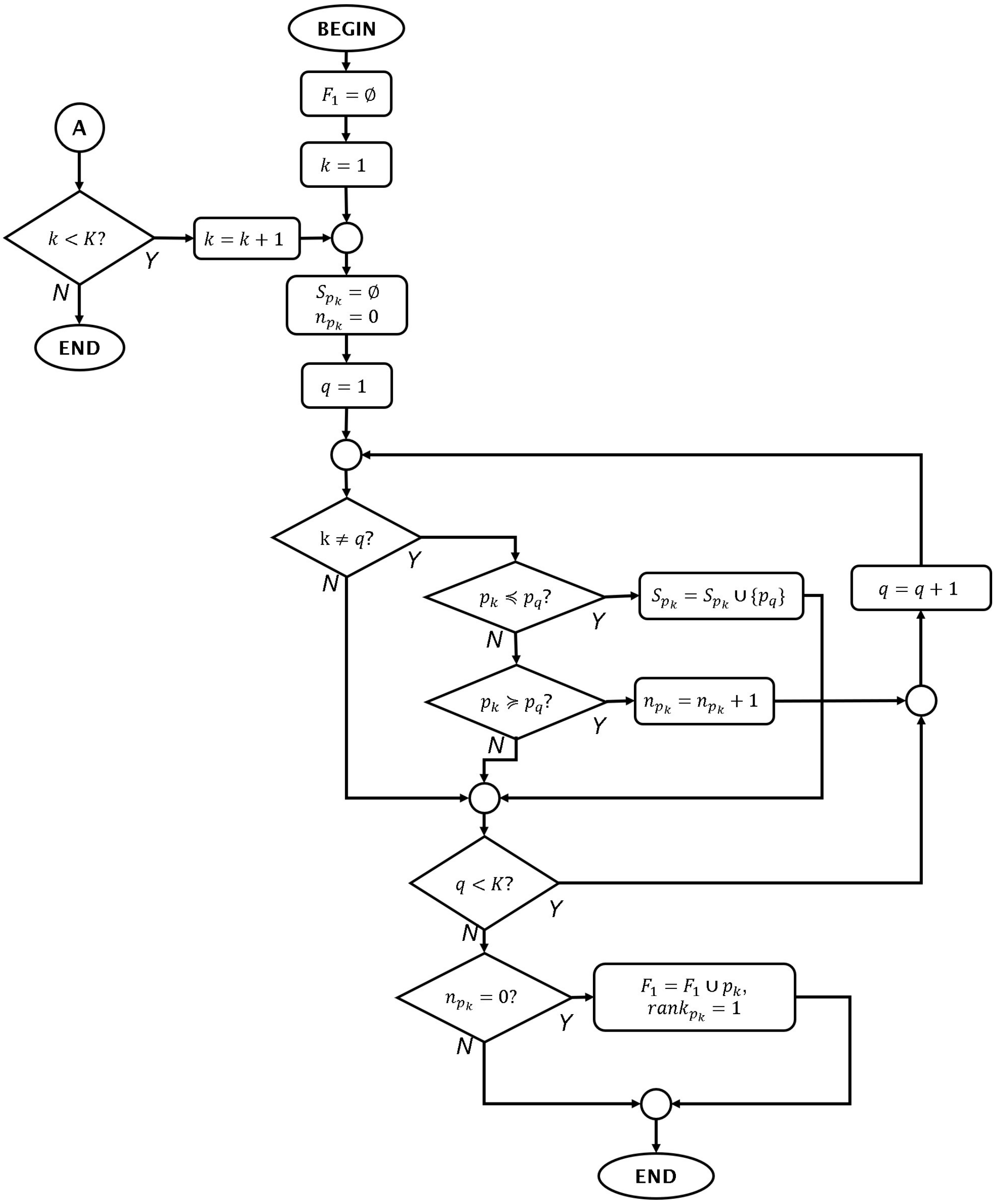

Figure 13: First frontier classification.

After completing the first frontier, we have to find out enough frontiers to fill the population P_{g+1}; we need at least T individuals for that. If we already had T individuals from the first frontier, we could stop sorting the population. We describe the frontier assessment in Figure 14.

- P_{g+1}: new population;
- T: quantity of individuals in the new population;
- Z: temporary set that holds the next frontier;
- p_u and q_w: both are solutions, q represents a dominated solution;

- U_{F_l}: size of the frontier F_l.

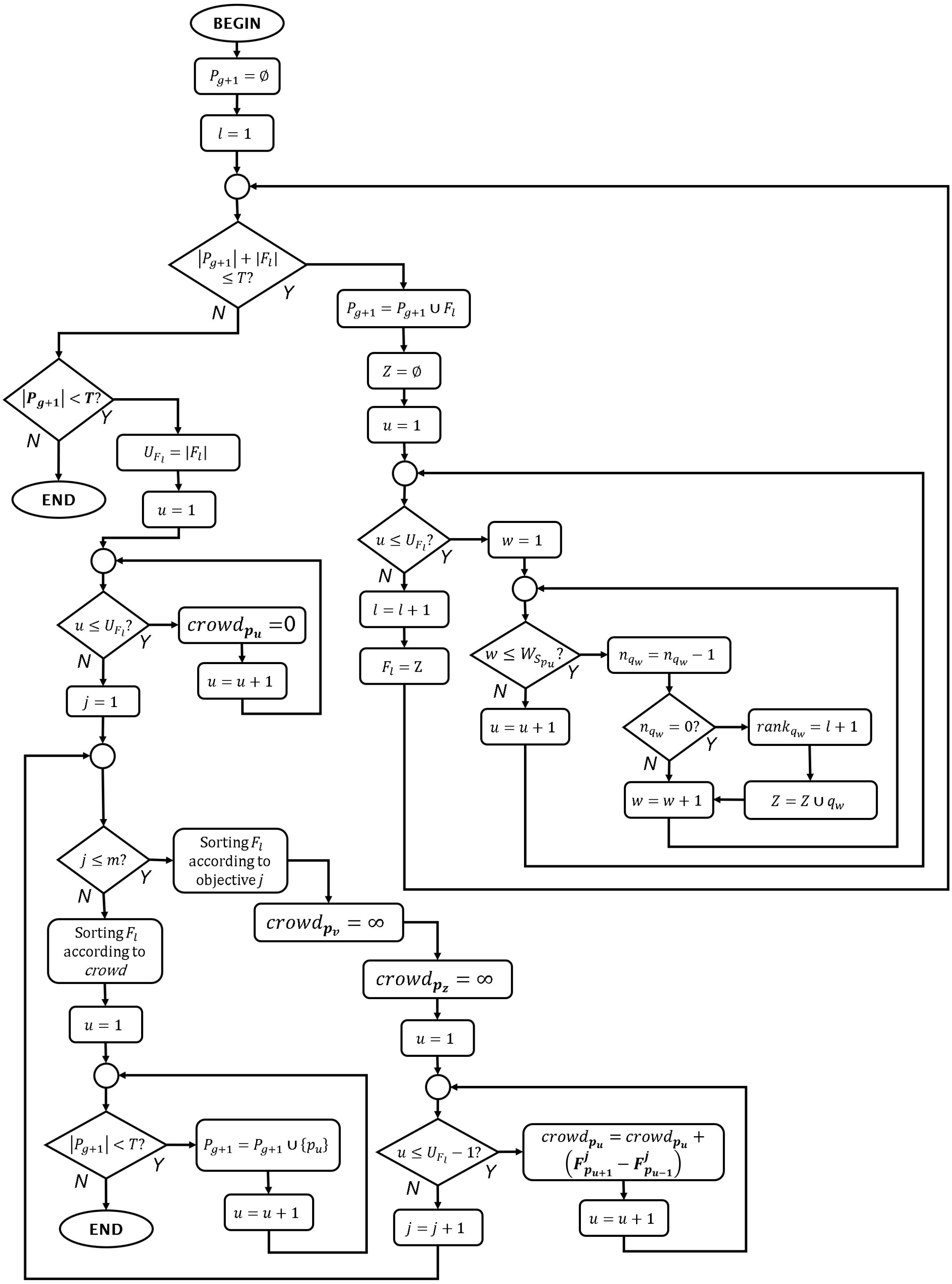

Figure 14: Composing frontiers in the NSGA-II: B.

If there are still empty spots in the population P_{g+1} and we cannot place a complete frontier, we will select individuals from the last frontier using crowding distance.

After ranking all the members of the mixed population Pm_g, we can transfer them to the population P_{g+1}. Figure 15 represents the selection of individuals for population P_{g+1}. Placing F_1 and F_2 does not exceed the maximum number of members T, and we can copy these frontiers completely. But, if we try to copy the full F_3 frontier, there are already too many members in P_{g+1}. If all the solutions from F_3 have the same rank, we need another criterion to decide which should move forward to P_{g+1} or stay behind. Then, we select members based on the crowding parameter, prioritizing solutions from less populated regions (with the highest crowding).

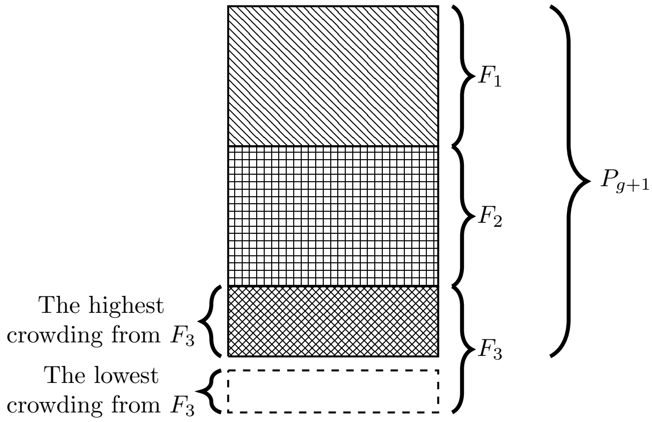

Figure 15: Assigning individuals to the new population P_{g+1}

6 MONO-OBJECTIVE OPTIMIZATION: DISCUSSION ON RESULTS

Designing genetic algorithms (GAs) parallels natural systems. Three operators define performance in GAs: (1) selection, (2) crossover, and (3) mutation. According to Schott (1995), these operators evolve a population of potential members to increase the average solution of the optimization problem.

Mutation rate, crossover probability, and population size define the GA's efficiency. In this chapter, several of PLGA configurations were explored. We want to analyze performance and robustness. There are two dimensions of CRP in these computational tests, we selected a few numbers of plots in the first one (Section 6.1) and, in the second, a complex group of plots (Section 6.2). Appendix D presents the configuration settings in these computational tests.

6.1 Instance A

The crop set and its parameters are presented in Appendix A. The total number of crops in this database is 67 ($N = 67$) and the total number of crop's families is 11 ($N_f = 11$). We set a long-term planning horizon of 8 years (in the PLGA, it is a total of 192 periods [$M = 192$]). There is a group of 3 plots in this computational test ($L = 3$), Table 12 describes the adjacency among plots and the cultivable area of each plot. We neglect the main diagonal in Table 12, but the other positions in the table filled with dots represent the plots that are adjacent to each other.

Table 12: Plot's adjacency of Instance A.

Plot—Plot	Plot 1	Plot 2	Plot 3	Area (ac.)
Plot 1				1.00
Plot 2				1.50
Plot 3				0.80

The maximum, minimum, and standard deviation are the performance parameters we have selected to evaluate PLGA. Running 10 times each parameter configuration avoids misfortunes among executions. Average elapsed time in this chapter also bases on ten executions.

The total number of generations is the first parameter we would like to observe. A generation is the process of creating a new set of individuals, in other words, generating a new population. The generation total represents how many times the process is repeated. In Figure 16, generation total ranges from 50 to 1000 (number of iterations). In the simulated range, we notice that the greater is the total number of generations, the better are the population members. Standard deviation increases when the generation total gets large.

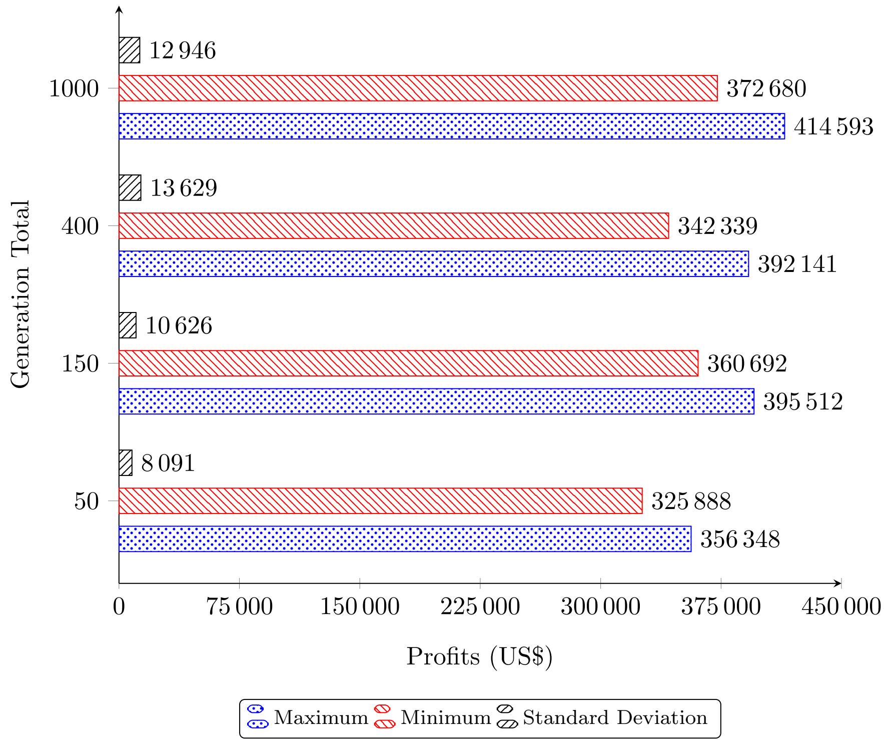

Figure 16: PLGA's parameter set: evaluating generation totals.

Although we have found better results using a large total number of generations, Figure 17 represents the main drawback of increasing generations, which is the average elapsed time. Running 1000 generations is almost 20 times longer than a 50-generation execution on average.

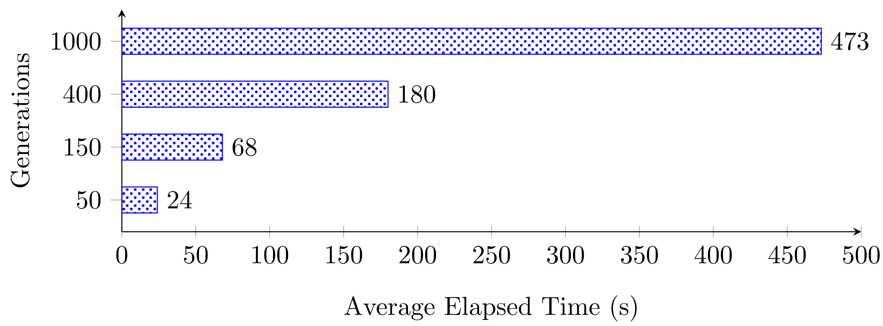

Figure 17: Average elapsed time from distinct generation totals.

Mutation operators introduce small modifications in a population, creating off-springs at random. By doing so, GAs can exploit sparse regions in the search space.

In general, setting relative small mutation rates should avoid convergence loss. But, if the mutation rate were too low, it would cause premature convergence in the population, reaching a local optimum instead of trying to produce global optimum. Evaluating the mutation rate is essential to select the proper criterion, which would avoid weak quality solutions. We have tested the mutation rate from 0.01 to 0.25 and reported the results in Figure 18, the best performance is from setting the mutation rate at 0.25. Standard deviation slightly increases from altering mutation rate, which may suggest more diversity populations from computational tests with large mutation rates.

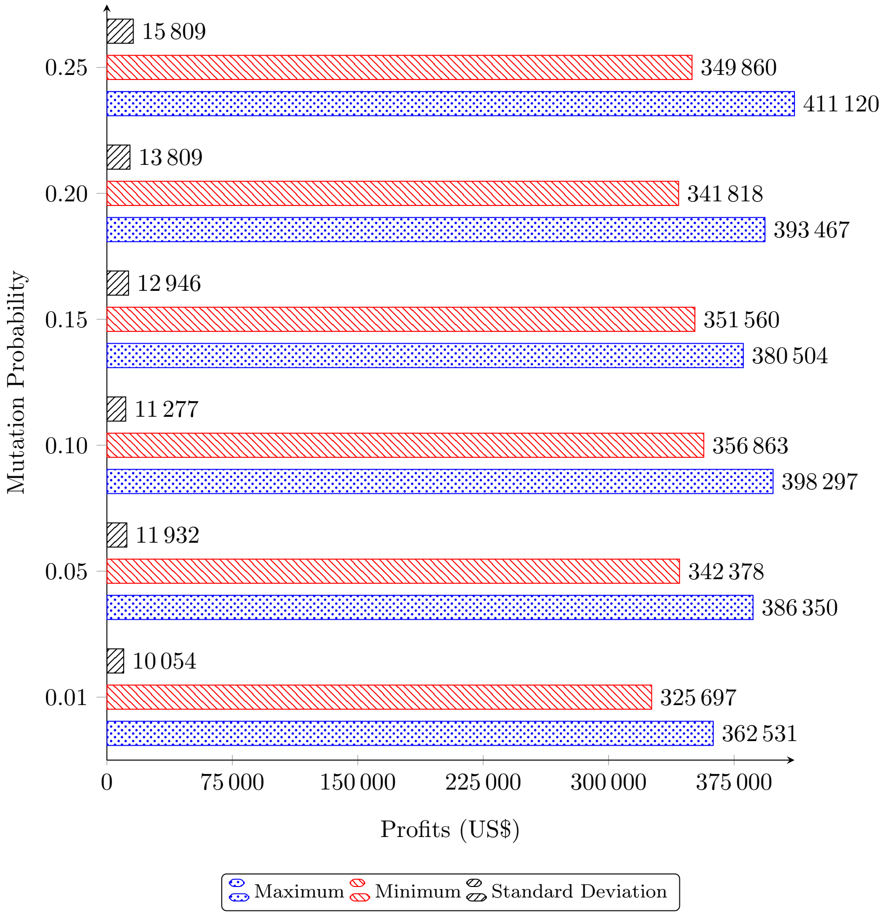

Figure 18: PLGA's parameter set: testing mutation rates.

Fortunately, elapsed time does not profoundly change in the evaluated mutation range. There are just small variations from one batch to another, as shown in Figure 19. Hence, selecting a proper mutation rate would enhance performance without demanding more computational resources.

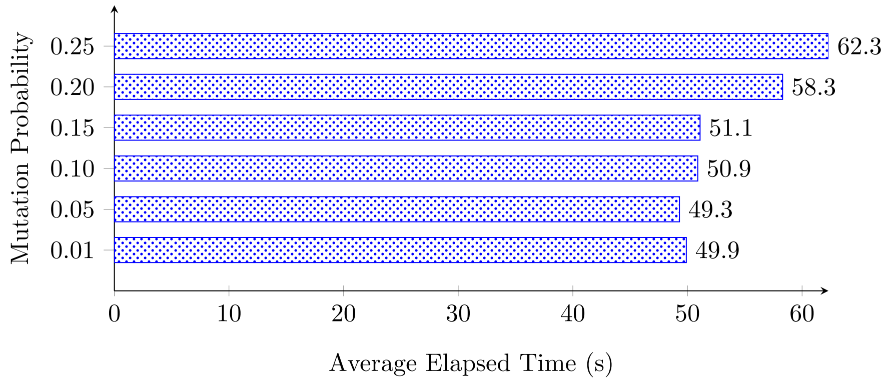

Figure 19: Average elapsed time from distinct mutation rates.

There are three populations in the GA design: (1) a parent population (P_g), (2) a population of new individuals (PN_g) and (3) a mixed population (PM_g). The size of mixed population is equivalent to the sum of P_g and PN_g sizes. In Figure 20, we presented the PLGA performance from 100 to 1600 individuals in the P_g, which is represented by P-XXX; while the size of PN_g is shown as S-XXX. In the tested range, we could verify that the greater is the population size, the best is the maximum fitness in the final population. Improvements from increasing the population size are smaller than setting big generation totals, but large population sizes in the PLGA would hold more different individuals in the population.

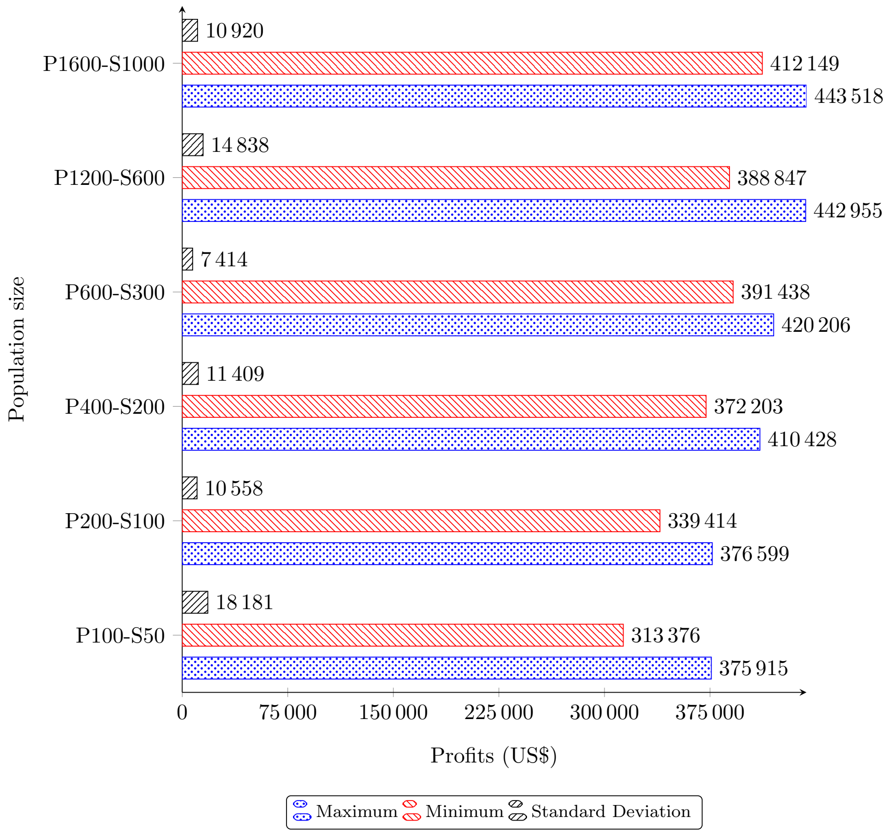

Figure 20: PLGA's parameter set: testing population sizes.

Elapsed time gets large each time population size increases. 21 exhibits the simulated average time from each parameter set in Figure 20.

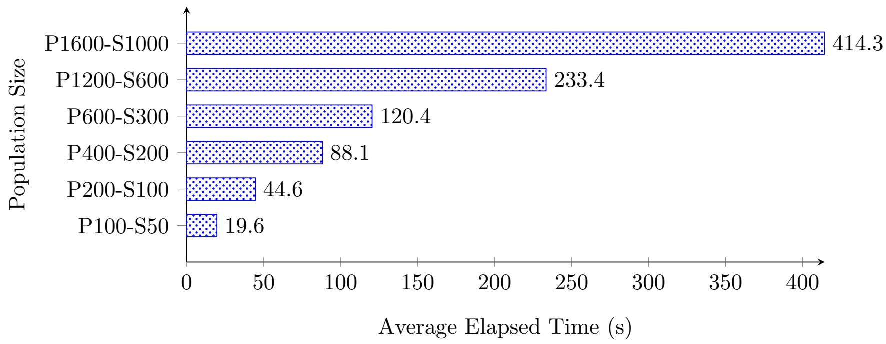

Figure 21: Average elapsed time from each population size.

Figure 22 represents how the population evolves from generation to generation. The best solutions develop fast during the first 100 generations; then, quality growth slows down when it becomes hard to produce better individuals. The plotted data is from a single execution.

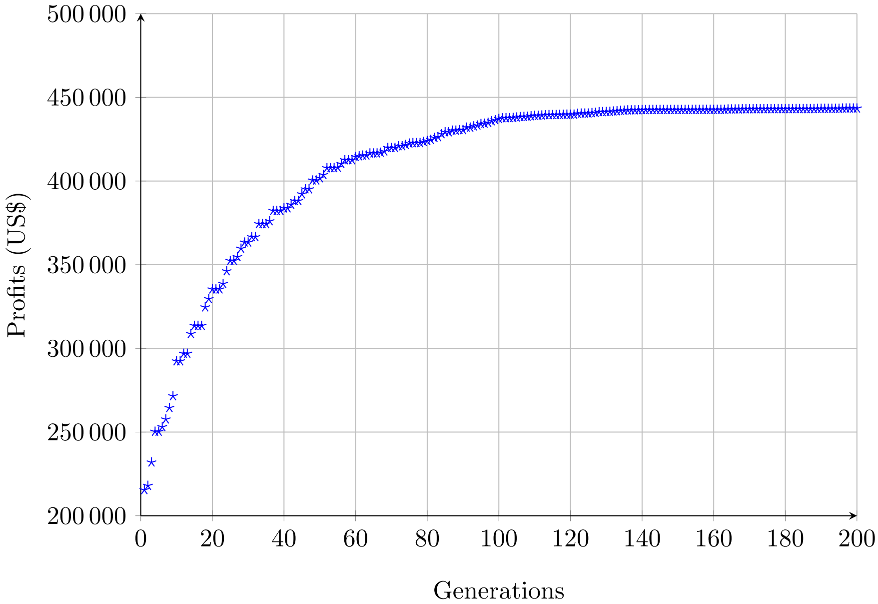

Figure 22: A generation overview in the PLGA.

If the crossover probability is low, the crossover operator would produce almost identical individuals rather than generating new off-springs. From our computational tests in Figure 23, we got the best results when the crossover probability rate is 1.0, which means all PN_g members are new off-springs, generated by the crossover operator. Also, Figure 24 shows that crossover probability barely affects the average elapsed time.

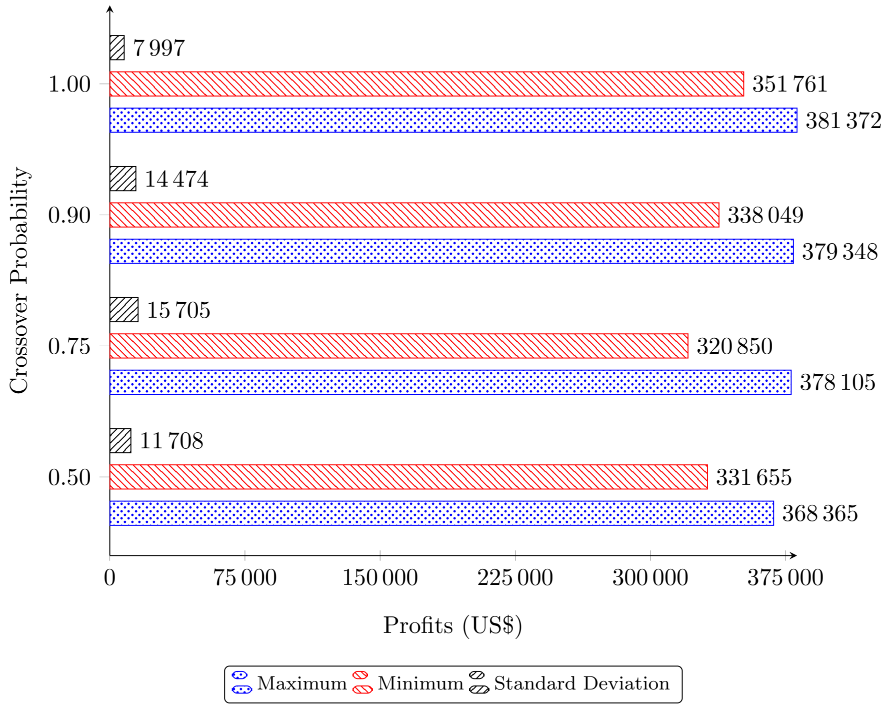

Figure 23: Crossover rate and PLGA's performance.

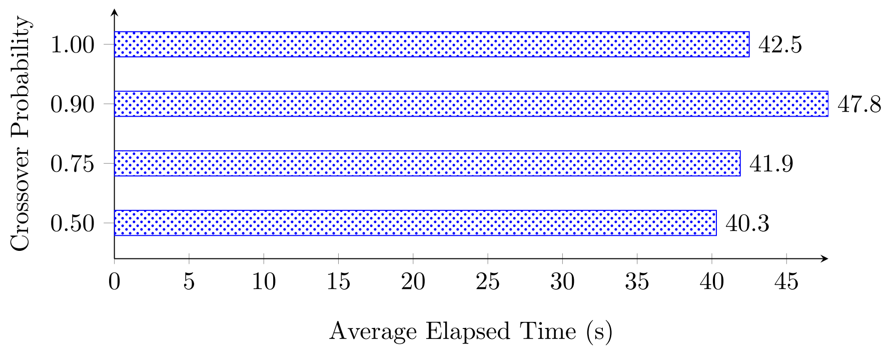

Figure 24: Average elapsed time from distinct crossover rates.

For comparison purposes, we have generated the solutions in Figure 25 using IBM ILOG CPLEX OPL. In these generated solutions, CPLEX was called off when it had reached the time limit, and so, these are not Pareto-optimal solutions, finishing executions before proofing optimal.

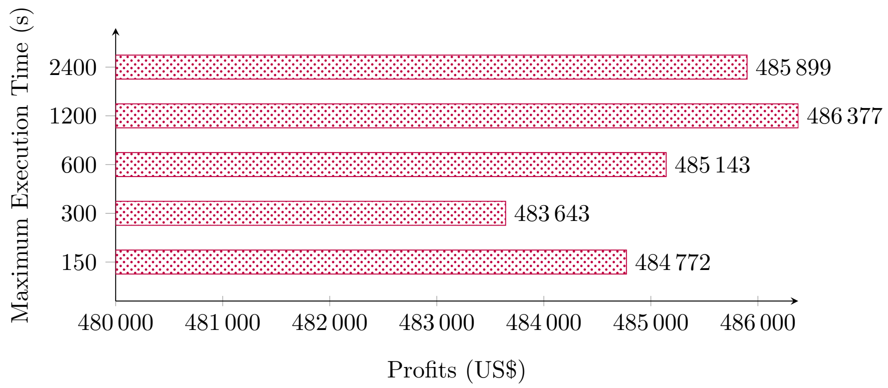

Figure 25: Instance A: generating solutions with IBM ILOG CPLEX OPL.

6.2 Instance B

In these computational tests, we have increasingly difficult using a large set of plots in the crop sequence. Long-term planning in several plots is a great deal hard due to the broad set of adjacency constraints.

The computational tests in this section used the same crop set and parameters in Appendix A. The total number of crops is still 67 ($N = 67$) and the total number of crop's families is 11 ($N_f = 11$). The planning horizon is eight years long (in the PLGA, it is a total of 192 periods [$M = 192$]). The features of the plot set are exhibited in Table 13. Filled cells outside the main diagonal represent that the column plot is adjacent to the row plot.

Table 13: The plot set in Instance B.

Plot—Plot	Plot 1	Plot 2	Plot 3	Plot 4	Plot 5	Plot 6	Plot 7	Plot 8	Plot 9	Plot 10	Plot 11	Plot 12	Plot 13	Plot 14	Area (ac.)
Plot 1															1.00
Plot 2															1.50
Plot 3															0.80
Plot 4															1.25
Plot 5															1.25
Plot 6															1.40
Plot 7															1.60
Plot 8															1.00
Plot 9															1.50
Plot 10															0.80
Plot 11															1.25
Plot 12															1.25
Plot 13															1.4
Plot 14															1.6

Following the way we have proceeded before, our first trial is to verify PLGA's performance over distinct total number of generations. Figure 26 exhibits the results of these computational tests. Our preliminary observation from the previous generation trial holds well-grounded: the greater is the generation total, the better are solutions in the final population.

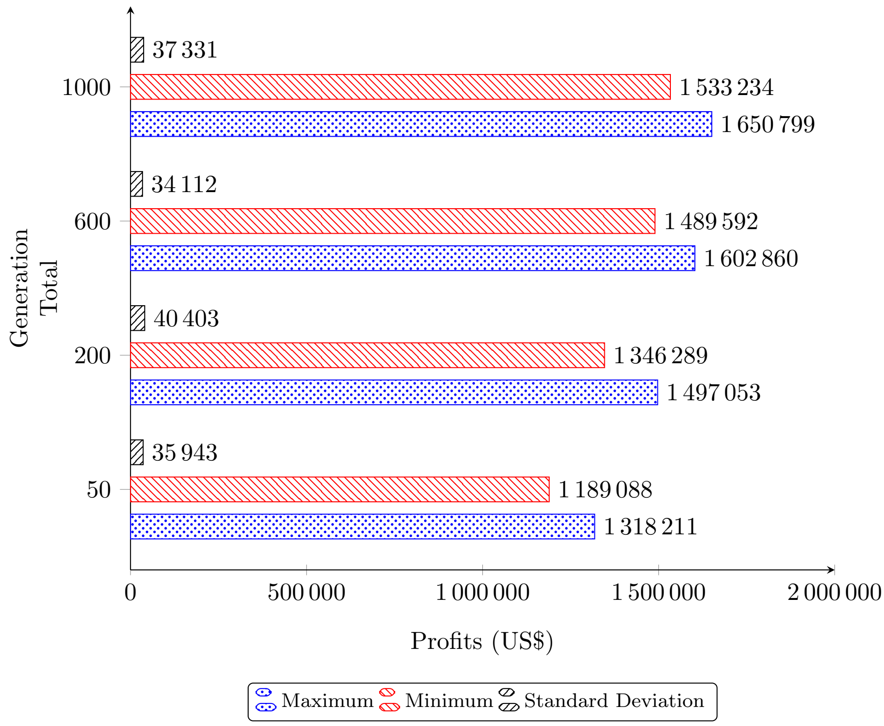

Figure 26: PLGA: testing generation totals in Instance B.

Changing the plot's configuration increases the dimension of the CRP, and computational tests would also take more time. Average elapsed time from the generation trial is displayed in 27.

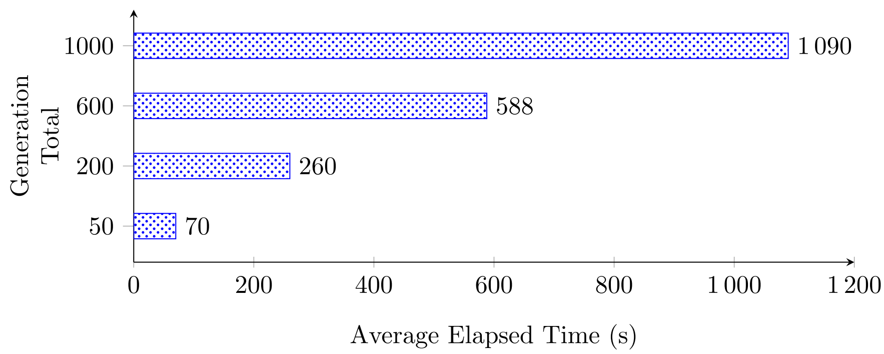

Figure 27: Average elapsed time from testing generation totals.

The results from the mutation rate trial in the PLGA are displayed in Figure 28. The tested range is from 0.05 to 0.25. The best outcome in these computational tests was from a mutation rate of 0.15.

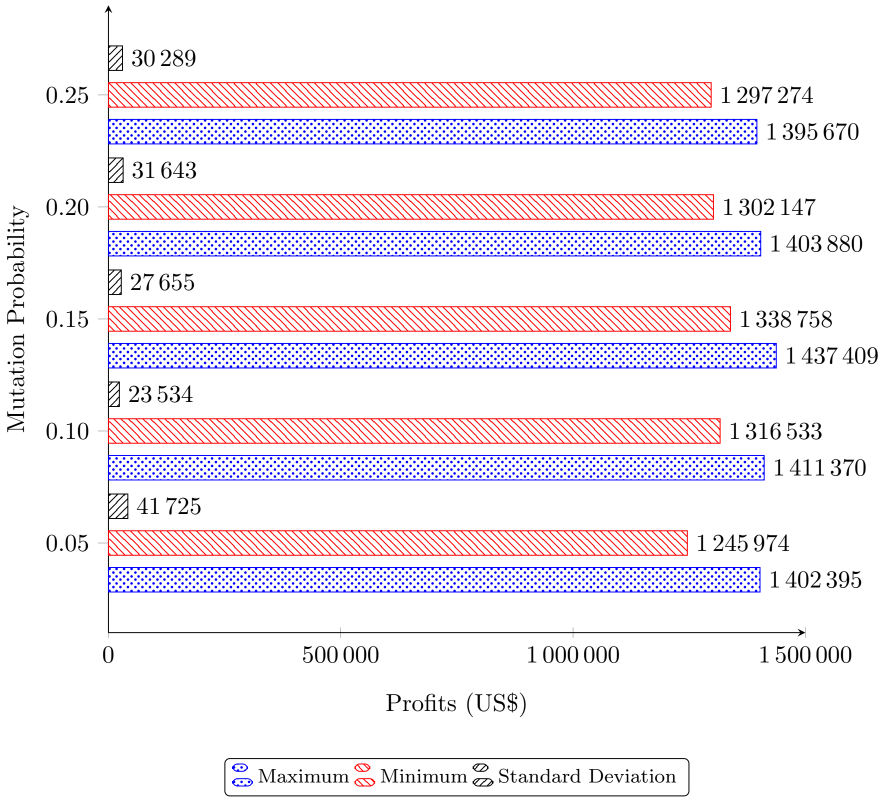

Figure 28: PLGA: evaluating mutation rates in Instance B.

Mutation rates slightly alter average computational time, much less than increasing generation total. Figure 29 displays the average elapsed time from each configuration.

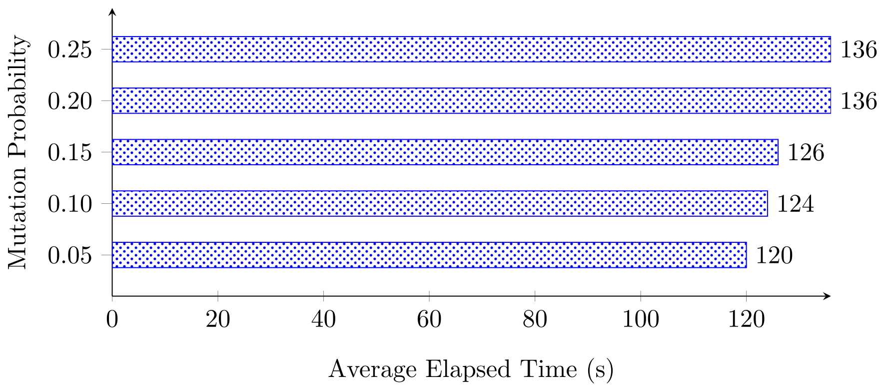

Figure 29: Average elapsed time from trying distinct mutation rates.

Optimization models have different properties, and, indeed, we could not establish default parameters for all the problems. A large number of population size may improve results, but there is no strong relationship between the population size and the best population members. In general, the population size should be large enough to provide a diverse set in the initialization. From 100 to 600 members, we present simulated results in Figure 30. Again, large instances of the CRP would require large populations to generate reasonable solutions.

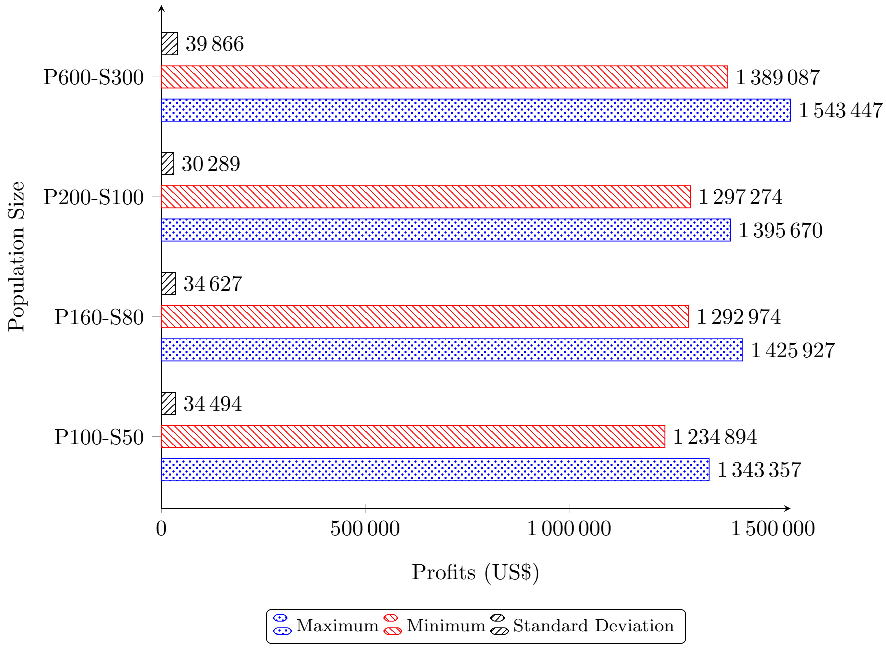

Figure 30: PLGA: testing population sizes in Instance B.

From the population size trial, Figure 31 presents the average computational time of each parameter batch.

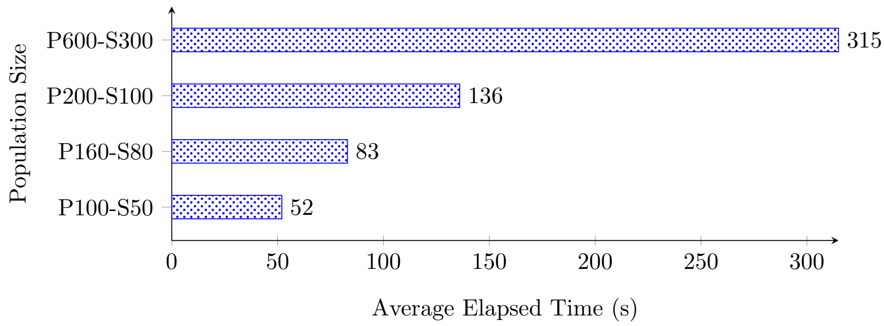

Figure 31: Average elapsed time from testing population sizes.

A comparison of the generated solutions using exact methods and the PLGA solutions could provide another perspective of the CRP. Using the same database and parameters, such as the planning horizon and the plot set configuration in Table 13, we generated the CRP solutions in Figure 32. IBM ILOG CPLEX OPL produced this solution set, and the maximum execution time is also displayed.

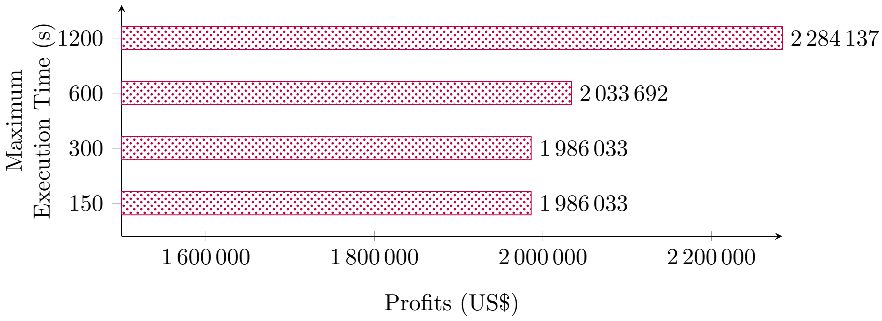

Figure 32: Instance B: generating solution using IBM ILOG CPLEX OPL.

7 CLASSICAL MULTIOBJECTIVE OPTIMIZATION METHODS: EVALUATING RESULTS

In this chapter, we analyzed classical optimization methods we have presented in Chapter 5. They are highly acclaimed methodologies and common topics in optimization literature. We want to evaluate their features in the CRP, recognizing potential advantages and side effects on the proposed multiobjective model of the CRP. Appendix D presents the configuration settings in these computational tests.

7.1 CRP database and fundamental parameters

Computational tests in this chapter are based on the database in Appendix A, which was composed of reliable sources and captured the complexity and diversity of the CRP. Using the IBM ILOG CPLEX OPL, we have implemented the classical optimization techniques in Subsection 5.2. Table 13 presents the plot configuration. The following list summarizes the main parameters:

- Number of Crops: 67 ($N = 67$);
- Total number of Crop's families: 11 ($N_f = 11$);
- Planning horizon: 2 years ($M = 48$);
- Total number of plots: 14 ($L = 14$);

On a computer where parallel threads are available to CPLEX, the automatic setting typically results in the *concurrent optimizer* being called in either deterministic or opportunistic parallel mode. The *concurrent optimizer* launches distinct solvers in multiple threads. We ran the computational tests using a device with a 2-core and 4-thread processor. Hence, CPLEX has called the concurrent optimizer, running parallel tasks, and speeding up optimization. That is a distinct characteristic in comparison with the proposed genetic algorithms in this dissertation because we have not developed multi-threaded programs.

7.2 The Weighted Sum Method

Setting the time limit at 3600 seconds, we performed several computational tests using distinct weight sets. Figure 33 presents the achieved solutions. The status of these solutions produced by CPLEX is 11, which represents that optimization has stopped due to a time limit. Hence, the solution set in Figure 33 has not been proven optimal. Even in these long computational tests, we could not get the problem's efficient points due to the dimension of CRP.

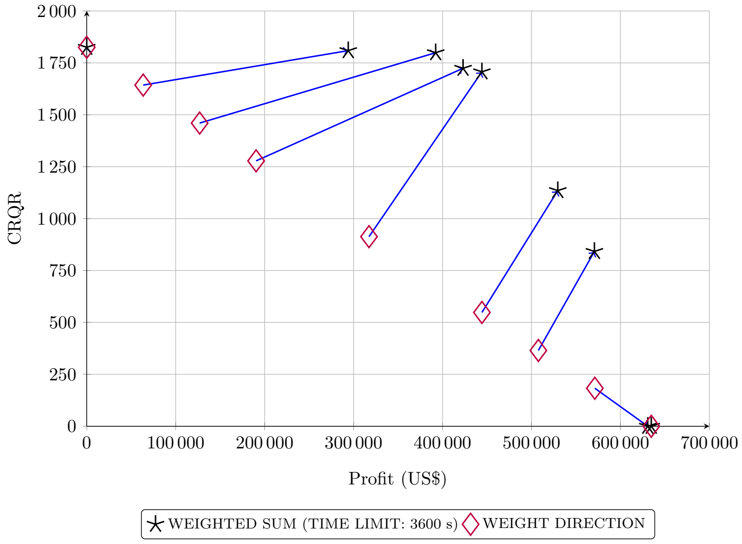

Figure 33: Weighted Sum in the CRP (time limit at 3600s).

We generated the solution set in Figure 34 by reducing the time limit to 150s, which accounts for execution time for all the solver's steps, such as model initialization and internal calls. The objective functions are normalized to get consist of results. Weight direction represents a linear combination of the weight set, which may result in a solution in that orientation. Figure 35 represents the solution set using 300-second time limit. CPLEX also reported status 11 in these generated solutions.

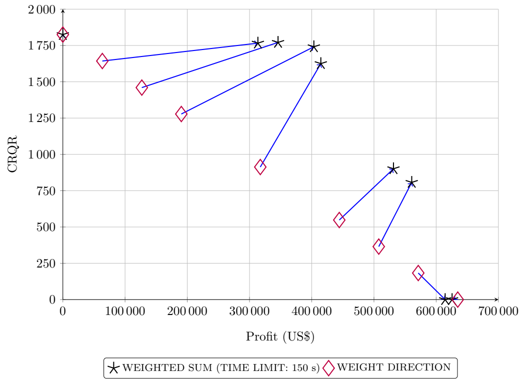

Figure 34: Weighted Sum in the CRP (time limit at 150s).

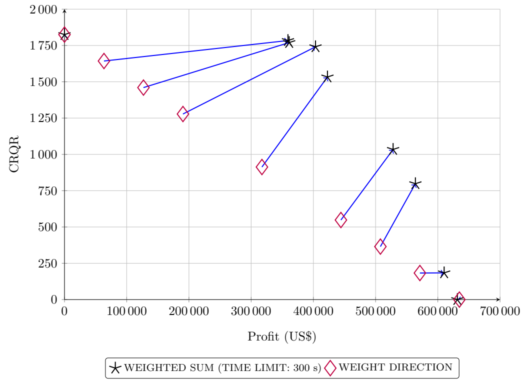

Figure 35: Weighted Sum in the CRP (time limit at 300s).

Figure 36 displays a frontier comparison. We want to check how setting execution time in a more convenient manner could compromise the generated solution. As we could notice in Figure 36, setting a long execution time would slightly enhance CPLEX performance, although frontiers using a time limit of 150 and 300 seconds are very close to the 3600-second frontier. Hence, from now on, we proceed with 150-second computational tests.

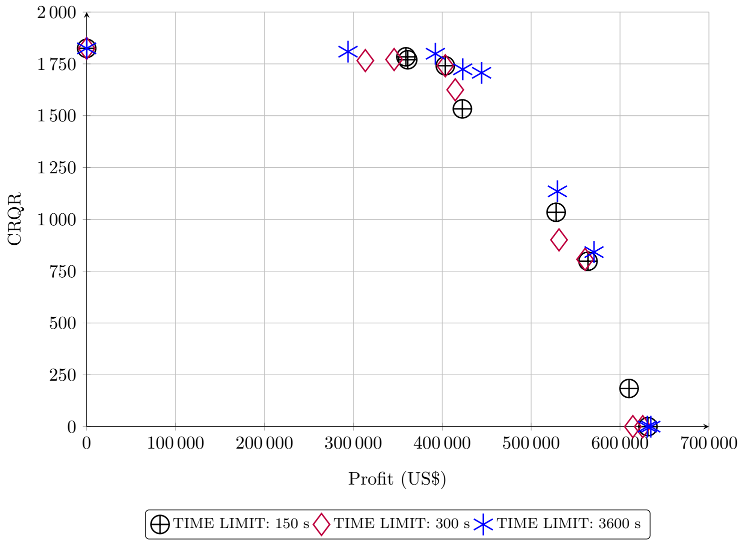

Figure 36: Weighted Sum in the CRP: frontier comparison.

7.3 Goal Programming

Figure 37 expresses the desired conditions of the CRP, which are called *goals* and ought to be met very closely when it is possible. Goals might be specific values or ranges (CHARNES; COOPER, 1977). In these computational tests, goals are represented by specific values.

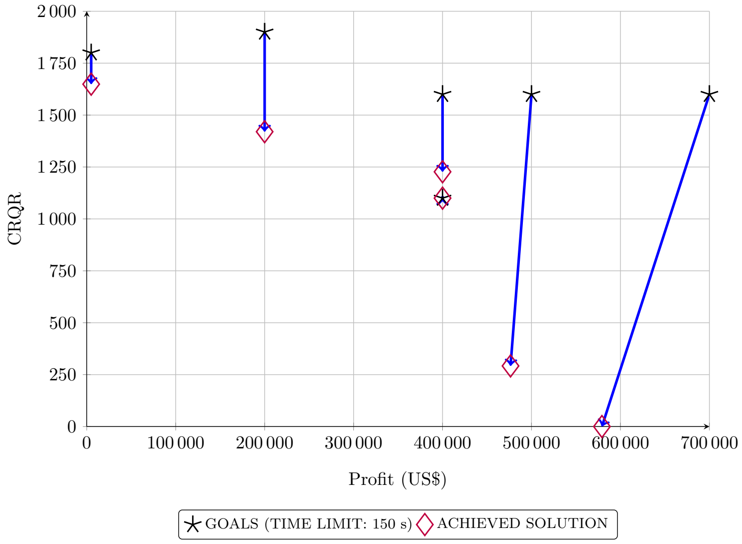

Figure 37: Goal programming in the CRP.

Trade-offs are usual, balancing performance and quality. Figure 38 presents how expensive it is to keep the search for better solutions and how impressive is the improvement. Increasing the slightly time limit returned a better solution as well as jumping to ten times the initial limit. Time comparison is deeply connected to the user's needs. If farmers would like to run a lot of situations to make a decision, finally, time must be critical. But let us suppose that the parameters are well-established, and our goal is digging just one unique solution, we might let our solver running almost indefinitely.

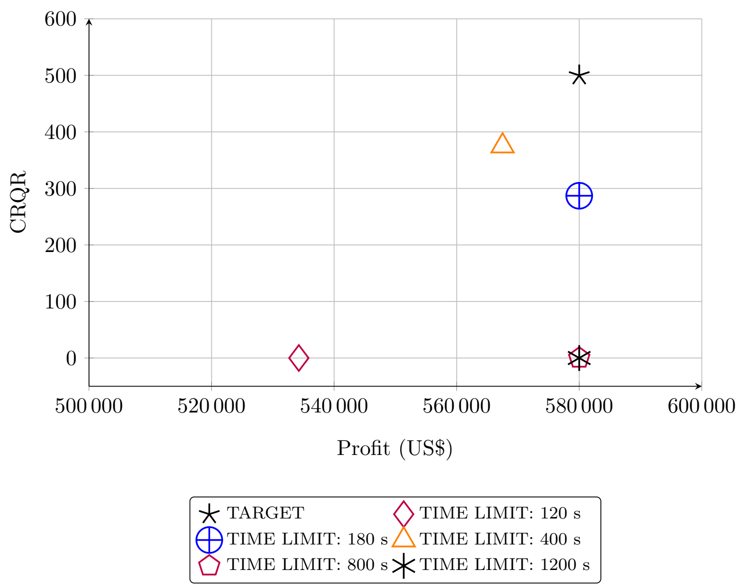

Figure 38: Goal programming in the CRP: a time limit comparison.

7.4 The Achievement Scalarizing Function Method

Figure 39 introduces the results from the Achievement Scalarizing Function Method (ASFM). Taking a close look at target [200000, 1200] and its corresponding solution, we noticed that even setting a relatively easy goal; we got a surplus solution (which means that it is a solution better than its target). Hence, targets in the achievement scalarizing function method are directions of search, guiding the optimization into specific regions from the combined objectives. Providing directions instead of closed targets also ensure that the optimization process would keep digging toward optimal. Considering targets as directions, we also observe that all the pairs (target and solution) are aligned.

One of the drawbacks is the ASFM initialization. Users must provide y_{nadir} and y_{ideal}, which shows in general that mono-objectives optimizations should take place before proceeding to the algorithm itself. For instance, let us consider a two-objective space, y_{ideal} is the combination of the best results from optimizing each objective individually, and y_{nadir} is the combination of worst results gathered from the previous pair of solutions. Any change on the database would require an update of y_{nadir} and y_{ideal}. Back to the CRP and thinking about crop demand, if farmers decide to set a few product demands and watch the outcomes many times, they will be very busy with the setup of the algorithm.

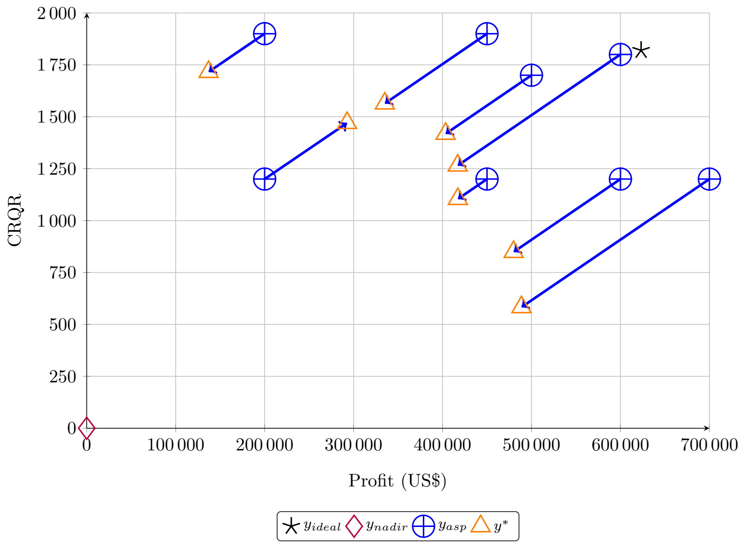

Figure 39: ASFM in the CRP.

7.5 Optimization based on boundaries

Figure 40 shows several computational tests using the optimization based on boundaries procedure. In Figure 40, we can see the generated solution by CPLEX (y^*), which is connected to the boundaries: y_{min} and y_{max}. Regardless selection of the boundaries produces an unfeasible problem, as we had found out when the lower boundary (y_{min}) was set at [550000, 400].

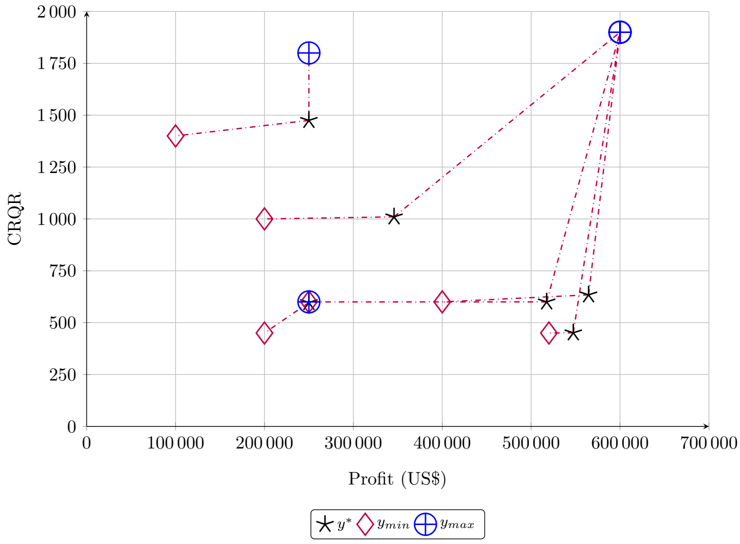

Figure 40: Optimization based on boundaries.

7.6 ϵ-Constraint Method

Setting minimum returns or reaching at least some level of rotation quality might be very appealing. Results from computational tests with the ϵ-constraint method are exhibited in Figure 41. There are two major groups in these computational tests: (1) minimum profitability and (2) minimum crop rotation quality rating. There are two limits without finding a solution moving along the x-axis. And along the y-axis also misses one solution for one feasible limit (it does not mean there are no solutions. It just indicates that the solver could not provide a solution without triggering time limit). Setting tight bounds would turn the optimization problem unfeasible; it fails to generate solutions in [0,1600] and [600000,0].

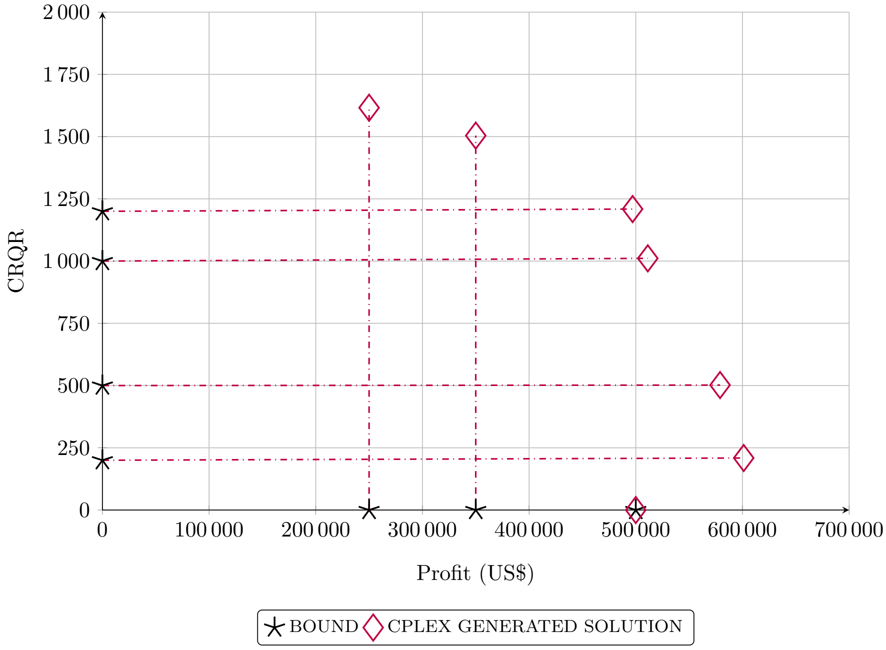

Figure 41: ϵ-Constraint Method in the CRP.

7.7 Other considerations

Describing all the crop sequences we have presented results in this chapter would require many additional pages. Let us move directly to the final statements and downsizing this work right to the essential content. The first point is the conflict between objective functions: profit maximization and CRQR maximization. If it is still unclear why they are conflicting, let us discuss something more practical.

Cash crops are the most profitable crops. The highest yields are the primary goal when these crops are scheduled in the crop sequence. But, as harvest removes the production from the field, it could deplete soil fertility. Then, we did not consider them soil builders, and CRQR does not account for any effects in the quality index when cash crops are scheduled. Of course, cultural practices could change this condition, and some results from the cash crop seeding may enhance several soil properties, but we have not followed this path in this dissertation. In Appendix A, parameters related to the quality rating skipped the cash crops.

Cover crops have entirely different management. Harvesting is not the objective when they are scheduled. Rather than profit, we look forward to strength soil characteristics and reduce pests and weeds. In general, farmers plow down and kill cover crops when they produce more benefits. In the same cases, that is before the cover crops reach the reproductive stage, and so, they may not produce any fruit. Profits from cover crops are usually small or, sometimes, they generate just costs.

Once we have understood the role of cash and cover crops, we would have a better idea about what looks like the highest profitable crop sequence and the highest CRQR solution. The first one would have mainly cash crops, while the second one would consist of cover crops only.

8 PLMGA PERFORMANCE ANALYSIS

Performance analysis of a multiobjective genetic algorithm is harder than evaluating mono-objective GA's performance. We have to account for many other parameters besides comparisons with the problem's efficient frontier and cost evaluation (computational time). We introduce some evaluation parameters in Section 8.1, which have helped us to check the PLMGA's performance. The following sections present results from distinct scenarios. In addition, Appendix D presents the system settings from these computational tests.

8.1 Performance Parameters: Multiobjective Genetic Algorithms

We can evaluate performance of multiobjective genetic algorithms using the following criteria: (1) Diversity: it gives a broad measure of the different non-dominated solutions composing the trade-off surface; and (2) Closeness: this metric formulates the nearness (or distance) of each non-dominated solution to the nearest point belonging to the true (or global) optimal Pareto front denoted by P^*. We can use average distances to check the accuracy of a single multiobjective genetic algorithm (JEDIDI; CAMINADA; FINKE, 2004).

As with mono-objective optimization, the decision-maker might be concerned about how long it takes to evaluate feasibility. Therefore, multiobjective genetic algorithm performance must consider the required computational time, which should be competitive with other multiobjective methods. Schott (1995) presented several concrete measures for evaluating performance and comparing configuration parameters. We have selected some of them and described ahead:

8.1.1 Measuring efficient points

This performance indicator represents the number of efficient points in the method's final solution set. If we do not know the problem's efficient set, we cannot make any reliable conclusions about the position quality of the method's efficient set. But, if the method generates points that are mainly dominated solutions, they do not belong to the problem's efficient set by definition. Then, measuring a reasonable number of non-dominated solutions indicates that the method produces a representation of the problem's efficient set with significant resolution.

8.1.2 Determining efficient set spacing

This metric is the variance of the range (distance) of each solution to its closest neighbor. Just solutions from the method's efficient set should account in this measure. It will return null if the method's efficient set has members equally spaced from each other. In general, optimization methods are efficient in finding E when they minimize $f_{spacing}$.

- $f_{spacing}$: efficient set spacing;
- e: the method's efficient set (number of members);
- $\bar{d}$: average distance;
- ${J_1}^i$: first objective function of i solution;
- ${J_1}^j$: first objective function of j solution;
- ${J_2}^i$: second objective function of i solution;
- ${J_2}^j$: second objective function of j solution.

$$f_{spacing} = s^2 = \frac{1}{e-1} \sum_{i=1}^{e} \bar{d} - {d_i}^2 \tag{8.1}$$

$$where \quad d_i = min_j \quad |{J_1}^i - {J_1}^j| + |{J_2}^i - {J_2}^j| \tag{8.2}$$

8.1.3 Clones in the population

According to Schott (1995), the robustness of the GA depends heavily on the ability of reproduction, crossover, and mutation to balance diversity (exploration) and efficacy (efficiency). Clones, which are duplicate strings within the population, could severely reduce exploration and efficiency in the multiobjective genetic algorithm. The existence of clones limits diversity and hinders available resolution points of the efficient set. Assuming a population of Y members and W the current number of clones in the population, for example, a multiobjective genetic algorithm without clone prevention can create Y - W point resolution of the problem's efficient set.

8.2 Scenario 1: validating the algorithm

Scenario 1 is a low complexity instance of the CRP. Using the crop database in Appendix C, there are 23 available crops in this scenario. The cropping sequence should cover a two-year interval (48 periods). The plot's configuration is a minimum set, described in Table 14, with two adjacent plots and equivalent cultivable areas. Although there are few crops in Scenario 1, the total of crop's families remains the same. There is no family left behind; they are just in small quantities.

Table 14: Plot's characteristics at Scenario 1: adjacency and cultivable area.

Plot—Plot	Plot 1	Plot 2	Area (ac.)
Plot 1			1.00
Plot 2			1.00

Simplicity is the main reason we have chosen Scenario 1. Due to the low complexity, we can produce some problem's efficient points using classical approaches presented in Section 5.2. Comparison with the problem's efficient frontier is a good reference, but, in general, the efficient set is not available. Hence, running a small instance first is a good choice. The following list resumes parameters in this scenario:

- Number of Crops: 23 ($N = 23$);
- Total number of Crop's families: 11 ($N_f = 11$);
- Planning horizon: 2 years ($M = 48$);
- Mutation Probability Rate: 0.25;
- Crossover Probability Rate: 1.00.

We selected mutation and crossover probabilities based on the results from Chapter 6. We are going to test a range of the total number of generations and the population size.

Figure 42 exhibits efficient frontiers. The problem's efficient points are gathered in the legend entry CPLEX; all the others are PLMGA's efficient ones. There are two population sizes: (1) 400 members and producing 200 each generation from 100 to 1000 generations, and (2) 100 members, with 50 new individuals, in the 5000-generation case. The resolution of the PLMGA's efficient frontier is well-proportioned, leaving few underpopulated regions. Even the smallest generation total provided a reasonable frontier.

Figure 42 represents a more general perspective of PLMGA's performance. Let us skip any definitive statements about the proper total generation.

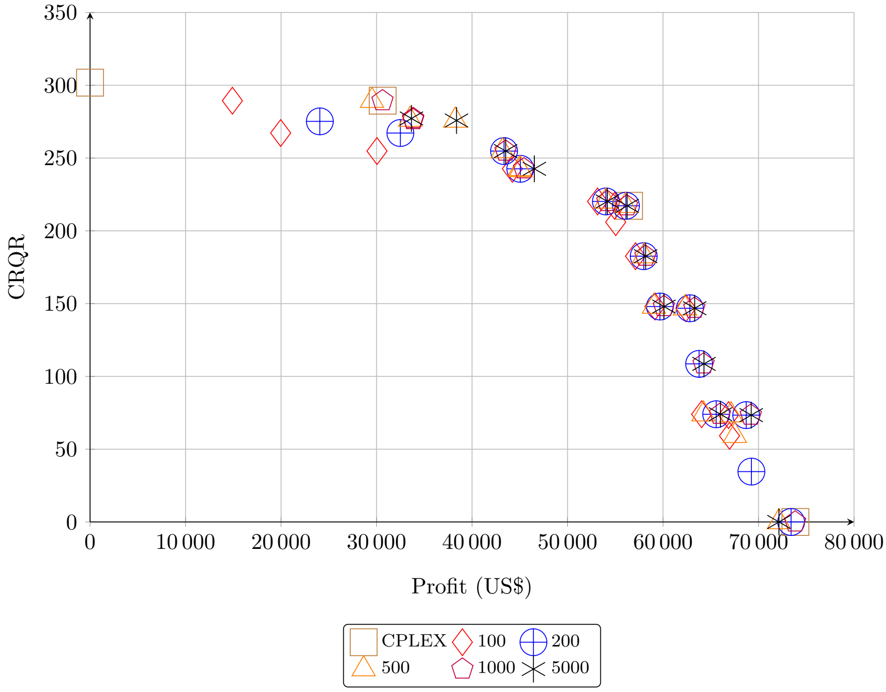

Figure 42: Comparison among efficient frontiers.

As search operators in the GA work with random numbers, performance could vary a great deal from executions. Our trials based on repeating multiple times each parameter set. In Figure 43, we ran each generation total 10 times.

Good resolution in the generated frontier is essential in multiobjective optimization. An efficient method should provide as many efficient points as possible. From Scenario 1, Figure 43 presents the total of generated efficient points.

Even though the population has 400 members (exception is the 5000-generation case with 100 members), we reached 17 non-dominated in the final population at maximum. Although it is a small percentage of the population, we manage to produce just four problem's efficient points using the Weighted Sum Method. Increasing the generation total does not change the total of efficient points expressively. The worst computational test produced just nine efficient points. Standard deviation exceeds 2.00 in only one case and remains low on average.

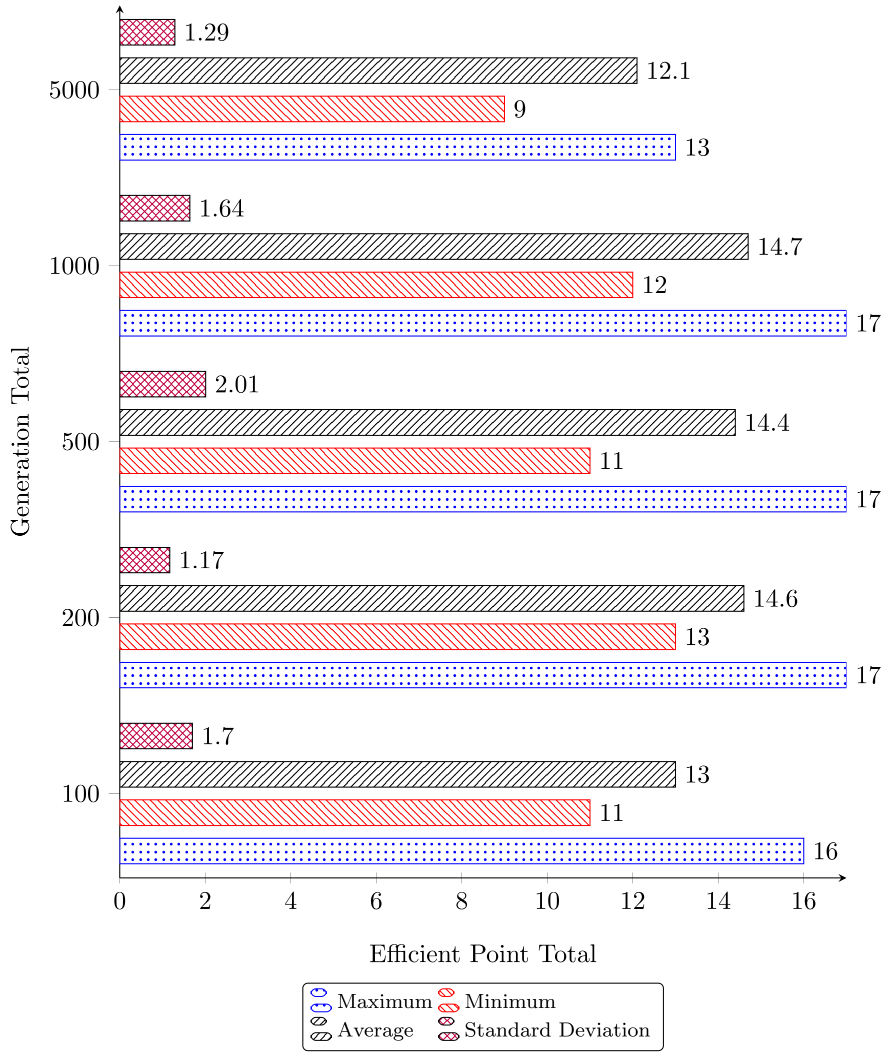

Figure 43: PLMGA: testing generation totals.

The efficient set spacing from the trial parameters in Figure 44. We reported the smallest average and the minimum efficient set spacing in the highest generation totals (1000 and 5000). The largest deviations are from 100 and 200 generations.

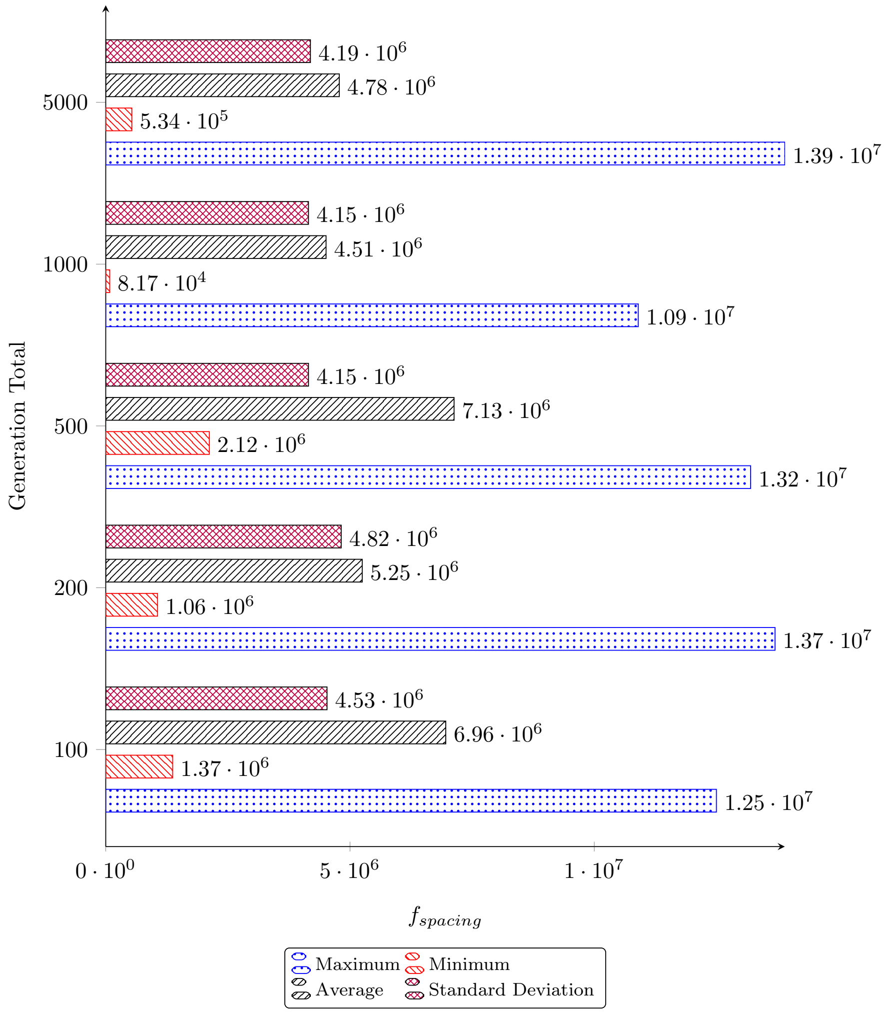

Figure 44: Efficient set spacing: Scenario 1.

Computation time is proportional to the generation total. We present the average elapsed time from each execution parameter in Figure 45. It increases dramatically at 1000 generations. Computation time at 5000 generations is shorter than 1000 generations due to the size of the population.

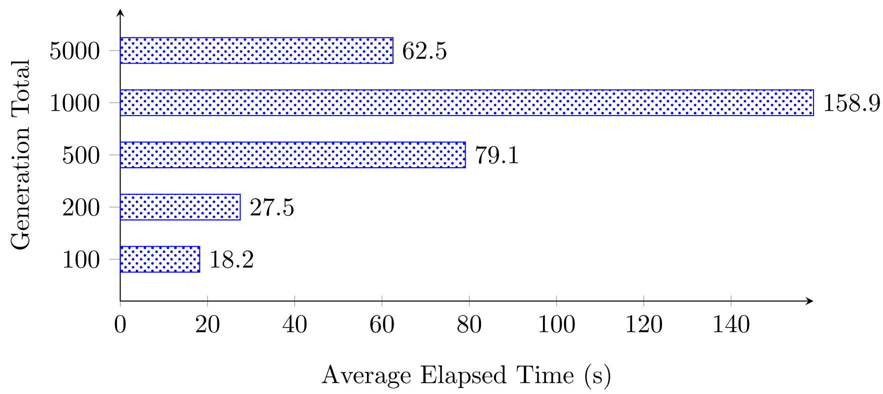

Figure 45: Average elapsed time from each generation total.

Clones reduce diversity in the population significantly. In the PLMGA, we remove clones from the population and create new individuals in their places. Looking at Figure 46, the maximum presence of clones was ten members in this execution. Considering that the population size is 400 combined with 200 new individuals each generation, clones represent less than 2% at the maximum rate.

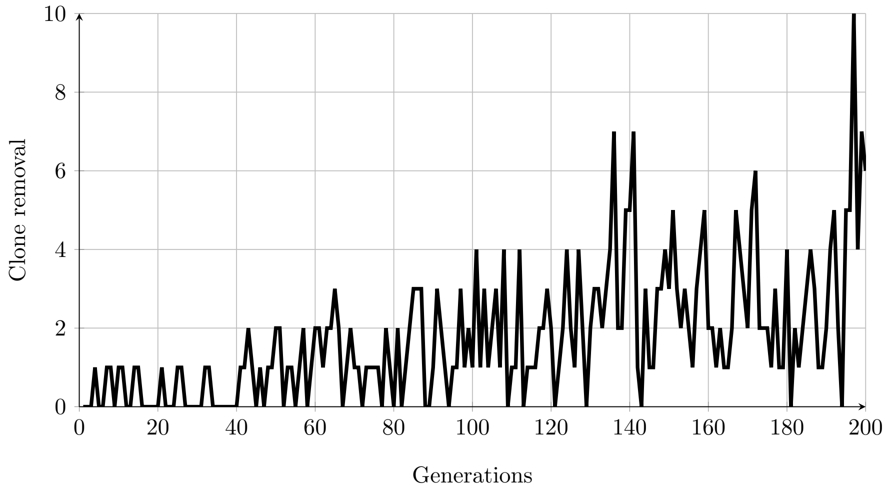

Figure 46: Clone removal from each generation (population size: 400 members).

8.3 Scenario 2: a long-term planning

Scenario 2 is a regular instance of CRP. We are using the full set of crops and long-term planning. The plot's distribution and cultivable are described in Table 12. Mutation probability and crossover keep unchanged in these computational tests. Appendix A presents the crop set features. The total number of crops in this database is 67 ($N = 67$) and the total number of crop's families is 11 ($N_f = 11$). In the list ahead, we summarize the trial parameters in these scenarios.

- Number of Crops: 67 ($N = 67$);
- Total number of Crop's families: 11 ($N_f = 11$);
- Total number of Plots: 3 ($L = 3$);
- Planning horizon: 8 years ($M = 192$).
- Mutation Probability Rate: 0.25;
- Crossover Probability: 1.00;

Figure 47 exhibits the total number of clones in the population from each generation. Despite the total generation, clones remain low in this scenario.

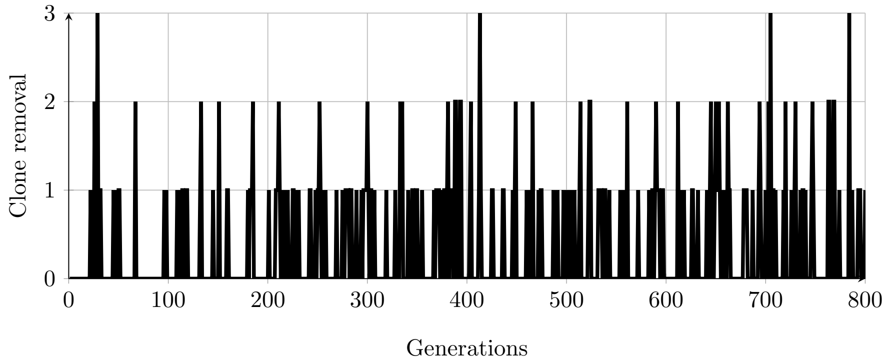

Figure 47: Clone removal from each generation (population size: 400 members).

Figure 48 is an overview of the parameter range. Increasing generations tends to improve resolution (number of efficient points) and the quality of the solutions. We could say that 400-generation and 800-generation fit better than the other parameters in this computational test.

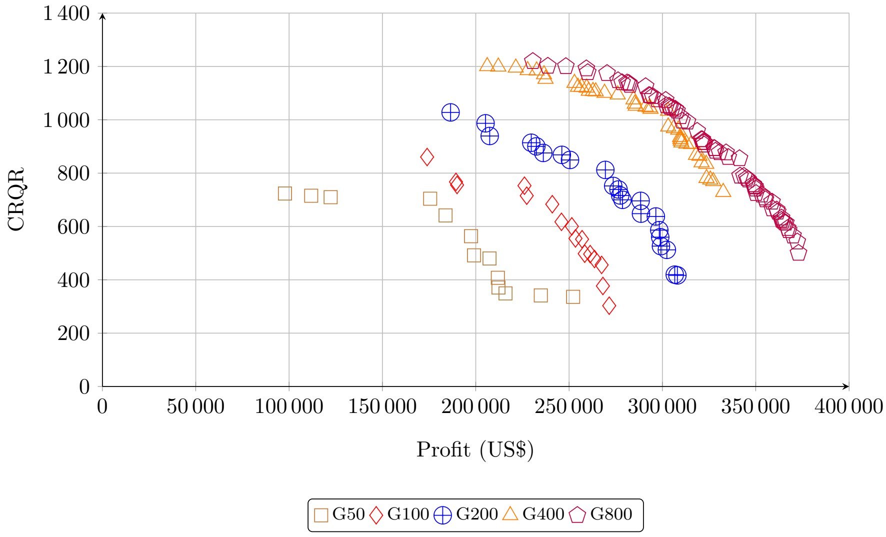

Figure 48: PLMGA: combining efficient frontiers using 400 generation total.

Although the efficient set spacing analyzes dispersion among multiple executions, it is quite hard to picture a clear view in mind. Figure 49 shows a 5-execution combination using the same parameters, which are the generation total of 400 and population size of 200 members. There are more efficient points, and their distribution is a great deal better than Figure 50, where computational tests ran a generation total of 100. Efficient points from Figure 49 are also in the upper region. Even with a smaller generation total, the efficient frontiers in Figure 50 are not far away from each other, which indicates good repeatability in the PLMGA.

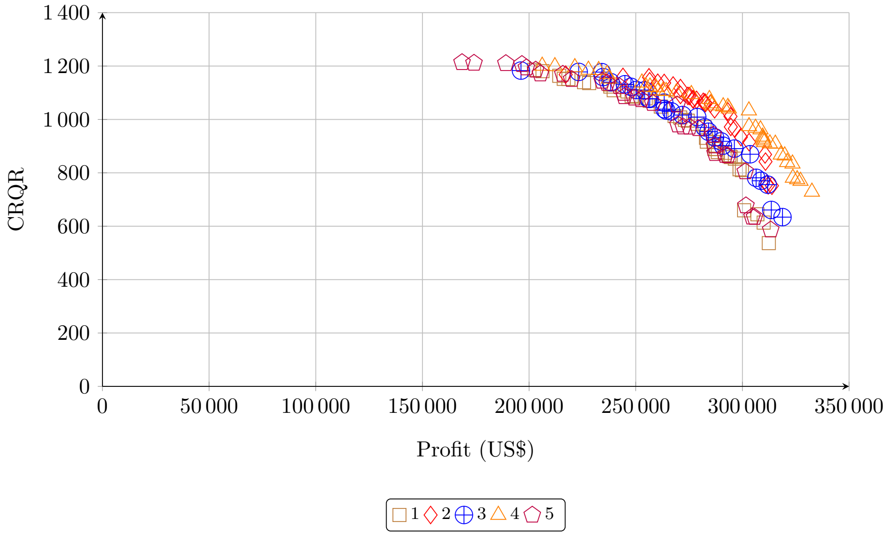

Figure 49: PLMGA: combining efficient frontiers using 100 generation total.

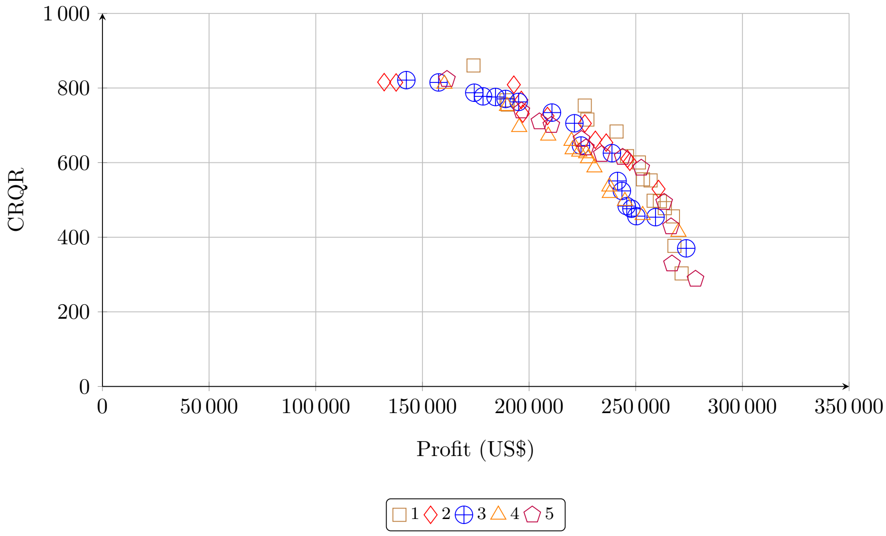

Figure 50: PLMGA: a comparison among PLMGA efficient frontiers.

We present the amount of gathered efficient points in Figure 51. The best performance among these parameters batch is from the maximum generation total. Overall, the number of efficient points only goes higher, moving from the smallest to the largest generation total.

Figure 52 exhibits the efficient set spacing of Scenario 2. As we might expect, computational test with 800 generation total reported the minimum efficient set spacing. The highest $f_{spacing}$ is from the computational test with 50 generation total.

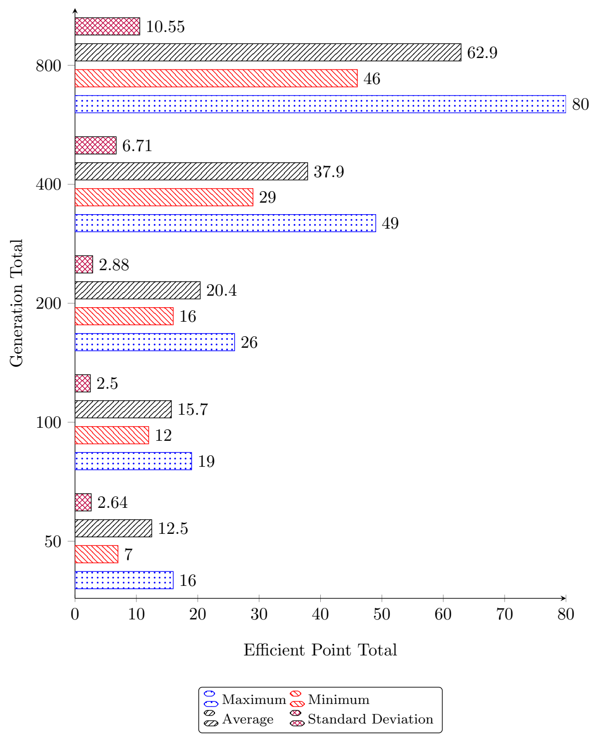

Figure 51: The total number of efficient points in Scenario 2.

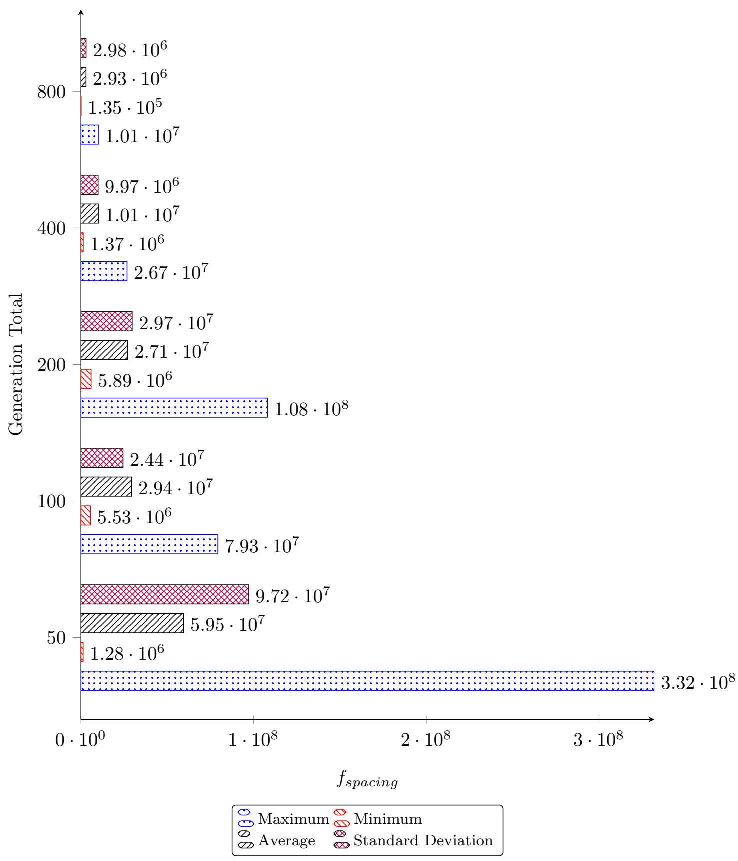

Figure 52: Efficient set spacing in Scenario 2.

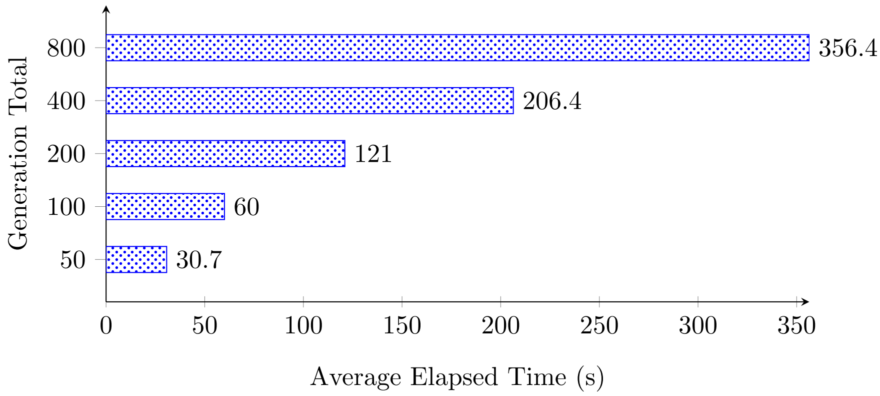

Figure 53: Average elapsed time from each generation total.

computational tests with 800 generation total outperformed all the other parameters until this point, but average elapsed time is a side-effect. As we can see in Figure 53, the average elapsed time from 800 generation total is 356.4 s, while running computational tests with 50 generation total are just 30.7.

8.4 Scenario 3: complexity in several plots

Several computational tests using the full set of crops in Appendix A are presented in Scenario 3. The plot's distribution and cultivable are the same described in Table 13. Mutation probability and crossover rates keep unchanged. A brief description of the parameters follows ahead:

- Number of Crops: 67 ($N = 67$);
- Total number of Crop's Families: 11 ($N_f = 11$);
- Total number of Plots: 14 ($L = 14$);
- Planning horizon: 2 years ($M = 48$);
- Mutation Probability: 0.25;
- Crossover Probability: 1.00.

Multiple executions compose the plot in Figure 54. The solution group with legend entry *ACH* is generated by the Achievement Scalarazing Function Method using IBM ILOG CPLEX. All the efficient frontiers in Figure 54 are computational tests of PLMGA. The prefix G indicates the generation total and P, the population size (parent population, son population is half the size of the parent population, and mixed is the sum of both population sizes).

PLMGA's frontiers are quite optimistic in Figure 54. Although there is none extreme point generated by PLMGA, frontiers are rich in efficient points and well-positioned in comparison with the deterministic method.

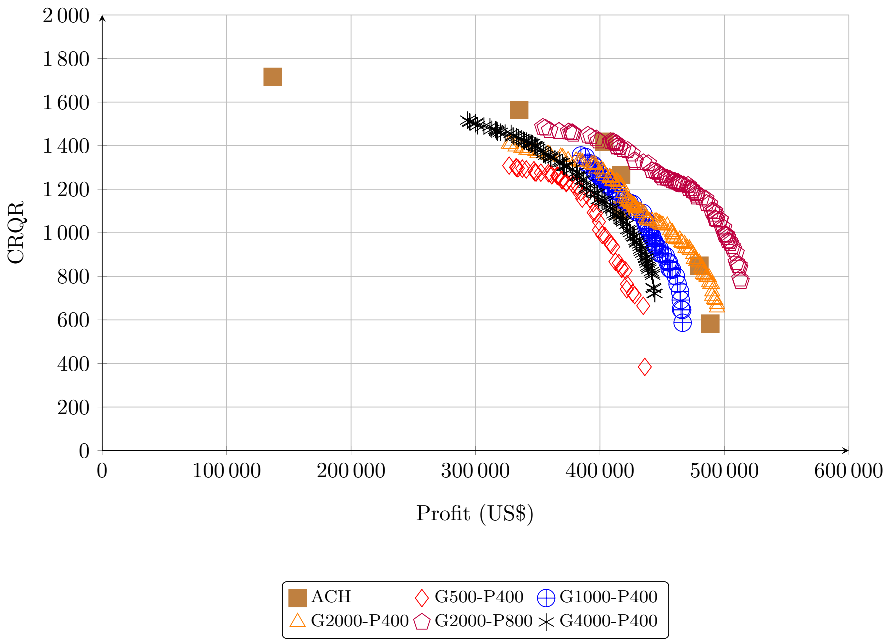

Figure 54: A comparison of efficient frontiers: CPLEX and PLMGA.

Figure 55 shows how many efficient points are generated on average using the selected generation total and population size. G4000-P400 and G2000-P800 are almost tied, leaving all the other parameters behind. Even G500-P400 produced a high-resolution method's efficient frontier with 39 points on average. PLMGA can generate many alternatives, enhancing the perspective of the DM. The standard deviation at G4000-P400 is lower than G500-P400 if we consider the average number of efficient points is more than twice as much as G500-P400.

Figure 56 represents the efficient set spacing in these computational tests. Computational test with G2000-P400 scored the best performance, the smallest efficient set spacing on average.

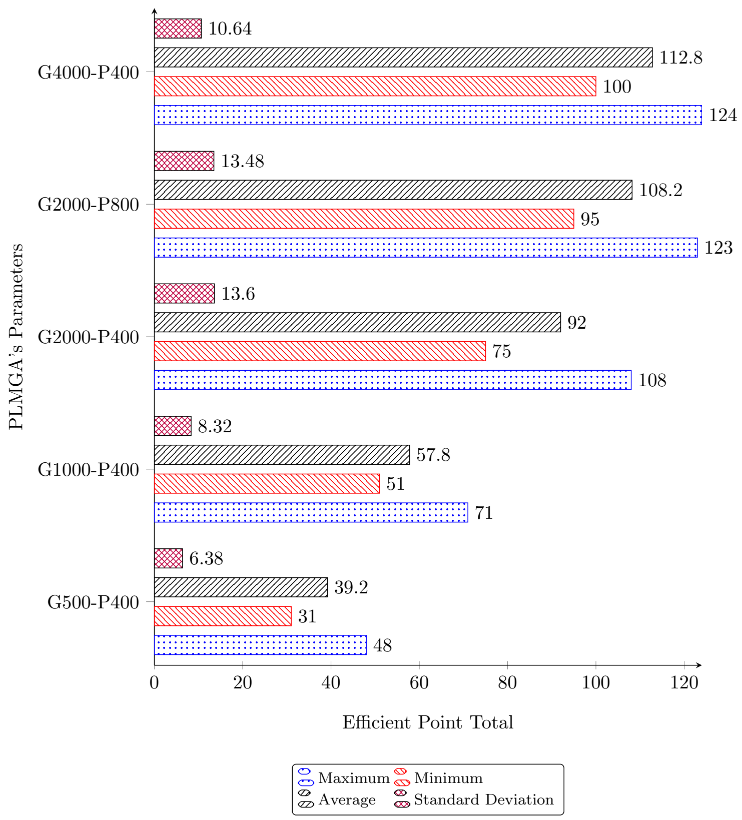

Figure 55: PLMGA: testing generation totals in Scenario 3.

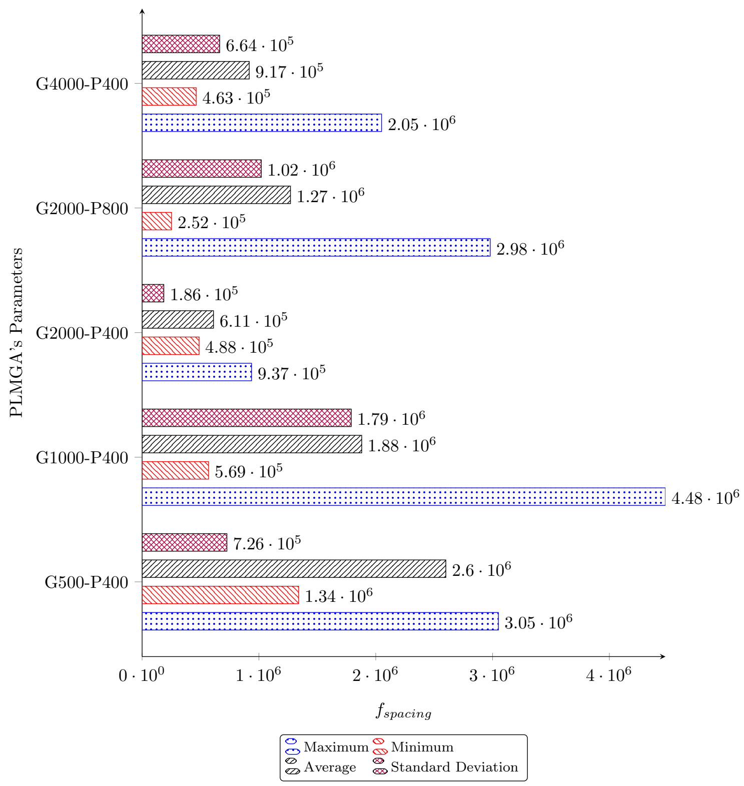

Figure 56: Efficient set spacing: Scenario 3.

Although we would like to increase generations and population size as much as possible to get better results with PLMGA, we have to be cautious and consider the growth of computational time required from each parameter configuration. Figure 57 presents average elapsed time from the executions in this section.

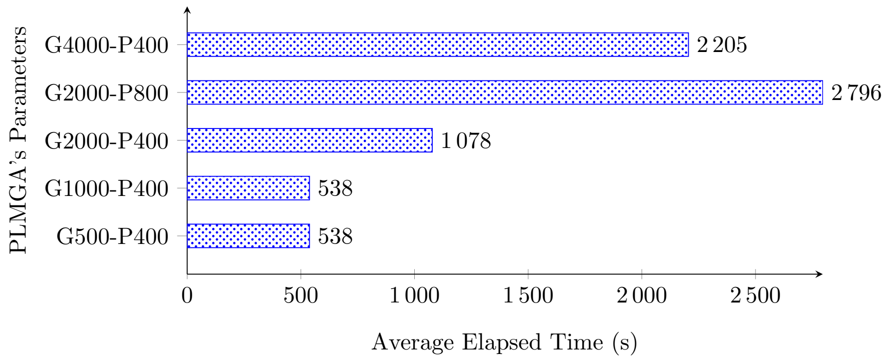

Figure 57: Average elapsed time from each generation total in Scenario 3.

8.5 Chapter's considerations

Several computational tests have been discussed in this chapter. Evaluating multiobjective optimization algorithms is a non-trivial task, and so, we discussed suitable performance indicators in Section 8.1. Unlike mono-objective optimization problems where a single optimum solution would be more interesting, generating a complete solution set is highly valuable in multiobjective applications. Hence, how many efficient points and how well distributed they are in the solution set are significant information. As we have noticed in the previous results, PLMGA can generate a diverse solution set in a single execution, while we have to run the deterministic methods several times and changing parameters every test to compile a solution frontier.

Concerning the practice appeal of the CRP, experience farmers usually have their targets well defined, their growing crops are carefully selected and we might expect fewer crops in the database, which eventually reduces the problem's dimension. Hence, experience farmers might find the classic optimization method more interesting. In the other direction, PLMGA would more appropriate to newcomer farmers because creating an overview of the problem would be much faster using PLMGA.

9 Conclusions and Final Considerations

The Crop Rotation Problem (CRP) is a highly acclaimed agrarian problem, which we have explored and presented many contributions in this dissertation. By modeling the relationship between soil nutrient availability and cropping nutrient demand, we have proposed an innovative mono-objective optimization model of the CRP without compromising any sustainable characteristic of the crop rotation technique. Even with profit from the crop sequence remains as the primary concern in the proposed model, considering fertilization amendments into the objective function reveals a straight pathway to observe the amount of external resources required in the crop sequence and lead to more conscious management decisions.

In the proposed multiobjective model of the CRP, we have analyzed many attributes from cover crops and their benefits on soil management. We have developed a novel crop rotation quality rating. Beyond trying to reach a nutrient balance in the crop sequence, we would also like to enhance soil properties using the multiobjective formulation. Conflicting objectives are usual in many optimization problems, and the multiobjective CRP represents one of them. Increasing profits would reduce cover crop scheduling, and so, profitability and sustainable planning are competitors in the proposed multiobjective CRP.

Introducing new models of the CRP was not enough; we also have to assure the practical application of these models, leading our way into the optimization techniques. We tested classical deterministic procedures and developed evolutionary metaheuristics for the mono-objective and multiobjective models. Nature-inspired algorithms deeply engage the reality of the CRP; they generate solutions in a broad range and quickly adapt in many scenarios. After performing several trials, PLGA and PLMGA results have shown that the proposed algorithms are efficient and produce valuable solutions to the CRP. Although we have analyzed several optimization scenarios, we believe that evaluating the generated solutions in the agrarian field is still a complex task and would take long-term analyzis to make conclusive statements.

Therefore, our contributions extend over the agriculture and optimization fields. We have tried really hard to build a strong standpoint on the CRP elements and formulate them into the proposed models. Developing optimization algorithms contributes mostly to the optimization field. From the GAs, we have presented new approaches in the encoding technique, the fitness evaluation and the population handling.

The future of this research would lead to developing dynamic models of the CRP, combining weather prediction and risk assessment, as well as increasing the response to upcoming market opportunities. Restricting certain sub-sequences in the crop sequence could reduce crop and soil diseases, and so, coupling illness features in the model would enhance the CRP capabilities. Testing other optimization techniques and developing novel stochastic procedures are still on the table as prospect alternatives.

9.1 Related works and publications

During the developments of this research, we have published an international conference paper (**A New Approach for Crop Rotation Problem in Farming 4.0**) and a national conference paper (**A multiobjective approach for crop rotation planning**). Recently, we submitted a journal paper entitled **Optimization in Agriculture: a novel metaheuristic for the multiobjective Crop Rotation Problem**, which has been under review currently.

MIRANDA, B.S., YAMAKAMI, A., RAMPAZZO, P.C.B. A New Approach for Crop Rotation Problem in Farming 4.0. In: Camarinha-Matos L., Almeida R., Oliveira J. (eds) Technological Innovation for Industry and Service Systems. DoCEIS 2019. IFIP Advances in Information and Communication Technology, vol 553. Springer, Cham. **Best Paper Award.**

MIRANDA, B. S.; YAMAKAMI, A.; RAMPAZZO, P. C. B. A multiobjective approach for crop rotation planning. In: LI SIMPóSIO BRASILEIRO DE PESQUISA OPERACIONAL, 2019, Limeira. Anais eletrônicos... Campinas, GALOÁ, 2019.

Bibliography

ABRAHAM, A.; JAIN, L. Evolutionary multiobjective optimization: Theoretical advances and applications. In: ______. 1. Londres: Springer-Verlag, 2005. cap. Evolutionary Multiobjective Optimization, p. 1–6. Cited on page 61.

ALFANDARI, L. et al. A mip flow model for crop rotation planning in a sustainable development context. *Annals of Operations Research*, v. 190, p. 149–164, 10 2011. Cited on page 23.

ALIANO, A.; FLORENTINO, H.; PATO, M. Metaheuristics for a crop rotation problem. *Int. J. Metaheuristics*, v. 3, n. 3, 2014. Cited 5 times on pages 22, 26, 27, 28, and 32.

ALIANO, A.; FLORENTINO, H. de O.; PATO, M. V. Metodologias de escalarizações para o problema de rotação de culturas biobjetivo. *Proceeding Series of the Brazilian Society of Applied and Computational Mathematics*, v. 6, n. 1, 2018. Cited on page 22.

ATCHATHA, M.; JAYAKUMAR, M. Crop rotation based on space and time. *2016 IEEE Technological Innovations in ICT for Agriculture and Rural Development (TIAR)*, p. 23–27, July 2016. Cited on page 21.

BASSANINO, M. et al. Nutrient balance as a sustainability indicator of different agro-environments in italy. *Ecological Indicators*, v. 1, n. 11, p. 721–723, 2011. Cited on page 38.

BEHNKE, G. D. et al. Long-term crop rotation and tillage effects on soil greenhouse gas emissions and crop production in Illinois, USA. *Agriculture, Ecosystems and Environment*, v. 261, p. 62 – 70, 2018. ISSN 0167-8809. Cited 2 times on pages 20 and 37.

BOLD, N.; NIJLOVEANU, D. Using genetic algorithms to outline crop rotations and a cropping-system model. *International Journal of Computer and Information Engineering*, v. 10, n. 6, p. 1028–1032, 2016. Cited on page 21.

BRANKE, J. et al. *Multiobjective Optimization*. Heidelberg, German: Springer-Verlag Berlin Heidelberg, 2008. Cited on page 63.

BRAUN, A.-T.; COLANGELO, E.; STECKEL, T. Farming in the era of industrie 4.0. *Procedia CIRP*, v. 72, p. 979 – 984, 2018. ISSN 2212-8271. 51st CIRP Conference on Manufacturing Systems. Disponível em: <http://www.sciencedirect.com/science/article/pii/S2212827118303342>. Cited 2 times on pages 16 and 20.

CAPITANESCU, F. et al. Multi-stage farm management optimization under environmental and crop rotation constraints. *Journal of Cleaner Production*, v. 147, p. 197 – 205, 2017. ISSN 0959-6526. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0959652617300835>. Cited on page 24.

CHANKONG, V.; HAIMES, Y. Y. *Multiobjective Decision Making Theory and Methodology*. New York, NY: Elsevier Science Publishing Co., 1983. Cited on page 66.

CHARNES, A.; COOPER, W. Goal programming and multiple objective optimizations: Part 1. *European Journal of Operational Research*, v. 1, n. 1, p. 39 – 54, 1977. ISSN 0377-2217. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0377221777810072>. Cited 2 times on pages 64 and 85.

CHARNES, A.; COOPER, W. W. *Management Models and the Industrial Applications of Linear Programming*. 1^{st}ed. New York: John Wiley, 1961. Cited on page 64.

CHARNES, A.; COOPER, W. W.; FERGUSON, R. O. Optimal estimation of executive compensation by linear programming. *Management Science*, v. 1, n. 2, p. 138–151, 1955. Cited on page 64.

CLARK, A. *Managing cover crops profitably*. 1. College Park, US: Sustainable Agriculture Research & Education (SARE) program, 2012. Cited 2 times on pages 40 and 114.

CLíMACO, J. N.; ANTUNES, C. H.; ALVES, M. J. G. *Programação linear multiobjectivo: do modelo de programação linear clássico à consideração explícita de várias funções objectivo*. 1*st* ed. Coimbra, Portugal: Imprensa da Universidade de Coimbra / Coimbra University Press, 2003. Cited on page 60.

DEB, K. et al. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, v. 6, n. 2, p. 182–197, April 2002. ISSN 1089-778X. Cited 2 times on pages 66 and 68.

DEEP, K. et al. A new crossover operator for real coded algorithms. *Applied Mathematics and Computation*, v. 1, n. 1, p. 895–911, 2007. Cited on page 54.

DEEP, K. et al. A real coded genetic algorithm for solving integer and mixed integer optimization problems. *Applied Mathematics and Computation*, v. 1, n. 1, p. 505–518, 2009. Cited on page 30.

DEEP, K.; THAKUR, M. A new mutation operator for real coded genetic algorithms. *Applied Mathematics and Computation*, v. 193, n. 1, p. 211 – 230, 2007. ISSN 0096-3003. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0096300307003918>. Cited 2 times on pages 56 and 57.

DETLEFSEN, N. K.; JENSEN, A. L. Modelling optimal crop sequences using network flows. *Agricultural Systems*, v. 94, p. 566–572, May 2007. Cited on page 23.

DURY, J. et al. Models to support cropping plan and crop rotation decisions. a review. *Agronomy for Sustainable Development*, v. 32, n. 2, p. 567–580, Apr 2012. ISSN 1773-0155. Disponível em: <https://doi.org/10.1007/s13593-011-0037-x>. Cited on page 20.

EDWARDS, L.; BURNEY, J. An overview of nutrient budgets for use in nutrient management planning. Sustainable Agriculture Research & Education (SARE) program. 2005. Cited on page 39.

FAR, S. T.; REZAEI-MOGHADDAM, K. Impacts of the precision agricultural technologies in iran: An analysis experts' perception & their determinants. *Information Processing in Agriculture*, v. 5, n. 1, p. 173 – 184, 2018. ISSN 2214-3173. Disponível em: <http://www.sciencedirect.com/science/article/pii/S2214317316301329>. Cited on page 16.

FERREIRA, P. A. V. *Otimização Multiobjetivo: Teoria e Aplicações*. Tese (Tese de Livre Docência) — Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, SP, 1999. Cited on page 60.

FINCH, H.; SAMUEL, A.; LANE, G. 8 - cropping techniques. In: FINCH, H.; SAMUEL, A.; LANE, G. (Ed.). *Lockhart & Wiseman's Crop Husbandry Including Grassland (Ninth Edition)*. Ninth edition. Woodhead Publishing, 2014. p. 193 – 217. ISBN 978-1-78242-371-3. Disponível em: <http://www.sciencedirect.com/science/article/pii/B9781782423713500084>. Cited on page 16.

FORRESTER, R. J.; RODRIGUEZ, M. An integer programming approach to crop rotation planning at an organic farm. *The UMAP Journal*, p. 5–23, 2018. Cited 4 times on pages 16, 21, 28, and 29.

FRANCHINI, J. C. et al. Evolution of crop yields in different tillage and cropping systems over two decades in southern brazil. *Field Crops Research*, v. 137, p. 178–185, 2012. Cited on page 23.

FRANCIS, C. Crop rotations. In: HILLEL, D. (Ed.). *Encyclopedia of Soils in the Environment*. Oxford: Elsevier, 2005. p. 318 – 322. ISBN 978-0-12-348530-4. Disponível em: <http://www.sciencedirect.com/science/article/pii/B0123485304002538>. Cited on page 16.

GASS, S.; SAATY, T. The computational algorithm for the parametric objective function. In: . [S.l.: s.n.], 1955. v. 2, p. 39 – 45. Cited on page 64.

GHORBANI, R. et al. Soil management for sustainable crop disease control: a review. *Environ Chem Lett*, v. 6, p. 149–162, 2008. Cited on page 21.

GOLDBERG, D. E. *Genetic algorithms in search, optimization, and machine learning*. New York, US: Addison-Wesley Publishing Company, 1989. Cited on page 46.

GOLDBERG, D. E.; DEB, K. A comparative analysis of selection schemes used in genetic algorithms. Morgan Kaufmann, p. 69–93, 1991. Cited 2 times on pages 48 and 54.

GPS. *Official U.S. government information about the Global Positioning System (GPS) and related topics*. 2018. <https://www.gps.gov/applications/agriculture/>. Access date: September 11th, 2019. Cited on page 16.

HAIMES YV, Y.; LASDON, L.; WISMER DA, D. On a bicriterion formation of the problems of integrated system identification and system optimization. *IEEE Transactions on Systems, Man and Cybernetics*, Institute of Electrical and Electronics Engineers Inc., SMC-1, n. 3, p. 296–297, 1 1971. ISSN 0018-9472. Cited on page 66.

HANEVELD, W.; STEGEMAN, A. Crop succession requirements in agricultural production planning. *European Journal of Operational Research*, v. 166, n. 2, p. 406 – 429, 2005. ISSN 0377-2217. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0377221704002358>. Cited on page 23.

IMADI, S. et al. Sustainable crop production system. In: HAKEEM, K. R.; AKHTAR, M. S.; ABDULLAH, S. N. A. (Ed.). *Plant, Soil and Microbes: Volume 1: Implications in Crop Science*. [S.l.]: Springer International Publishing Switzerland, 2016. p. 103–116. ISBN 978-3-319-27455-3. Cited on page 21.

JEDIDI, A.; CAMINADA, A.; FINKE, G. 2-objective optimization of cells overlap and geometry with evolutionary algorithms. In: RAIDL, G. R. et al. (Ed.). *Applications of Evolutionary Computing*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004. p. 130–139. Cited on page 91.

JONES, D.; TAMIZ, M. *Practical Goal Programming*. 1^{st}ed. New York: Springer US, 2010. Cited on page 64.

JULIAN, J. et al. *Enterprise Budget: Broccoli, Processed Market, Willamette Valley Region*. 2010. <http://arec.oregonstate.edu/oaeb/files/pdf/AEB0003.pdf>. Access date: August 30th, 2019. Cited on page 120.

JULIAN, J. et al. *Enterprise Budget: Sweet Corn, Processed Market, Willamette Valley Region*. 2010. <http://arec.oregonstate.edu/oaeb/files/pdf/AEB0006.pdf>. Access date: August 30th, 2019. Cited on page 120.

KAMILARIS, A.; KARTAKOULLIS, A.; PRENAFETA-BOLDú, F. A review on the practice of big data analysis in agriculture. *Computers and Electronics in Agriculture*, v. 143, p. 23–37, 2017. Cited on page 16.

KNOWLES, J.; CORNE, D.; DEB, K. *Multiobjective Problem Solving from Nature From Concepts to Applications*. 1 ed. Berlin: Springer-Verlag, 2008. Cited 2 times on pages 18 and 46.

LIN, T. T.; HSIEH, C.-S. A decision analysis for the dynamic crop rotation model with markov process's concept. *2013 IEEE International Conference on Industrial Engineering and Engineering Management*, v. 11, n. 4, p. 159–163, 2013. Cited on page 21.

LOSS, A. et al. Carbono orgânico total e agregação do solo em sistema de plantio direto agroecológico e convencional de cebola. *Revista Brasileira de Ciência do Solo*, v. 1, n. 39, p. 1212–1224, 2015. Cited on page 39.

LUKOWIAK, R.; GRZEBISZ, W.; SASSENRATH, G. F. New insights into phosphorus management in agriculture — a crop rotation approach. *Science of The Total Environment*, v. 542, p. 1062 – 1077, 2016. ISSN 0048-9697. Special Issue on Sustainable Phosphorus Taking stock: Phosphorus supply from natural and anthropogenic pools in the 21st Century. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0048969715306690>. Cited on page 22.

MARLER, R.; ARORA, J. The weighted sum method for multi-objective optimization: New insights. *Structural and Multidisciplinary Optimization*, v. 41, p. 853–862, 06 2010. Cited 2 times on pages 63 and 64.

MEISINGER, J.; CALDERON, F.; JENKINSON, D. Soil nitrogen budgets. p. 505–562, 01 2008. Cited on page 37.

MEMMAH, M.-M. et al. Metaheuristics for agricultural land use optimization. a review. *Agronomy for Sustainable Development*, v. 35, n. 3, p. 975–998, Jul 2015. ISSN 1773-0155. Disponível em: <https://doi.org/10.1007/s13593-015-0303-4>. Cited 2 times on pages 20 and 22.

MIETTINEN, K. *Nonlinear Multiobjective Optimization*. 1*st*. New York: Springer US, 1998. Cited on page 63.

MILLER, B.; GOLDBERG, D. Genetic algorithms, tournament selection, and the effects of noise. *Complex Systems*, n. 9, p. 193–212, 1995. Cited on page 54.

MOHLER, C.; JOHNSON, S. *Crop rotation on organic farms: a planning manual*. 1. New York, US: Natural Resource, Agriculture, and Engineering Service. Cooperative Extension, 2009. Cited 3 times on pages 15, 39, and 114.

NASS-USDA. *Usual Planting and Harvesting Dates for Field Crops*. 2018. <https://www.nass.usda.gov/Publications/Todays_Reports/reports/crop0918.pdf>. Access date: August 30th, 2019. Cited on page 114.

OENEMA, O.; KROS, H.; VRIES, W. Approaches and uncertainties in nutrient budgets: implications for nutrient management and environmental policies. *European Journal of Agronomy*, v. 1, n. 20, p. 3–16, 2003. Cited on page 38.

OREGON STATE UNIVERSITY. *OREGON AGRICULTURAL ENTERPRISE BUDGETS*. 2019. <http://arec.oregonstate.edu/oaeb/>. Access date: August 30th, 2019. Cited on page 120.

OUDA, S.; ZOHRY, A.; NORELDIN, T. *Crop Rotation Maintains Soil Sustainability: An Approach to Secure Future Food*. 1. ed. [S.l.]: Springer International Publishing, 2018. 55-76 p. ISBN 978-3-030-05350-5. Cited on page 16.

PAVóN, R.; BRUNELLI, R.; LüCKEN, C. von. Determining optimal crop rotations by using multiobjective evolutionary algorithms. *Environ Chem Lett*, v. 6, p. 147–154, 2009. Cited on page 22.

ROSENZWEIG, S. T.; STROMBERGER, M. E.; SCHIPANSKI, M. E. Intensified dryland crop rotations support greater grain production with fewer inputs. *Agriculture, Ecosystems and Environment*, v. 264, p. 63–72, 2018. Cited on page 22.

RÜTTING, T.; ARONSSON, H.; DELIN, S. Efficient use of nitrogen in agriculture. *Nutrient Cycling in Agroecosystems*, v. 110, n. 1, p. 1–5, Jan 2018. ISSN 1573-0867. Disponível em: <https://doi.org/10.1007/s10705-017-9900-8>. Cited on page 37.

SAHA, B. K. et al. Nitrogen dynamics in soil fertilized with slow release brown coal-urea fertilizers. *Scientific Reports*, n. 14577, p. 1–10, 2018. Cited on page 37.

SAINJU, U. Determination of nitrogen balance in agroecosystems. *MethodsX*, v. 1, n. 4, p. 199–208, 2017. Cited on page 38.

SANTOS, L. et al. Crop rotation scheduling with adjacency constraints. *Ann Oper Res*, v. 190, p. 165–180, 2011. Cited 4 times on pages 16, 20, 33, and 39.

SANTOS, L. M. R. dos et al. Crop rotation scheduling with adjacency constraints. *Springer Science+Business Media, LLC*, 26 Nov. 2008. Cited 4 times on pages 22, 24, 26, and 28.

SCHOTT, J. R. *Fault tolerant design using single and multicriteria genetic algorithm optimization*. Dissertação (Mestrado) — Massachusetts Institute of Technology, Dept. of Aeronautics and Astronautics, Cambridge, Massachusetts, 1995. Cited 3 times on pages 72, 91, and 92.

SEAVERT, C. et al. *Enterprise Budget: Leaf Lettuce, Organic, Fresh Market, Willamette Valley Region*. 2007. <http://arec.oregonstate.edu/oaeb/files/pdf/EM8931.pdf>. Access date: August 30th, 2019. Cited on page 120.

SEAVERT, C. et al. *Enterprise Budget: Spinach, Conventional, Fresh Market, Willamette Valley Region*. 2007. <http://arec.oregonstate.edu/oaeb/files/pdf/EM8928.pdf>. Access date: August 30th, 2019. Cited on page 120.

SHOBER, A.; HOCHMUTH, G.; WIESE, C. An overview of nutrient budgets for use in nutrient management planning. U.S. Department of Agriculture, UF/IFAS Extension Service, University of Florida. 2011. Cited on page 38.

SIGNOR, D.; CERRI, C. E. P. Nitrous oxide emissions in agricultural soils: a review. *Pesquisa Agropecuária Tropical*, v. 43, n. 3, p. 322–338, Jul/Sep 2013. Cited on page 37.

United States Department of Agriculture. *Usual Planting and Harvesting Dates for Fresh Market and Processing Vegetables*. 2007. <https://naldc.nal.usda.gov/download/CAT30992961/PDF>. Access date: August 30th, 2019. Cited on page 114.

UNIVERSITY OF ARKANSAS SYSTEM: DIVISION OF AGRICULTURE. *Arkansas Field Crop Enterprise Budgets*. 2019. <https://www.uaex.edu/farm-ranch/economics-marketing/farm-planning/budgets/crop-budgets.aspx>. Access date: August 30th, 2019. Cited on page 120.

USDA. *Usual Planting and Harvesting Dates for Field Crops*. 2010. <https://downloads.usda.library.cornell.edu/usda-esmis/files/vm40xr56k/dv13zw65p/w9505297d/planting-10-29-2010.pdf>. Access date: August 30th, 2019. Cited on page 114.

VANDERPLAATS, G. N. *Multidiscipline Design Optimization*. 1 ed. United States: Vanderplaats Research & Development, 2007. Cited on page 17.

WASHINGTON STATE UNIVERSITY. *Agriculture Business*. 2019. <https://pubs.extension.wsu.edu/agriculture-business>. Access date: August 30th, 2019. Cited on page 120.

WATSON, C.; ATKINSON, D. Using nitrogen budgets to indicate nitrogen use efficiency and losses from whole farm systems: a comparison of three methodological approaches. *Nutrient Cycling in Agroecosystems*, v. 1, n. 53, p. 259–267, 1999. Cited on page 38.

WIERZBICKI, A. P. A methodological approach to comparing parametric characterizations of efficient solutions. In: FANDEL, G. et al. (Ed.). *Large-Scale Modelling and Interactive Decision Analysis*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1986. p. 27–45. ISBN 978-3-662-02473-7. Cited on page 65.

ZADEH, L. Optimality and non-scalar-valued performance criteria. *IEEE Transactions on Automatic Control*, v. 8, n. 1, p. 59–60, January 1963. ISSN 0018-9286. Cited on page 64.

ZUBER, S. M. et al. Multivariate assessment of soil quality indicators for crop rotation and tillage in illinois. *Soil and Tillage Research*, v. 174, p. 147 – 155, 2017. ISSN 0167-1987. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0167198717301344>. Cited on page 23.

Appendix

APPENDIX A – Composing a CRP database

Even though large and long-term experiments are usual in agriculture, it is still hard to assemble a database with all the parameters we need to evaluate the proposed models in this research. In general, agrarian experiments are regional and deep connected to the agrarian practices of the local farmers, almost impossible to get reasonable parameters to the macro environment. In opposite, other experiments are laboratory tests and they conceal few aspects from the agrarian business model.

Using a real data of CRP from the United States provides high degree of comprehension in the agrarian field, which is essential to propose new approaches and optimization models. With real information, we can assure that our models are close to the reality of the problem. Hence, after spending many research hours and reviewing several references, we managed to build a well-grounded database from combining renowned sources.

From the United States Department of Agriculture (USDA) and the National Agricultural Statistics Service (NASS), we gathered the usual seeding and harvesting intervals of each crop presented in Table 15 and Table 16. Our references are mainly reports in United States Department of Agriculture (2007) and USDA (2010).

Cropping production (typical yields) in Tables 17 and 18 is mostly from NASS-USDA (2018). Units in these tables also represents the usual for each crop in the USA market. Cropping family are public domain information. Nutrient removals (N (kg/acre), P (kg/acre) and K (kg/acre)) are adapted from Mohler e Johnson (2009).

Ranking cover crops in Table 19 bases on Clark (2012). Clark (2012) classified cover crops using absolute categorization rating (excellent / poor), which we have transposed to a numerical scale from 0 (poor) to 4 (excellent).

Profits from Tables 17 and 18 are gathered from several sources. Using cropping budgets and crossing data, we were able to pull off the profit column. Appendix B presents some of budgets we have explored in this dissertation.

Table 15: Crop's attributes: seeding and harvesting.

Index	Crops	Seeding - Begin	Seeding - End	Harvesting - Begin	Harvesting - End	Cycle (Days)
1	Leaf Lettuce for Fresh Market, Winter	1-Sep	31-Jan	1-Nov	30-Apr	60
2	Leaf Lettuce for Fresh Market, Spring	1-Jan	31-Mar	1-Apr	31-Jul	90
3	Leaf Lettuce for Fresh Market, Summer	1-Apr	31-Jul	1-Jun	31-Oct	60
4	Leaf Lettuce for Fresh Market, Fall	1-Aug	30-Sep	1-Oct	30-Nov	60
5	Sweet corn, Spring	1-Oct	31-Jan	1-Jan	31-Mar	120
6	Sweet corn, Summer	1-Jan	15-Apr	1-Apr	10-Jul	120
7	Sweet corn, Winter	1-Feb	31-May	15-Jun	10-Sep	120
8	Broccoli for Fresh Market, Winter	1-Aug	15-Dec	1-Nov	15-Mar	120
9	Broccoli for Fresh Market, Spring	1-Mar	31-May	15-Apr	15-Jun	120
10	Broccoli for Fresh Market, Summer	1-Jun	31-Aug	15-Jul	10-Sep	120
11	Broccoli for Fresh Market, Fall	1-Sep	30-Nov	15-Oct	15-Dec	120
12	Tomato for Fresh Market, Spring	15-Jan	31-May	1-May	31-Jul	120
13	Tomato for Fresh Market, Summer	1-May	15-Jun	1-Jul	1-Oct	60
14	Spinach for Fresh Market, Winter	1-Oct	1-Mar	1-Nov	31-Mar	60
15	Spinach for Fresh Market, Spring	15-Jan	25-Apr	1-May	31-May	120
16	Spinach for Fresh Market, Summer	15-Mar	15-Aug	15-Jun	15-Nov	90
17	Spinach for Fresh Market, Fall	10-Aug	2-Sep	15-Sep	24-Dec	60
18	Summer Squash for Fresh Market	5-May	15-Jul	1-Jul	31-Oct	60
19	Carrots for Fresh Market, Winter	1-Jul	30-Sep	1-Nov	1-Mar	120
20	Carrots for Fresh Market, Spring	1-Nov	30-Nov	1-Mar	30-Jun	120
21	Carrots for Fresh Market, Summer	1-Dec	31-Mar	1-May	31-Jul	150
22	Potatoes for Fresh Market, Fall	2-May	10-Jun	6-Sep	17-Oct	120
23	Potatoes for Fresh Market, Summer	8-Apr	27-May	6-Aug	4-Oct	120
24	Watermelons for Fresh Market, Spring	15-Dec	31-Mar	15-May	15-Jul	150
25	Watermelons for Fresh Market, Summer	1-Mar	30-Jun	30-Jun	31-Aug	120
26	Cucumber for Fresh Market, Winter	1-Nov	31-Dec	1-Jan	15-Feb	60
27	Cucumber for Fresh Market, Spring	1-Jan	15-Feb	15-Mar	30-Jun	90
28	Cucumber for Fresh Market, Summer	1-May	10-Jun	15-Aug	15-Oct	90
29	Cucumber for Fresh Market, Fall	1-Jun	15-Jul	1-Aug	30-Sep	60
30	Cabbage for Fresh Market, Spring	1-Sep	1-Feb	1-Nov	1-Apr	60
31	Cabbage for Fresh Market, Summer	10-Apr	15-Jul	15-Jun	20-Nov	60
32	Spring Onions for Fresh Market	1-Oct	31-Dec	1-May	31-Jul	120
33	Strawberries for Fresh Market, Winter	20-Sep	10-Nov	25-Nov	25-Apr	60
34	Strawberries for Fresh Market, Spring	1-Dec	31-Mar	25-Mar	30-Sep	90

Table 16: Crop's attributes: continuation 2.

Index	Crops	Planting - Begin	Planting - End	Harvesting - Begin	Harvesting - End	Cycle (Days)
35	Strawberries for Fresh Market, Summer	1-Apr	15-May	1-Jun	15-Jul	60
36	Bell Peppers for Fresh Market and Processing, Winter	15-Sep	15-Dec	1-Jan	31-Mar	120
37	Bell Peppers for Fresh Market and Processing, Spring	15-Nov	1-Mar	1-Apr	30-Jun	120
38	Bell Peppers for Fresh Market and Processing, Summer	1-May	10-Jun	1-Jul	15-Oct	90
39	Annual rye-grass, Early Spring	15-Mar	1-May	20-Apr	15-May	300
40	Annual rye-grass, Late Summer	15-Aug	25-Sep	20-Apr	15-May	300
41	Annual rye-grass, Fall	1-Oct	5-Dec	20-Apr	15-May	300
42	Barley, Fall	1-Oct	30-Nov	20-Apr	15-May	300
43	Barley, Winter	1-Dec	25-Jan	4-Jul	20-Jul	120
44	Barley, Spring	1-Mar	31-May	10-Jul	30-Jun	120
45	Oats, green manure	5-Apr	10-May	5-Jun	20-Jun	60
46	Oats, Winter Cover	15-Aug	25-Sep	5-Jun	20-Jun	60
47	Rye	15-Sep	15-Dec	10-May	20-Jun	300
48	Wheat	1-Sep	15-Dec	20-May	7-Jun	300
49	Spelt	1-Sep	15-Dec	20-May	7-Jun	300
50	Triticale	25-Aug	15-Dec	20-May	7-Jun	300
51	Buckwheat	20-May	30-Aug	4-Jul	15-Oct	60
52	Sorghum-sudangrass, Late Spring, Summer	15-May	31-Jul	20-Sep	30-Nov	90
53	Mustard, Spring	20-Mar	31-May	15-May	31-May	90
54	Mustard, Late Summer	15-Aug	30-Sep	20-Apr	15-May	300
55	Forage Radish, Spring	20-Aug	10-Sep	20-Apr	15-May	300
56	Forage Radish, Late Summer, Early Fall	15-Sep	30-Nov	20-Apr	15-May	300
57	Rapeseed (or Canola), Fall	30-Sep	30-Nov	20-Apr	15-May	270
58	Rapeseed (or Canola), Spring	20-Mar	31-May	20-Apr	15-May	270
59	Crimson clover, Winter	5-Feb	20-Mar	5-Feb	20-Mar	180
60	Crimson clover, Summer	20-Jun	31-Aug	5-Feb	20-Mar	300
61	Field peas, Fall	15-Sep	20-Nov	25-Apr	10-May	240
62	Field peas, Early Spring	20-Mar	30-Apr	25-Apr	10-May	240
63	Hairy vetch, Fall	20-Aug	10-Sep	15-May	31-May	270
64	Hairy vetch, Early Spring	20-Mar	30-Apr	15-May	31-May	270
65	Red clover, Late Summer	20-Aug	30-Sep	20-Apr	10-Jul	360
66	Red clover, Early Spring	20-Mar	30-Apr	15-May	31-May	360
67	White clover, Late Winter, Spring	5-Feb	10-May	20-Apr	15-May	240

Table 17: Crop's attributes: profit and nutrient demand.

Index	Crops	Family	Typical yield (Unit/acre)	Unit	Profit ($/Unit)	N (kg/acre)	P (kg/acre)	K (kg/acre)
1	Leaf Lettuce for Fresh Market, Winter	Lettuce	569	carton	5.712	27.22	13.06	54.43
2	Leaf Lettuce for Fresh Market, Spring	Lettuce	569	carton	5.712	27.22	13.06	54.43
3	Leaf Lettuce for Fresh Market, Summer	Lettuce	569	carton	5.712	27.22	13.06	54.43
4	Leaf Lettuce for Fresh Market, Fall	Lettuce	569	carton	5.712	27.22	13.06	54.43
5	Sweet corn, Spring	Grass	12	ton	26.736	24.95	3.49	12.47
6	Sweet corn, Summer	Grass	12	ton	26.736	24.95	3.49	12.47
7	Sweet corn, Winter	Grass	12	ton	26.736	24.95	3.49	12.47
8	Broccoli for Fresh Market, Winter	Mustard	6.1	ton	478.217	12.7	3.63	18.14
9	Broccoli for Fresh Market, Spring	Mustard	6.1	ton	478.217	12.7	3.63	18.14
10	Broccoli for Fresh Market, Summer	Mustard	6.1	ton	478.217	12.7	3.63	18.14
11	Broccoli for Fresh Market, Fall	Mustard	6.1	ton	478.217	12.7	3.63	18.14
12	Tomato for Fresh Market, Spring	Nightshade	1500	box	7.111	15.42	7.26	27.22
13	Tomato for Fresh Market, Summer	Nightshade	1500	box	7.111	15.42	7.26	27.22
14	Spinach for Fresh Market, Winter	Beet	8.53	ton	227.774	18.14	3.63	18.14
15	Spinach for Fresh Market, Spring	Beet	8.53	ton	227.774	18.14	3.63	18.14
16	Spinach for Fresh Market, Summer	Beet	8.53	ton	227.774	18.14	3.63	18.14
17	Spinach for Fresh Market, Fall	Beet	8.53	ton	227.774	18.14	3.63	18.14
18	Summer Squash for Fresh Market	Cucurbit	176	cwt	9.796	13.88	3.08	0
19	Carrots for Fresh Market, Winter	Carrot	430	cwt	3.403	18.87	4.72	47.17
20	Carrots for Fresh Market, Spring	Carrot	430	cwt	3.403	18.87	4.72	47.17
21	Carrots for Fresh Market, Summer	Carrot	430	cwt	3.403	18.87	4.72	47.17
22	Potatoes for Fresh Market, Fall	Nightshade	450	cwt	-0.812	50.35	9.53	68.04
23	Potatoes for Fresh Market, Summer	Nightshade	450	cwt	-0.812	50.35	9.53	68.04
24	Watermelons for Fresh Market, Spring	Cucurbit	25	ton	44.82	28.58	5.44	36.06
25	Watermelons for Fresh Market, Summer	Cucurbit	25	ton	44.82	28.58	5.44	36.06
26	Cucumber for Fresh Market, Winter	Cucurbit	161	cwt	-12.15	28.12	9.98	47.17
27	Cucumber for Fresh Market, Spring	Cucurbit	161	cwt	-12.15	28.12	9.98	47.17
28	Cucumber for Fresh Market, Summer	Cucurbit	161	cwt	-12.15	28.12	9.98	47.17
29	Cucumber for Fresh Market, Fall	Cucurbit	161	cwt	-12.15	28.12	9.98	47.17
30	Cabbage for Fresh Market, Spring	Mustard	832	box	3.07	49.21	12.7	52.39
31	Cabbage for Fresh Market, Summer	Mustard	832	box	3.07	49.21	12.7	52.39
32	Spring Onions for Fresh Market	Lily	27.6	ton	17.18	30.62	5.67	30.62
33	Strawberries for Fresh Market, Winter	Rose	10192	lbs	0.097	4.54	1.36	9.98
34	Strawberries for Fresh Market, Spring	Rose	10192	lb	0.097	4.54	1.36	9.98

Table 18: Crop's attributes: continuation 1.

Index	**Crops**	**Family**	**Typical yield (Unit/acre)**	**Unit**	**Profit ($/Unit)**	**N (kg/acre)**	**P (kg/acre)**	**K (kg/acre)**
35	Strawberries for Fresh Market, Summer	Rose	10192	lb	0.097	4.54	1.36	9.98
36	Bell Peppers for Fresh Market and Processing, Winter	Nightshade	1280	carton	3.139	18.14	2.72	20.87
37	Bell Peppers for Fresh Market and Processing, Spring	Nightshade	1280	carton	3.139	18.14	2.72	20.87
38	Bell Peppers for Fresh Market and Processing, Summer	Nightshade	1280	carton	3.139	18.14	2.72	20.87
39	Annual rye-grass, Early Spring	Grass	1		0	0	0	0
40	Annual rye-grass, Late Summer	Grass	1		0	0	0	0
41	Annual rye-grass, Fall	Grass	1		0	0	0	0
42	Barley, Fall	Grass	1		0	0	0	0
43	Barley, Winter	Grass	1		0	0	0	0
44	Barley, Spring	Grass	1		0	0	0	0
45	Oats, green manure	Grass	1		0	0	0	0
46	Oats, Winter Cover	Grass	1		0	0	0	0
47	Rye	Grass	1		0	0	0	0
48	Wheat	Grass	1		0	0	0	0
49	Spelt	Grass	1		0	0	0	0
50	Triticale	Grass	1		0	0	0	0
51	Buckwheat	Buckwheat	1		0	0	0	0
52	Sorghum-sudangrass, Late Spring, Summer	Grass	1		0	0	0	0
53	Mustard, Spring	Mustard	1		0	0	0	0
54	Mustard, Late Summer	Mustard	1		0	0	0	0
55	Forage Radish, Spring	Mustard	1		0	0	0	0
56	Forage Radish, Late Summer, Early Fall	Mustard	1		0	0	0	0
57	Rapeseed (or Canola), Fall	Mustard	1		0	0	0	0
58	Rapeseed (or Canola), Spring	Mustard	1		0	0	0	0
59	Crimson clover, Winter	Legume	1		0	0	0	0
60	Crimson clover, Summer	Legume	1		0	0	0	0
61	Field peas, Fall	Legume	1		0	0	0	0
62	Field peas, Early Spring	Legume	1		0	0	0	0
63	Hairy vetch, Fall	Legume	1		0	0	0	0
64	Hairy vetch, Early Spring	Legume	1		0	0	0	0
65	Red clover, Late Summer	Legume	1		0	0	0	0
66	Red clover, Early Spring	Legume	1		0	0	0	0
67	White clover, Late Winter, Spring	Legume	1		0	0	0	0

Table 19: Cover crop's performance and roles.

Index	Crops	Average Total N (lb./A)	Average Dry Matter (lb./A/yr.)	Dry matter	N Scav-enger	Soil Builder	Erosion Fighter	Weed Fighter	Good Grazing	Quick Growth	Lasting Residue	Duration	Harvest Value - Forage	Harvest Value - Seed	Cash Crop Inter-seed
39	Annual rye-grass, Early Spring	0	5500	2.4	3	3	3	3	3	3	3	3	2	1	4
40	Annual rye-grass, Late Summer	0	5500	2.4	3	3	3	3	3	3	3	3	2	1	4
41	Annual rye-grass, Fall	0	5500	2.4	3	3	3	3	3	3	3	3	2	1	4
42	Barley, Fall	0	5500	2.4	3	3	3	3	3	3	3	3	2	1	4
43	Barley, Winter	0	6000	2.7	3	3	4	4	3	3	4	2	3	2	3
44	Barley, Spring	0	6000	2.7	3	3	4	4	3	3	4	2	3	2	3
45	Oats, green manure	0	6000	2.7	3	2	3	3	2	4	2	1	2	2	4
46	Oats, Winter Cover	0	6000	2.7	3	2	3	3	2	4	2	1	2	2	4
47	Rye	0	6500	2.9	4	4	4	4	2	4	4	3	1	1	3
48	Wheat	0	5500	2.4	3	3	3	3	3	3	3	3	2	3	1
49	Spelt	0	5500	2.4	3	3	3	3	3	3	3	3	2	3	1
50	Triticale	0	5500	2.4	3	3	3	3	3	3	3	3	2	3	1
51	Buckwheat	0	3000	1.3	0	2	1	1	3	4	0	1	0	1	3
52	Sorghum-sudangrass, Late Spring, Summer	0	9000	4.0	4	4	4	4	2	4	3	4	4	0	0
53	Mustard, Spring	75	6000	2.7	2	3	3	3	2	3	1	2	0	1	0
54	Mustard, Late Summer	75	6000	2.7	2	3	3	3	2	3	1	2	0	1	0
55	Forage Radish, Spring	125	5500	2.4	4	3	3	3	2	3	1	2	3	1	1
56	Forage Radish, Late Summer, Early Fall	125	5500	2.4	4	3	3	3	2	3	1	2	3	1	1
57	Rapeseed (or Canola), Fall	100	3500	1.6	3	2	3	3	4	3	2	3	1	4	0
58	Rapeseed (or Canola), Spring	100	3500	1.6	3	2	3	3	4	3	2	3	1	4	0
59	Crimson clover, Winter	100	4500	2.0	2	3	3	3	3	2	2	1	4	3	4
60	Crimson clover, Summer	100	4500	2.0	2	3	3	3	3	2	2	1	4	3	4
61	Field peas, Fall	120	4500	2.0	1	2	3	3	2	3	1	2	4	3	4
62	Field peas, Early Spring	120	4500	2.0	1	2	3	3	2	3	1	2	4	3	4
63	Hairy vetch, Fall	145	3650	1.6	1	3	2	2	2	1	1	3	1	3	2
64	Hairy vetch, Early Spring	145	3650	1.6	1	3	2	2	2	1	1	3	1	3	2
65	Red clover, Late Summer	110	3500	1.6	2	3	2	2	3	1	1	2	4	3	4
66	Red clover, Early Spring	110	3500	1.6	2	3	2	2	3	1	1	2	4	3	4
67	White clover, Late Winter, Spring	140	4000	1.8	1	2	3	3	3	1	1	4	3	2	3

APPENDIX B – Enterprise Budgets

Profit information are based on several sources, such as OREGON STATE UNIVERSITY (2019), WASHINGTON STATE UNIVERSITY (2019) and UNIVERSITY OF ARKANSAS SYSTEM: DIVISION OF AGRICULTURE (2019). We have composed cropping enterprise budgets concealing specific characteristics and cultural practices. We are not going to present all the enterprise budgets in this dissertation, but we selected some of them in Tables 20, 21, 22 and 23.

Leaf lettuce budget in Table 20 is based on Seavert et al. (2007a), we have updated some information and adjusted inflation rate to be more coherent. Sweet corn budget in the Table 21 is adapted from Julian et al. (2010b). Broccoli enterprise budget in Table 22 is related to Julian et al. (2010a) and, the last one, spinach enterprise budget, is based on Seavert et al. (2007b).

Table 20: Leaf lettuce, production budget.

Leaf Lettuce, Organic, Fresh Market, $/acre economic costs and returns							
GROSS INCOME			Quantity	Unit	$/Unit	Total	Price/Carton
Leaf Lettuce			569	cartons	$14.00	$7,966.00	$14.0000
Total gross income						$7,966.00	$14.0000
VARIABLE CASH COSTS	Description	Units	Labor	Machinery	Materials	Total	Cost/Carton
Field Preparations and Planting							
Lime application, custom	0.25	x/acre	$-	$-	$75.00	$75.00	$0.1318
Seed Cover Crop	0.5	x/acre	$1.77	$3.80	$20.00	$25.57	$0.0449
Disk down cover crop	1	x/acre	$3.44	$7.55	$-	$10.99	$0.0193
Deep chisel	0.5	x/acre	$3.64	$7.87	$-	$11.51	$0.0202
Disk before plowing	2	x/acre	$6.87	$17.87	$-	$21.96	$0.03865
Moldboard plow	1	x/acre	$9.70	$22.23	$-	$31.93	$0.0561
Disk	2	x/acre	$6.87	$15.09	$-	$21.96	$0.0386
Transplanting	1	x/acre	$205.38	$63.04	$665.00	$844.42	$1.4840
Planting labor	18	hours					
Lettuce transplants	0.024	each					
Lettuce transplants	24,000	transplanting					
Top-dress application	1	x/acre	$3.05	$4.89	$101.00	$107.94	$0.1897
Cultivating weeds	3	x/acre	$24.74	$32.75	$-	$57.49	$0.1010
Weed control			$300.00	$-	$-	$300.00	$0.5272
Hand labor	30	hours					
Irrigation			$22.50	$-	$85.00	$107.50	$0.1889
Labor, 7.50/set	3	sets					
Electricity, 10.00/set	3	sets					
Pipe rental, 110.00/acre	0.5	acre					
Spray insecticides	2	x/acre	$6.09	$9.78	$300.00	$315.87	$0.5551
Organic Certification			$-	$-	$45.50	$45.50	$0.0800
Fee per gross income	0.005	per					
Harvesting							
Harvesting labor	5	carton/hr	$1,300.00	$-	$-	$1,300.00	$2.2847
Tractor and trailer	6.5	hours	$78.00	$125.74	$-	$203.74	$0.3581
Packing and Materials							
Cartons	650	carton	$-	$-	$845.00	$845.00	$1.4851
Hydro-cooling	650	carton	$-	$-	$65.00	$65.00	$0.1142
Refrigeration	650	carton	$-	$-	$65.00	$65.00	$0.1142
Delivery to market	650	carton	$52.00	$-	$-	$52.00	$0.0914
Other Costs							
Pickups, truck and ATV	0.5	x/acre	$-	$103.38	$-	$103.38	$0.1817
Shop and machine shed	0.5	x/acre	$-	$-	$4.00	$4.00	$0.0070
Miscellaneous and overhead	0.5	x/acre	$-	$-	$37.50	$37.50	$0.0659
Interest: operating capital	6	months	$-	$-	$95.40	$95.40	$0.1677
Total variable costs			$2,032.69	$429.08	$2,403.40	$4,865.17	$8.5504
FIXED CASH COSTS					Unit	Total	Cost/Carton
Property insurance					acre	$17.50	$0.0308
Property taxes					acre	$17.50	$0.0308
Field sanitation equipment					acre	$15.00	$0.0264
Land rent					acre	$200.00	$0.3515
Total cash costs						$250.00	$0.4394
FIXED NON-CASH COSTS					Unit	Total	Cost/Carton
Machinery and equip - depreciation, interest and insurance					acre	$115.19	$0.2024
Pickups, truck and ATV - depreciation, interest and insurance					acre	$47.77	$0.0840
Shop and machine shed - depreciation, interest and insurance					acre	$10.88	$0.0191
Total non-cash costs						$173.84	$0.3055
Total fixed costs						$423.84	$0.07449
Total of all costs per acre						$5,289.01	$9.2953
Net projected returns						$2,676.99	$4.7047
U.S. Inflation Rate	2007	2019	21.4%			$3,249.87	$5.712

Table 21: Sweet corn, production budget.

Sweet Corn, Processed Market, $/acre economic costs and returns							
GROSS INCOME			Quantity	Unit	$/Unit	Total	Price/Ton
Sweet Corn			12	Ton	$105.00	$1,260.00	$105.00
Total gross income							
VARIABLE CASH COSTS	Description	Units	Labor	Machinery	Materials	Total	Cost/Ton
Field Preparations and Planting							
V-Rip	1	x/acre	$2.54	$5.77	$-	$8.31	$0.6925
Harrow/Roller Packer	2	x/acre	$3.05	$7.67	$65.00	$75.72	$6.3100
Tandem Disk Harrow	1	x/acre	$1.69	$4.29	$-	$5.98	$0.4983
Plant Corn	1	x/acre	$2.54	$6.57	$170.00	$179.11	$14.9258
Seed	$100						
Pre-harvest							
Cultivating weeds	1	x/acre	$0.85	$1.84	$-	$2.69	$0.2242
Pest Control, Boom Sprayer	2	x/acre	$0.86	$0.93	$70.00	$71.79	$5.9825
Herbicide	$30						
Insecticide	$40						
Soil analysis	1	x/acre	$0.69	$0.97	$60.00	$61.66	$5.1383
Top Dress	$60						
Custom Topping	1	x/acre	$-	$-	$10.00	$10.00	$0.8333
Irrigation			$33.00	$-	$85.00	$118.00	$9.8333
Labor, $11.00	3	hours					
Electricity, $3.50	10	acre-inch					
Maintenance and Repairs, $50.00	1	x/acre					
Harvesting							
Corn Picker	1	x/acre	$4.32	$27.02	$-	$31.34	$2.6117
Truck	2	x/acre	$-	$7.45	$-	$7.45	$0.6208
Post-harvest							
Soil Test	1	x/acre	$-	$-	$2.00	$2.00	$0.1667
Lime application, custom	0.25	x/acre	$-	$-	$75.00	$75.00	$6.2500
Disk	2	x/acre	$3.05	$8.59	$-	$11.64	$0.9700
Other Costs							
Pickup and ATV	1	x/acre	$-	$5.31	$-	$5.31	$0.4425
Interest: operating capital	6	months	$-	$-	$24.04	$24.04	$2.0033
Total variable costs			$52.59	$76.41	$561.04	$690.04	$57.5033
FIXED CASH COSTS					Unit	Total	Cost/Ton
Property insurance					acre	$25.00	$2.08
Property taxes					acre	$20.00	$1.67
Land Rent					acre	$200.00	$16.67
Total fixed cash costs						$245.00	$20.42
FIXED NON-CASH COSTS					Unit	Total	Cost/Ton
Machinery and equip - depreciation, interest and insurance					acre	$32.73	$2.73
Pickup, truck and ATV - depreciation, interest and insurance					acre	$14.29	$1.19
Total fixed non-cash costs						$47.02	$3.92
Total fixed costs						$292.02	$24.34
Total of all costs per acre						$982.06	$81.84
Net projected returns						$277.94	$23.16
U.S. Inflation Rate	2010	2019	15.43%			$320.83	$26.7355

Table 22: Broccoli, production budget.

Broccoli, Processed Market, 2010, $/acre economic costs and returns							
GROSS INCOME			Quantity	Unit	$/Unit	Total	Price/Ton
Broccoli			6.1	Ton	745	4544.5	$745.00
VARIABLE CASH COSTS	Description	Units	Labor	Machinery	Materials	Total	Cost/Ton
Field Preparations and Planting							
Tandem Disk Harrow	2	x/acre	$3.39	$8.59	$-	$11.98	$1.9639
Mold Board Plow	1	x/acre	$2.13	$5.25	$-	$7.38	$1.2098
Harrow/Roller Packer	1	x/acre	$1.52	$3.83	$65.00	$70.35	$11.5328
Field Cultivator	2	x/acre	$5.86	$15.04	$-	$20.90	$3.4262
Rotovator	1	x/acre	$3.81	$10.98	$20.00	$34.79	$5.7033
Plant Broccoli	1	x/acre	$2.54	$6.57	$240.00	$179.11	$40.8377
Seed	$220.00						
Insecticide	$20.00						
Pre-harvest							
Cultivating weeds	1	x/acre	$1.54	$2.81	$65.00	$69.34	$11.3689
Hand Weed	14	hours	$154.00	$-	$-	$154.00	$25.2459
Self-propelled Boom Sprayer	2	x/acre	$0.86	$0.93	$40.00	$41.79	$6.8508
Insecticide	$20.00						
Irrigation			$55.00	$-	$85.00	$140.00	$22.9508
Labor, $11.00	5	hours					
Electricity, $3.50	10	acre-inch					
Maintenance and Repairs $50.00	1	x/acre					
Harvesting							
Hand Harvest Labor	45	hours	$495.00	$-	$-	$495.00	$81.1475
Harvest Aid	3	x/acre	$28.35	$38.96	$-	$67.31	$11.0344
Bin Trailer	3	x/acre	$28.29	$34.96	$-	$63.25	$10.3689
Fork Lift	3	x/acre	$28.29	$20.14	$-	$48.43	$7.9393
Truck	3	x/acre	$-	$11.18	$-	$11.18	$1.8328
Post-harvest							
Flail Crop Residue	1	x/acre	$3.05	$4.48	$-	$7.53	$1.2344
Soil Test	1	x/acre	$-	$-	$2.00	$2.00	$0.3279
Lime application, custom	0.25	x/acre	$-	$-	$75.00	$75.00	$12.2951
Other Costs							
Pickup and ATV	1	x/acre	$-	$5.31	$-	$5.31	$0.8705
Interest: operating capital	6	months	$-	$-	$62.99	$62.99	$10.3262
Total variable costs			$813.63	$169.03	$654.99	$1,637.65	$268.47
FIXED CASH COSTS					Unit	Total	Cost/Ton
Property insurance					acre	$25.00	$4.0984
Property taxes					acre	$20.00	$3.2787
Land Rent					acre	$200.00	$32.7869
Total fixed cash costs						$245.00	$40.1639
FIXED NON-CASH COSTS					Unit	Total	Cost/Ton
Machinery and equip - depreciation, interest and insurance					acre	$120.38	$19.7344
Pickup, truck and ATV - depreciation, interest and insurance					acre	$14.29	$2.3426
Total fixed non-cash costs						$134.67	$22.0770
Total fixed costs						$379.67	$62.2410
Total of all costs per acre						$2,017.32	$330.7082
Net projected returns						$2,527.18	$414.2918
U.S. Inflation Rate	2010	2019	15.43%			$2,917.12	$478.2170

Table 23: Spinach, enterprise budget.

Spinach, Conventional, Fresh Market, 2007, $/acre economic costs and returns							
GROSS INCOME			Quantity	Unit	$/Unit	Total	Price/Ton
Spinach			8.53	Ton	$824.59	7,033.75	$824.59
VARIABLE CASH COSTS	Description	Units	Labor	Machinery	Materials	Total	Cost/Ton
Field Preparations and Planting							
Deep chisel	0.5	x/acre	$3.64	$7.86	$-	$11.50	$1.35
Spray herbicide	1	x/acre	$3.05	$4.97	$12.00	$20.02	$2.35
Lime application, custom	0.25	x/acre	$-	$-	$75.00	$75.00	$8.79
Moldboard plow	1		$9.70	$22.19	$-	$31.89	$3.74
Disk	1	x/acre	$5.15	$11.22	$-	$16.37	$1.92
Soil analysis	1	x/acre	$1.77	$2.81	$100.00	$104.58	$12.26
Rotary till	1		$9.70	$25.18	$-	$34.88	$4.09
Seed Crop	1	x/acre	$11.78	$23.00	$400.00	$434.78	$50.97
Cultivating weeds	1.5		$15.46	$18.45	$-	$33.91	$3.98
Irrigation			$22.50	$-	$85.00	$107.50	$12.60
Labor, $7.50/set	3	sets					
Electricity, $10.00/set	3	sets					
Pipe rental, $110.00/acre	0.5	x/acre					
Spray insecticides	1	acre	$3.05	$4.97	$75.00	$83.02	$9.73
Spray herbicide	1	x/acre	$3.05	$4.97	$60.00	$68.02	$7.97
Harvesting							
Harvesting labor	750	carton	$1,500.00	$-	$-	$1,500.00	$175.85
Field Crates, 20/acre	0.5	x/acre	$-	$-	$6.50	$6.50	$0.76
Tractor and trailer	6.5	hours	$78.00	$125.74	$-	$203.74	$23.89
Packing and Materials							
Packing labor	750	carton	$750.00	$-	$-	$750.00	$87.92
Boxes	750	carton	$-	$-	$975.00	$975.00	$114.30
Hydro-cooling	750	carton	$-	$-	$75.00	$75.00	$8.79
Refrigeration	750	carton	$-	$-	$75.00	$75.00	$8.79
Delivery to market	750	carton	$60.00	$-	$-	$60.00	$7.03
Packing Shed and Equipment	750	carton	$-	$-	$112.50	$112.50	$13.19
Other Costs							
Pickups, truck and ATV	0.5	x/acre	$-	$103.38	$-	$103.38	$12.12
Shop and machine shed	0.5	x/acre	$-	$-	$4.00	$4.00	$0.47
Miscellaneous and overhead	0.5	x/acre	$-	$-	$37.50	$37.50	$4.40
Interest: operating capital	6	months	$-	$-	$98.48	$98.48	$11.55
Total variable costs			$2,476.85	$354.74	$2,190.98	$5,022.57	$588.81
FIXED CASH COSTS					Unit	Total	Cost/Ton
Property insurance					acre	$25.00	$2.93
Property taxes					acre	$20.00	$2.34
Land Rent					acre	$200.00	$23.45
Field sanitation equipment					acre	$15.00	
Total fixed cash costs						$260.00	$30.48
FIXED NON-CASH COSTS					Unit	Total	Cost/Ton
Machinery and equip - depreciation, interest and insurance					acre	$92.11	$10.80
Pickup, truck and ATV - depreciation, interest and insurance					acre	$47.77	$5.60
Shop and machine shed - depreciation, interest and insurance					acre	$10.88	$1.28
Total fixed non-cash costs						$150.76	$17.67
Total fixed costs						$410.76	$48.15
Total of all costs per acre						$5,443.33	$636.97
Net projected returns						$1,600.42	$187.62
U.S. Inflation Rate	2007	2019	21.40%			$1,942.91	$227.7741

APPENDIX C – A small CRP database

This database is a small version of the database presented in Appendix A. We have selected a few index and presented a new group in Tables 24, 25 and 26. Previous tables have already detailed other croping traits.

Table 24: Crop's attributes: seeding and harvesting.

Index	Crops	Seeding - Begin	Seeding - End	Harvesting - Begin	Harvesting - End	Cycle (Days)
1	Leaf Lettuce for Fresh Market, Winter	1-Sep	31-Jan	1-Nov	30-Apr	60
2	Sweet corn, Winter	1-Feb	31-May	15-Jun	10-Sep	120
3	Broccoli for Fresh Market, Fall	1-Sep	30-Nov	15-Oct	15-Dec	120
4	Tomato for Fresh Market, Spring	15-Jan	31-May	1-May	31-Jul	120
5	Spinach for Fresh Market, Fall	10-Aug	2-Sep	15-Sep	24-Dec	60
6	Summer Squash for Fresh Market	5-May	15-Jul	1-Jul	31-Oct	60
21	Carrots for Fresh Market, Summer	1-Dec	31-Mar	1-May	31-Jul	150
7	Potatoes for Fresh Market, Summer	8-Apr	27-May	6-Aug	4-Oct	120
8	Cucumber for Fresh Market, Fall	1-Jun	15-Jul	1-Aug	30-Sep	60
9	Spring Onions for Fresh Market	1-Oct	31-Dec	1-May	31-Jul	120
10	Strawberries for Fresh Market, Winter	20-Sep	10-Nov	25-Nov	25-Apr	60
11	Bell Peppers for Fresh Market and Processing, Winter	15-Sep	15-Dec	1-Jan	31-Mar	120
12	Annual rye-grass, Early Spring	15-Mar	1-May	20-Apr	15-May	300
13	Barley, Winter	1-Dec	25-Jan	4-Jul	20-Jul	120
14	Buckwheat	20-May	30-Aug	4-Jul	15-Oct	60
15	Sorghum-sudangrass, Late Spring, Summer	15-May	31-Jul	20-Sep	30-Nov	90
17	Rapeseed (or Canola), Fall	30-Sep	30-Nov	20-Apr	15-May	270
18	Rapeseed (or Canola), Spring	20-Mar	31-May	20-Apr	15-May	270
19	Crimson clover, Winter	5-Feb	20-Mar	5-Feb	20-Mar	180
20	Field peas, Fall	15-Sep	20-Nov	25-Apr	10-May	240
21	Hairy vetch, Fall	20-Aug	10-Sep	15-May	31-May	270
22	Red clover, Late Summer	20-Aug	30-Sep	20-Apr	10-Jul	360
23	White clover, Late Winter, Spring	5-Feb	10-May	20-Apr	15-May	240

Table 25: Crop's attributes: profit and nutrient demand.

Index	**Crops**	**Family**	**Typical yield (Unit/acre)**	**Unit**	**Profit ($/Unit)**	**N (kg/acre)**	**P (kg/acre)**	**K (kg/acre)**
1	Leaf Lettuce for Fresh Market, Winter	Lettuce	569	carton	5.712	27.22	13.06	54.43
2	Sweet corn, Winter	Grass	12	ton	26.736	24.95	3.49	12.47
3	Broccoli for Fresh Market, Fall	Mustard	6.1	ton	478.217	12.7	3.63	18.14
4	Tomato for Fresh Market, Spring	Nightshade	1500	box	7.111	15.42	7.26	27.22
5	Spinach for Fresh Market, Fall	Beet	8.53	ton	227.774	18.14	3.63	18.14
6	Summer Squash for Fresh Market	Cucurbit	176	cwt	9.796	13.88	3.08	0
7	Carrots for Fresh Market, Summer	Carrot	430	cwt	3.403	18.87	4.72	47.17
8	Potatoes for Fresh Market, Summer	Nightshade	450	cwt	-0.812	50.35	9.53	68.04
8	Cucumber for Fresh Market, Fall	Cucurbit	161	cwt	-12.15	28.12	9.98	47.17
10	Spring Onions for Fresh Market	Lily	27.6	ton	17.18	30.62	5.67	30.62
11	Strawberries for Fresh Market, Winter	Rose	10192	lbs	0.097	4.54	1.36	9.98
12	Bell Peppers for Fresh Market and Processing, Winter	Nightshade	1280	carton	3.139	18.14	2.72	20.87
13	Annual rye-grass, Early Spring	Grass	1		0	0	0	0
14	Barley, Winter	Grass	1		0	0	0	0
15	Buckwheat	Buckwheat	1		0	0	0	0
16	Sorghum-sudangrass, Late Spring, Summer	Grass	1		0	0	0	0
17	Rapeseed (or Canola), Fall	Mustard	1		0	0	0	0
18	Rapeseed (or Canola), Spring	Mustard	1		0	0	0	0
19	Crimson clover, Winter	Legume	1		0	0	0	0
20	Field peas, Fall	Legume	1		0	0	0	0
21	Hairy vetch, Fall	Legume	1		0	0	0	0
22	Red clover, Late Summer	Legume	1		0	0	0	0
23	White clover, Late Winter, Spring	Legume	1		0	0	0	0

Table 26: Cover crop's performance and roles.

Index	Crops	Average Total N (lb./A)	Average Dry Matter (lb./A/yr.)	Dry matter	N Scav-enger	Soil Builder	Erosion Fighter	Weed Fighter	Good Grazing	Quick Growth	Lasting Residue	Duration	Harvest Value - Forage	Harvest Value - Seed	Cash Crop Inter-seed
13	Annual rye-grass, Early Spring	0	5500	2.4	3	3	3	3	3	3	3	3	2	1	4
14	Barley, Winter	0	6000	2.7	3	3	4	4	3	3	4	2	3	2	3
15	Buckwheat	0	3000	1.3	0	2	1	1	3	4	0	1	0	1	3
16	Sorghum-sudangrass, Late Spring, Summer	0	9000	4.0	4	4	4	4	2	4	3	4	4	0	0
17	Rapeseed (or Canola), Fall	100	3500	1.6	3	2	3	3	4	3	2	3	1	4	0
18	Rapeseed (or Canola), Spring	100	3500	1.6	3	2	3	3	4	3	2	3	1	4	0
19	Crimson clover, Winter	100	4500	2.0	2	3	3	3	3	2	2	1	4	3	4
20	Field peas, Fall	120	4500	2.0	1	2	3	3	2	3	1	2	4	3	4
21	Hairy vetch, Fall	145	3650	1.6	1	3	2	2	2	1	1	3	1	3	2
22	Red clover, Late Summer	110	3500	1.6	2	3	2	2	3	1	1	2	4	3	4
23	White clover, Late Winter, Spring	140	4000	1.8	1	2	3	3	3	1	1	4	3	2	3

APPENDIX D – Configuration settings

Computational tests in this research have been developed in the machine described in Table 27.

Table 27: System information: summary.

OS Name	Microsoft Windows 10 Home Single Language
Version	10.0.18362 Build 18362
OS Manufacturer	Microsoft Corporation
System Name	DESKTOP-7P9ASJD
System Manufacturer	LENOVO
System Model	80YM
System Type	x64-based PC
System SKU	LENOVO_MT_80YM_BU_idea_FM_Lenovo YOGA 520-14IKB
Processor	Intel(R) Core(TM) i7-7500U CPU @ 2.70GHz, 2901 Mhz, 2 Core(s), 4 Logical Processor(s)
BIOS Version/Date	LENOVO 4QCN36WW(V1.06), 11/07/2017
SMBIOS Version	3
Embedded Controller Version	1.36
BIOS Mode	UEFI
BaseBoard Manufacturer	LENOVO
BaseBoard Product	Lenovo YOGA 520-14IKB
BaseBoard Version	SDK0J40688 WIN
Platform Role	Mobile
Secure Boot State	Off
Locale	United States
Hardware Abstraction Layer	Version = "10.0.18362.628"
User Name	DESKTOP-7P9ASJD
Time Zone	E. South America Standard Time
Installed Physical Memory (RAM)	8,00 GB
Total Physical Memory	7,86 GB
Available Physical Memory	3,16 GB
Total Virtual Memory	9,11 GB
Available Virtual Memory	3,47 GB
Page File Space	1,25 GB
Kernel DMA Protection	Off
Virtualization-based security	Not enabled
Device Encryption Support	Elevation Required to View
Hyper-V - VM Monitor Mode Extensions	Yes
Hyper-V - Second Level Address Translation Extensions	Yes
Hyper-V - Virtualization Enabled in Firmware	Yes
Hyper-V - Data Execution Protection	Yes

The deterministic techniques in this research have been programmed in IBM ILOG CPLEX Optimization Studio, which supports Optimization Programming Language (OPL). Its version is **12.8.0.0**. IBM ILOG CPLEX is one of the most used large-scale solver. It is a robust and efficient optimization solver.

The proposed genetic algorithms (PLGA and PLMGA) have been written in **C programming language**. An Integrated Development Environment (IDE) has been used to code the algorithms called Code::Blocks, which is a free license C, C++ and Fortran IDE, which is very extensible and fully configurable. Its version is **Release 17.12 rev 11256**.