

**UNIVERSIDADE FEDERAL DO PAMPA  
BACHARELADO EM ENGENHARIA DE COMPUTAÇÃO**

**ERIC DIAS DA SILVA ROSSO**

**ALGORITMO PARA MINIMIZAÇÃO DA  
ENDOGAMIA E MAXIMIZAÇÃO DE  
ÍNDICE ECONÔMICO EM SISTEMAS  
DE ACASALAMENTO**

**Bagé  
2023**

**ERIC DIAS DA SILVA ROSSO**

**ALGORITMO PARA MINIMIZAÇÃO DA  
ENDOGAMIA E MAXIMIZAÇÃO DE  
ÍNDICE ECONÔMICO EM SISTEMAS  
DE ACASALAMENTO**

Trabalho de Conclusão de Curso apresentado ao curso de Bacharelado em Engenharia de Computação como requisito parcial para a obtenção do grau de Bacharel em Engenharia de Computação.

Orientadora: Ana Paula Lüdtke Ferreira

**Bagé  
2023**

Ficha catalográfica elaborada automaticamente com os dados fornecidos pelo(a) autor(a) através do Módulo de Biblioteca do Sistema GURI (Gestão Unificada de Recursos Institucionais).

D541a Rosso, Eric Dias da Silva

Algoritmo para minimização da endogamia e maximização de índice econômico em sistemas de acasalamento / Eric Dias da Silva Rosso.

- 2023.

116 f.: il.

Orientadora: Ana Paula Lüdtke Ferreira

Trabalho de Conclusão de Curso (Graduação)  
- Universidade Federal do Pampa, Campus Bagé,  
Bacharelado em Engenharia de Computação, 2023.

1. Melhoramento genético. 2. Seleção de acasalamentos. 3. Banco de dados baseado em grafos. 4. Otimização combinatória. I. Ana Paula Lüdtke Ferreira. II. Título.



SERVIÇO PÚBLICO FEDERAL  
MINISTÉRIO DA EDUCAÇÃO  
Universidade Federal do Pampa

**ÉRIC DIAS DA SILVA ROSSO**

**ALGORITMO PARA MINIMIZAÇÃO DA ENDOGAMIA E MAXIMIZAÇÃO DE ÍNDICE  
ECONÔMICO EM SISTEMAS DE ACASALAMENTO**

Trabalho de Conclusão de Curso apresentado ao Curso de Bacharelado em Engenharia de Computação da Universidade Federal do Pampa, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Computação.

Trabalho de Conclusão de Curso defendido e aprovado em: 31 de Janeiro de 2023.

Banca examinadora:

---

Profa. Dra. Ana Paula Lüdtke Ferreira  
Orientadora  
UNIPAMPA

---

Prof. Dr. Carlos Michel Betemps  
UNIPAMPA

---

Prof. Dr. Julio Saraçol Domingues Jr.  
UNIPAMPA



---

Assinado eletronicamente por **CARLOS MICHEL BETEMPS, PROFESSOR DO MAGISTERIO SUPERIOR**, em 07/02/2023, às 21:28, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.

---



Assinado eletronicamente por **ANA PAULA LUDTKE FERREIRA, PROFESSOR DO MAGISTERIO SUPERIOR**, em 07/02/2023, às 22:05, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.

---



Assinado eletronicamente por **ERIC DIAS DA SILVA ROSSO, Aluno**, em 07/02/2023, às 22:12, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.

---



Assinado eletronicamente por **JULIO SARACOL DOMINGUES JUNIOR, PROFESSOR DO MAGISTERIO SUPERIOR**, em 07/02/2023, às 23:54, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.

---



A autenticidade deste documento pode ser conferida no site [https://sei.unipampa.edu.br/sei/controlador\\_externo.php?acao=documento\\_conferir&id\\_orgao\\_acesso\\_externo=0](https://sei.unipampa.edu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0), informando o código verificador **1049211** e o código CRC **FDFAF324**.

---

Referência: Processo nº 23100.002573/2023-40 SEI nº 1049211

## **AGRADECIMENTOS**

Agradeço a Deus por me guiar e me dar forças em todos os momentos. Minha gratidão também vai para minha mãe, por todo amor, apoio e incentivo que me deu durante todo o processo.

Gostaria de agradecer especialmente a Guilherme, Ignácio, Kelvin, Maria Elizabeth, Michel e Thiago, meus amigos mais próximos, por sempre estarem presentes, me ajudando a superar as dificuldades e contribuindo com discussões e trocas de conhecimento ao longo da jornada.

Agradeço também à Embrapa Pecuária Sul, por me proporcionar a oportunidade de estagiar na instituição, o que foi fundamental para o meu crescimento profissional e pessoal.

Não posso deixar de mencionar minha orientadora, que foi fundamental na orientação e acompanhamento do meu trabalho, sempre disponível para esclarecer minhas dúvidas e me auxiliar na elaboração deste trabalho. Agradeço sua paciência e dedicação em guiar-me durante todo o processo.

Agradeço também à minha banca examinadora, por suas valiosas sugestões e contribuições para aperfeiçoamento do meu trabalho.

Por fim, agradeço aos meus professores, por todo conhecimento e ensinamentos que me proporcionaram, me preparando para este momento.

Agradeço a todos que de alguma forma contribuíram para a conclusão deste trabalho.

## RESUMO

Os processos de melhoramento animal buscam o aumento da qualidade dos rebanhos e da lucratividade dos produtores rurais. Os sistemas de seleção e de acasalamento formam o cerne dos sistemas de melhoramento, que visam direcionar as características genéticas e fenotípicas da prole por meio da combinação das características dos pais. Os processos de seleção e acasalamento devem levar em consideração a coancestralidade entre os animais do rebanho, evitando o cruzamento de animais aparentados de forma a minimizar a consanguinidade. A consanguinidade ocorre a partir de acasalamentos entre indivíduos aparentados, que pode ser útil para a fixação de padrões mas que geram problemas de saúde, menor rusticidade e menor valor genético o que, por consequência, traz prejuízo ao produtor. Este trabalho propõe encontrar uma solução algorítmica para o processo de melhoramento animal com o objetivo de minimizar a coancestralidade enquanto maximiza o índice econômico, buscando um balanceamento entre esses valores por meio de uma ponderação que é parâmetro do algoritmo. O algoritmo desenvolvido visa garantir um esquema de acasalamento ótimo, considerando os aspectos mais importantes e desejados dentro do escopo do melhoramento animal. Para isso, o problema foi modelado como um problema de programação linear binária. Para tratá-lo é utilizada a combinação das técnicas *simplex*, para solução do problema de programação linear relaxado e *branch-and-bound*, para encontrar a melhor solução inteira. O algoritmo denominado CoancestryMIN-Selection está concluído e futuramente poderá ser integrado com o banco de dados baseado em grafos já desenvolvido. Ele foi criado nas linguagens R e C++ com a IDE RStudio. O algoritmo desenvolvido é capaz de realizar a computação da matriz de pedigree do rebanho, assegurando sempre uma solução ótima para o problema em questão. Análises realizadas a partir dos testes apontam que a minimização da consanguinidade é uma questão de complexidade superior à maximização do índice econômico.

**Palavras-chave:** Melhoramento genético; Seleção de acasalamentos; Banco de dados baseado em grafos; Otimização combinatória.

## ABSTRACT

Animal breeding processes seek to increase the quality of the herds and the profitability of rural producers. Selection and mating systems form the core of breeding systems, aiming to target the offspring's genetic and phenotypic traits through the combination of parental traits. The selection and mating processes must consider the coancestry between the animals in the herd, avoiding the crossing of related animals to minimize inbreeding. Inbreeding occurs from matings between related individuals, which can be useful for setting standards but generate health problems, less rusticity, and lower genetic value, consequently harming the producer. This work proposes an algorithmic solution for the animal breeding process to minimize coancestry while maximizing the economic index, seeking a balance between these values through a weighting parameter of the algorithm. The developed algorithm aims to guarantee an optimal mating scheme, considering the most important and desired aspects within the scope of animal breeding. For this, the problem was modeled as a binary linear programming problem. To treat it, a combination of simplex techniques is used to solve the relaxed linear programming problem and branch-and-bound to find the best integer solution. The algorithm called CoancestryMIN-Selection is finished and in the future it may be integrated with the already developed graph-based database. It was created in the R and C++ languages with the RStudio IDE. The developed algorithm is able to perform the computation of the pedigree matrix of the herd, always ensuring an optimal solution for the problem in question. Analyzes carried out based on the tests indicate that the minimization of inbreeding is a matter of greater complexity than the maximization of the economic index.

**Keywords:** Animal breeding. Combinatorial optimization. Graph based database. Mating selection.



## LISTA DE FIGURAS

Figura 1	Extrato da base de dados .....	32
Figura 2	Ilustração da Equação (1) para $x \neq y$ .....	47
Figura 3	Ilustração da Equação (2) .....	48
Figura 4	Ilustração da Equação (1) dentro da Equação (2) .....	48
Figura 5	Ilustração da Equação (3) .....	50
Figura 6	Esquema de entradas e saída do algoritmo CoancestryMIN-Selection .....	60
Figura 7	Fluxograma do algoritmo CoancestryMIN-Selection .....	63
Figura 8	Exemplo da saída gerada pelo arquivo do algoritmo de testes .....	65
Figura 9	Resultados e parâmetros dos testes variando n° de fêmeas .....	66
Figura 10	Resultados e parâmetros dos testes variando n° de machos.....	67
Figura 11	Resultados e parâmetros dos testes variando n° de fêmeas .....	67
Figura 12	Resultados e parâmetros dos testes variando n° de machos.....	68
Figura 13	Modelo de dados.....	71
Figura 14	Recorte do banco de dados populado com dados da EMBRAPA .....	72
Figura 15	Amostra do banco de dados - nodos do tipo “Animal” e “Safra” .....	73

## LISTA DE TABELAS

Tabela 1	Termos de pesquisa para a revisão da literatura .....	25
Tabela 2	Resultados da aplicação do protocolo .....	27
Tabela 3	Técnicas utilizadas.....	44
Tabela 4	Índice econômico de ciclo completo (IEEC).....	49
Tabela 5	Exemplo do problema de acasalamento - índice de coancestralidade.....	55
Tabela 6	Exemplo do problema de acasalamento - coeficiente de consanguinidade .....	55
Tabela 7	Exemplo do problema de acasalamento - IECC .....	55
Tabela 8	Exemplo do problema de acasalamento - média ponderada de melhoramento	56
Tabela 9	Solução do exemplo do problema de acasalamento .....	59

## LISTA DE DEFINIÇÕES

Definição 1 Coancestralidade .....	46
Definição 2 Consanguinidade .....	47
Definição 3 Índice de Seleção.....	48
Definição 4 Índice de Coancestralidade.....	49
Definição 5 Contribuição Individual.....	50
Definição 6 Contribuição de acasalamento.....	51
Definição 7 Contribuição ponderada de melhoramento.....	51
Definição 8 Problema de minimização da contribuição ponderada de melhoramento ....	52

## LISTA DE ABREVIATURAS E SIGLAS

ABNT	Associação Brasileira de Normas Técnicas
APR	<i>Angus Performance Register</i>
BLUP	<i>Best Linear Unbiased Prediction</i>
EBV	<i>Estimated Breeding Values</i>
EMBRAPA	Empresa Brasileira de Pesquisa Agropecuária
IECC	Índice Econômico de Ciclo Completo
UNIPAMPA	Universidade Federal do Pampa
USDA	<i>United States Department of Agriculture</i>

## SUMÁRIO

<b>1 INTRODUÇÃO</b> .....	<b>13</b>
1.1 Processos e problemas do melhoramento animal.....	13
1.2 Objetivos .....	14
<b>2 REFERENCIAL TEÓRICO</b> .....	<b>17</b>
2.1 Sistemas de acasalamento.....	17
2.2 Consanguinidade, coancestralidade e valor econômico .....	18
2.3 Programação linear e o método Simplex .....	20
2.4 Programação inteira e método <i>branch-and-bound</i> .....	22
<b>3 MATERIAL E MÉTODOS</b> .....	<b>24</b>
3.1 Caracterização e fases da pesquisa.....	24
3.2 Revisão da literatura.....	24
3.3 Ferramental tecnológico .....	28
3.4 Fontes de dados .....	31
<b>4 TRABALHOS RELACIONADOS</b> .....	<b>33</b>
4.1 Abordagens para minimização da coancestralidade .....	33
4.2 Formalização do problema.....	43
<b>5 O ALGORITMO COANCESTRYMIN-SELECTION</b> .....	<b>46</b>
5.1 Caracterização do problema .....	46
5.2 Modelo de programação linear/inteira .....	53
5.3 Codificação da solução.....	60
<b>6 RESULTADOS E DISCUSSÃO</b> .....	<b>64</b>
<b>7 MODELO DE DADOS</b> .....	<b>69</b>
7.1 Banco de dados baseado em grafos .....	69
7.2 Modelo do banco de melhoramento .....	70
<b>8 CONSIDERAÇÕES FINAIS</b> .....	<b>74</b>
8.1 Conclusões .....	74
8.2 Trabalhos futuros .....	75
<b>REFERÊNCIAS</b> .....	<b>77</b>
<b>APÊNDICE A – CÓDIGO-FONTE DOS PROGRAMAS DESENVOLVIDOS</b> .....	<b>80</b>
<b>APÊNDICE B – REQUISIÇÕES DO BANCO DE DADOS DESENVOLVIDO</b> .....	<b>99</b>
<b>APÊNDICE C – ALGORITMO DE TESTES</b> .....	<b>109</b>

# 1 INTRODUÇÃO

## 1.1 Processos e problemas do melhoramento animal

Processos de melhoramento animal dependem de duas estratégias: a seleção e o sistema de acasalamento. No processo de seleção, os melhores animais são escolhidos para se tornarem pais da nova geração do rebanho. Essa escolha pode ser feita de modo empírico ou usando um índice de seleção que favoreça características desejáveis e desfavoreça problemas na próxima geração (MOTTA, 2021). O sistema de acasalamento consiste na estratégia de escolha dos pares com vistas a maximizar o índice esperado da futura prole, com produção de animais excepcionais (acasalamento entre iguais) ou com vistas a uma maior homogeneidade das características do rebanho (acasalamento compensatório).

Ferreira, Yokoo e Motta (2021) apresentaram um algoritmo polinomial ótimo para solucionar o problema da maximização do índice esperado do rebanho que foi posteriormente usado para mostrar que o problema do acasalamento compensatório também tem solução polinomial, mantendo a otimalidade da solução (FERREIRA, 2021). Ambos os algoritmos, contudo, usam uma estratégia empírica para definir o número máximo de vezes que um reprodutor pode acasalar: esse valor é parâmetro do algoritmo e não explicitamente calculado.

A restrição no número de vezes que um reprodutor pode acasalar tem o papel de restringir a consanguinidade da próxima geração do rebanho. O valor máximo da prole, dado um grupo de matrizes e reprodutores, consistiria em cruzar todas as fêmeas com o melhor dos machos, conforme demonstrado em (FERREIRA; YOKOO; MOTTA, 2021). Essa escolha, contudo, faria com que todos os animais da próxima geração sejam irmãos. Cruzamentos consanguíneos são usados para fixação de fenótipos em animais, mas devem ser evitados, visto que o aumento da endogamia causa redução no desempenho de características quantitativas, incluindo reprodução e sobrevivência (MI *et al.*, 1965, CHARLESWORTH AND WILLIS, 2009 *apud* FERREIRA, 2021, p. 2).

Embora a existência de relações de parentesco entre animais possa ser usada como restrição, sendo eliminada ou penalizada pelos algoritmos de escolha de acasalamentos, o ideal é que o esquema de cruzamentos não envolva somente o valor do índice, mas também a possibilidade de novos acasalamentos a longo prazo. Ou seja, deve ser possível fazer uma escolha de acasalamentos de maneira que as opções de seleção e cruzamentos

sejam mantidas, sem necessidade de compra de novos animais ou sêmen como forma de aumentar a variabilidade genética.

Ainda que seja inevitável a ocorrência de consanguinidade em um rebanho fechado, ou seja, no qual não haja a inserção de novos animais, é necessário que haja um controle das taxas de consanguinidade, para que tanto a saúde do rebanho quanto a lucratividade do produtor não sejam comprometidas. Diversos sistemas de seleção e acasalamento foram desenvolvidos ao longo do tempo, com vistas a buscar meios de controlar as taxas de consanguinidade a longo prazo. A consanguinidade de cada indivíduo é calculada com base na coancestralidade entre seus pais (VLECK, 1993). Quanto mais próximo o parentesco entre os pais, maior será a consanguinidade do indivíduo. A coancestralidade é determinada usando informações de pedigree de cada indivíduo, que mostram as relações de parentesco com outros animais no rebanho. Para controlar a consanguinidade média do rebanho é necessário evitar que as novas gerações possuam muitos ancestrais comuns com o restante do rebanho, isto significa, controlar sua coancestralidade. Caso este aspecto não seja tratado com a devida consideração, abre-se margem para ocorrência de depressão por consanguinidade (*inbreeding depression*). A consanguinidade eleva a quantidade de homozigotos e diminui a de heterozigotos, reduz o valor de combinação gênica, não só diminuindo o valor genotípico como apresentando problemas de saúde como baixa fertilidade, agressividade, perda de rusticidade, entre outros problemas (ELER, 2017).

## 1.2 Objetivos

O objetivo deste trabalho é construir um algoritmo para seleção de acasalamentos com vistas à minimização da coancestralidade entre os animais do rebanho e a maximização do lucro financeiro para o sistema produtivo. O algoritmo deverá ser usado na fase de seleção dos animais que devem acasalar, de forma a determinar quais são os melhores pais para a próxima geração do rebanho e como eles devem ser agrupados, de forma que tanto o valor médio da prole seja maximizado quanto o índice de coancestralidade do futuro rebanho seja minimizado.

São objetivos específicos deste trabalho:

1. Investigar quais são as técnicas usadas para minimização da coancestralidade de animais de um rebanho e qual o grau de formalização do problema existente na

literatura.

2. Definir formalmente o problema da minimização da coancestralidade e investigar quais são os problemas computacionais similares e suas respectivas complexidades.
3. Construir uma estratégia de minimização de coancestralidade que possa ser mantida ao longo do tempo, sem prejuízo dos valores das proles obtidas nos sucessivos processos de seleção e acasalamentos, além de maximizar o índice econômico.

O Capítulo 2 busca introduzir o leitor aos conceitos que serão discutidos de maneira mais aprofundada, nos capítulos seguintes. Na Seção 2.1 são discutidos os diferentes tipos de acasalamento que são praticados e quais suas finalidades. Já na Seção 2.2 é discutida a importância de aspectos como coancestralidade e consanguinidade média de um rebanho, além de buscar explicar suas diferenças. Na Seção 2.2 é discutido o índice econômico que é utilizado neste trabalho e como foi desenvolvido. Na Seção 2.3 são discutidos os conceitos de programação linear e aplicabilidade do método Simplex. Na Seção 2.4 são discutidos os conceitos de programação inteira e aplicabilidade do método *branch-and-bound* dentro do contexto, sendo a programação inteira um caso específico de programação linear.

O Capítulo 3 trata da metodologia de pesquisa utilizada para encontrar os resultados mais relevantes relacionados ao problema discutido, o estado da arte no tema e os trabalhos correlatos. Na Seção 3.1 são definidas as fases da pesquisa e como ela é caracterizada. A Seção 3.2 apresenta o protocolo de revisão utilizado e os resultados obtidos a partir dele. Na Seção 3.3 está listado o ferramental tecnológico utilizado no desenvolvimento do algoritmo proposto por este trabalho. Na Seção 3.4 é discutida a procedência dos dados que estão sendo utilizados no desenvolvimento do algoritmo proposto por este trabalho.

O Capítulo 4 traz os resultados apurados pelo método de pesquisa, apresentando o estado da arte sobre abordagens técnicas e demais temas das questões de pesquisa da revisão bibliográfica. Especificamente, na Seção 4.1 são descritos as principais abordagens e métodos utilizados para minimização da coancestralidade em animais; na Seção 4.2 é apresentada a classificação dos trabalhos correlatos frente ao seu nível de formalização do problema.

O Capítulo 5 traz consigo a caracterização do problema abordado com suas definições, além da exemplificação do problema. Na Seção 5.1 são definidos coancestralidade, consanguinidade, índice de seleção, além de outros conceitos. A Seção 5.2 busca ilustrar o problema que está sendo tratado, através de uma



exemplificação. Na Seção 5.3 é feita uma breve descrição do funcionamento do algoritmo desenvolvido, que pode ser melhor visualizado no Apêndice A.

No Capítulo 6, são expostos os resultados alcançados por meio de diversos experimentos. É descrito o método de experimentação, além de apresentadas as tabelas e gráficos que ilustram esses resultados e os parâmetros empregados em cada experimento, permitindo assim uma melhor compreensão dos resultados obtidos.

O Capítulo 7 traz à tona a ideia de um banco de dados criado propriamente para auxiliar nos problemas encontrados no armazenamento de dados de animais. Os conceitos que permeiam um banco de dados baseado em grafos são apresentados na Seção 7.1. Na Seção 7.2 é apresentado o modelo de banco de dados baseado em grafos desenvolvido propriamente para lidar com dados relacionados ao *pedigree* dos animais. Também são informadas quais tecnologias foram utilizadas neste processo.

O Capítulo 8 traz as considerações finais a respeito do que foi pesquisado, encontrado e desenvolvido, além de uma estimativa de trabalhos futuros. Na Seção 8.1 são discutidos os resultados de pesquisa e o que foi desenvolvido ao longo deste trabalho. Os trabalhos futuros que poderão vir a ser desenvolvidos são apresentados na Seção 8.2.

## 2 REFERENCIAL TEÓRICO

### 2.1 Sistemas de acasalamento

Os sistemas de acasalamento são desenvolvidos com o objetivo de escolher quem serão os pais da próxima geração do rebanho. Geralmente, os sistemas de acasalamento buscam ganhos genéticos ou fenotípicos, visando aumento da lucratividade e da saúde dos animais (ELER, 2017; SIMÕES et al., 2020).

No ponto de vista do desempenho dos animais, há três tipos de acasalamentos que podem ser implementados: o acasalamento aleatório, que parte da premissa que qualquer indivíduo tem chance igual de acasalar e deixar descendentes; o acasalamento entre semelhantes, que busca acasalar indivíduos com valor genético aproximado, fazendo com que os acasalamentos ocorram entre os melhores machos e as melhores fêmeas, restando os piores machos para as piores fêmeas, gerando um rebanho heterogêneo em termos das características avaliadas; e o acasalamento entre dessemelhantes ou compensatório, que resulta no acasalamento dos melhores machos com as piores fêmeas e das melhores fêmeas com os piores machos, proporcionando um rebanho mais homogêneo. O acasalamento entre semelhantes abre possibilidade para obtenção de animais com alto mérito genético, algo que dificilmente ocorre em acasalamentos entre dessemelhantes; por outro lado, o acasalamento entre semelhantes tende a gerar um alto grau de consanguinidade no rebanho, pelo fato de que os melhores animais são frequentemente aparentados (ELER, 2017).

O *pedigree* informa a genealogia de um animal de raça, o que atesta sua ancestralidade, de forma similar a uma certidão de nascimento humana. Em relação ao *pedigree* do rebanho, existem acasalamentos endogâmicos, exogâmicos e cruzamentos entre raças distintas. O acasalamento endogâmico consiste em acasalar indivíduos aparentados, geralmente buscando a fixação de alguma característica fenotípica no rebanho. Esse tipo de acasalamento causa um aumento da geração de pares de genes homozigotos e diminuição dos pares heterozigotos. A palavra heterozigoto provém do grego, da junção das palavras *héteros* (diferente) e *zygotos* (junto). O conceito de heterozigoto é retratar indivíduos formados a partir de genes diferentes. A palavra homozigoto também provém do grego formada a partir da junção das palavras *homo* (igual) e *zygotos* (junto). O conceito de homozigoto é retratar indivíduos formados a partir de genes idênticos. O acasalamento exogâmico, ocorre de maneira inversa ao

endogâmico, pois consiste em acasalar indivíduos que não possuem parentesco. Em geral, os acasalamentos buscam o aumento da heterozigose, do valor da combinação gênica e do valor genotípico. O cruzamento entre duas raças diferentes, desde que possuam características similares, está dentro desse contexto (ELER, 2017).

Um plano completo de acasalamento deve contemplar ambos conceitos de desempenho e *pedigree*. É desejável que as futuras proles possuam baixo parentesco com os demais animais do rebanho, uma baixa consanguinidade e um alto valor genético. O uso de informações do *pedigree* é fundamental para qualquer sistema de melhoramento, tendo em vista o controle da consanguinidade do rebanho.

## 2.2 Consanguinidade, coancestralidade e valor econômico

Nos diferentes sistemas de acasalamento e na busca por melhorias genéticas, existem os fatores coancestralidade (coeficiente de parentesco) e consanguinidade que devem ser levados em consideração. A coancestralidade é uma relação binária entre dois indivíduos, responsável por representar o grau de parentesco entre eles. A consanguinidade é uma relação unária (ou predicado), que consiste em um valor atribuído a um único indivíduo, decorrente das relações de parentesco entre seus ancestrais.

A consanguinidade de cada indivíduo é calculada a partir da coancestralidade entre seus pais (VLECK, 1993), e assim recursivamente. Quanto maior o grau de parentesco entre seus pais, maior será sua consanguinidade. A coancestralidade, por sua vez, é calculada utilizando as informações disponibilizadas pelo *pedigree* de cada indivíduo, que estabelecem as relações de parentesco entre ele e os demais animais do rebanho. As informações que constam no *pedigree* permitem evitar a cruza entre irmãos, meio-irmãos e primos. Porém, não é possível evitar que em algum momento haja parentesco entre todos os animais de um rebanho fechado: após sucessivas gerações todos os animais terão pelo menos um ancestral comum. Um rebanho fechado consiste de um rebanho em que não existe a inserção de novos animais.

É desejável que a consanguinidade média do rebanho seja controlada, principalmente quando se trata de um rebanho fechado, visto que o aumento demasiado da consanguinidade média pode causar uma depressão por consanguinidade (*inbreeding depression*). A depressão por consanguinidade acarreta diversos problemas de saúde e perda nos valores genéticos do rebanho, trazendo prejuízos ao produtor (MI *et al.*, 1965, CHARLESWORTH AND WILLIS, 2009 *apud* FERREIRA, 2021, p. 2).

Controlar somente a consanguinidade da possível prole que irá compor a próxima geração pode aumentar a coancestralidade entre os animais do rebanho, o que futuramente traria um aumento indesejado da consanguinidade. Um exemplo disso é o caso definido em Ferreira, Yokoo e Motta (2021): ao considerar um índice que se queira otimizar, é mostrado que o melhor acasalamento possível ocorreria ao utilizar o macho melhor classificado para acasalar com todas as fêmeas. Visto que em um processo que se atenta somente para a consanguinidade da prole que seria gerada, este seria um acasalamento hipoteticamente possível. Levando em consideração que este macho não possui parentesco com nenhuma das fêmeas, é esperado que a prole possua uma consanguinidade nula. O problema se dá no fato de que essa nova geração seria inteiramente aparentada, pois todos os novos animais seriam filhos de um mesmo pai. A nova geração não só possuiria um ancestral comum, mas também um elevado valor de coancestralidade entre si. Isto impossibilitaria novos acasalamentos sem que se utilize novos animais.

Com base nas premissas, é possível criar um sistema de acasalamento para controlar tanto a coancestralidade quanto a consanguinidade média do rebanho, buscando atingir valores aceitáveis para ambos. Isto é, acasalamentos que não prejudiquem o ganho genético ao mesmo tempo que não haja perdas por depressão endogâmica. Em ambos os casos, a lucratividade do produtor pode ser afetada negativamente, sendo necessário encontrar um equilíbrio entre as restrições do acasalamento. Minimizar as taxas de coancestralidade média do rebanho significa controlar a ocorrência de ancestrais comuns nas próximas gerações.

Embora o controle da coancestralidade – e correspondente controle da consanguinidade – sejam aspectos importantes da manutenção de um rebanho saudável e produtivo, não são os únicos aspectos que impactam em ganhos financeiros. Como já foi visto, sistemas de melhoramento animal visam fixar características genéticas que se traduzem em fenótipos que gerem retorno financeiro ao produtor. Nem todos os produtores, contudo, possuem condições de fazerem testes e mapeamentos genéticos em seus animais. Características fenotípicas, por outro lado, podem ser mais facilmente mensuradas. Exemplos dessas características são dimensões da área do lombo, idade da primeira cria, peso ao nascer, resistência a carrapatos e diversas outras métricas que podem ser usadas para avaliar a qualidade de um animal.

A expressão da qualidade de um animal a partir da mensuração de indicadores de qualidade pode ser expressa na forma de um índice. Um índice pode ser expresso como

uma média ponderada entre valores de características desejáveis (ou mesmo indesejáveis), de forma que o valor de um animal possa ser expresso por um número. Essa abordagem permite que a seleção de animais para acasalamento possa ser feita por meio de um simples ranking de valores. Essa abordagem, usando valores que geram retorno financeiro ao produtor é relativamente recente (SIMÕES et al., 2020) e vem sendo utilizada com sucesso.

A Embrapa Pecuária Sul desenvolveu um índice econômico de ciclo completo, ou simplesmente IECC, com o objetivo de caracterizar a produtividade do sistema de produção de bovinos de corte da raça Brangus (YOKOO et al., 2021), levando em consideração valores zootécnicos e econômicos, a partir de objetivos traduzidos em critérios de seleção para a prática de melhoramento animal. A partir deste estudo, foi criado um índice econômico que usa valores econômicos mensurados em dólares americano (US\$) para cada animal, estimando o seu valor. O objetivo deste índice econômico é classificar os animais por um valor individual único, sendo este valor uma representação da contribuição do animal para o lucro financeiro do sistema produtivo.

### **2.3 Programação linear e o método Simplex**

A análise da correção da resolução de um problema exige que ele seja formalizado, de forma a que alguma técnica possa ser usada em sua solução. Problemas de otimização são caracterizados pela tentativa de encontrar a melhor solução possível dentro de um espaço de soluções que, usualmente, não pode ser explorado na sua totalidade (CORMEN et al., 2002). Independente da ordem de complexidade do problema de otimização (que pode ser tratável ou intratável computacionalmente) (GAREY; JOHNSON, 1979), alguns arcabouços teóricos estão disponíveis para a construção de soluções para problemas de otimização.

Se um problema de otimização pode ser descrito por uma equação linear cujo valor deve ser maximizado (ou minimizado), sujeita a restrições nos valores das variáveis do problema que também são lineares, o problema pode ser modelado como um problema de programação linear. O termo “linear” é usado para identificar que todas as funções matemáticas utilizadas nesses modelos, são funções lineares. Enquanto que o termo “programação” neste caso é usado como um sinônimo de planejamento.

Uma função linear é caracterizada por seu gráfico ser uma linha reta (IEZZI et al.,

1995). Formalmente, uma função  $f : X_1 \times X_2 \times \dots \times X_k \rightarrow X$  é dita linear se é da forma

$$f(x_1, x_2, \dots, x_k) = a_1x_1 + a_2x_2 + \dots + a_kx_k$$

com  $a_1, a_2, \dots, a_k \in \mathbb{R}$ .

A programação linear abrange um planejamento de atividades com o intuito de se obter um resultado ótimo. Um resultado ótimo significa atingir o melhor objetivo especificado dentre todas as alternativas possíveis (HILLIER; LIEBERMAN, 2013).

Alguns problemas de otimização podem ser modelados como um problema de programação linear, tendo em vista que tanto sua função objetivo quanto suas restrições possam ser representadas por equações ou inequações lineares. Sendo uma função objetivo correspondente à combinação de valores que se deseja maximizar ou minimizar, dependendo do caso (CORMEN *et al.*, 2002, GOLDBARG; LUNA, 2005 *apud* MOTTA, 2021, p. 39).

Problemas de programação linear podem ser resolvidos com o método *Simplex*. Esse método é um procedimento iterativo que visa resolver problemas de programação linear por meio do caminhamento no espaço de soluções viáveis, restrito a um número finito de etapas, em que cada etapa move a solução para uma solução viável melhor que a anterior. Seu funcionamento pode ser descrito em três etapas: 1º: Conhecer uma solução básica viável inicial; 2º: Verificar se a solução é ótima; e 3º: Melhorar a solução a partir de um conjunto de regras e repetir o processo até que uma solução ótima seja encontrada. Existem soluções viáveis e inviáveis, uma solução viável é aquela que atende todas as restrições do problema e uma solução inviável é aquela que deixa de atender pelo menos uma das restrições. A verificação de otimalidade se dá ao analisar uma solução que esteja em um vértice da região de soluções viáveis. Caso não haja nenhuma outra solução adjacente a esta que seja melhor, esta solução é caracterizada como ótima. (HILLIER; LIEBERMAN, 2013).

O método Simplex possui uma interpretação geométrica, em que o espaço de soluções viáveis é delimitado pelas retas que representam as restrições do problema.

O tempo de execução do algoritmo Simplex depende da estrutura do espaço de soluções do problema. Problemas que podem ser representados como um problema de programação linear em que as variáveis podem tomar valores reais são usualmente resolvidos de forma eficiente pelo algoritmo Simplex, em tempo polinomial. Problemas de otimização modelados como um problema de programação linear em que o espaço de soluções somente admite valores inteiros podem gerar um tempo exponencial para a

solução do problema. A Seção 2.4, a seguir, descreve esses problemas e como o método Simplex ainda pode ser usado para conseguir uma solução mais eficiente.

## 2.4 Programação inteira e método *branch-and-bound*

A chamada programação inteira é um modelo de programação linear em que os valores das variáveis do problema devem ser inteiros. Essa restrição aplica-se a problemas em que somente números inteiros fazem sentido, como por exemplo alocar um número de funcionários para realizar determinadas atividades, neste caso não é possível alocar um número fracionário de funcionários. Há casos mais específicos, dentro da programação inteira conhecidos como de *programação inteira binária*. Estes problemas requerem uma resposta do tipo “sim-ou-não”. Nesse caso, as variáveis são de decisão e podem assumir os valores 0 ou 1 (HILLIER; LIEBERMAN, 2013). Problemas de programação inteira tendem a ser mais difíceis de se resolver do que problemas de programação linear. A solução de um problema modelado como programação inteira é NP-difícil mesmo que as variáveis sejam binárias (GAREY; JOHNSON, 1979).

Uma estratégia comumente usada para resolver problemas de programação inteira utiliza uma técnica conhecida como “relaxamento”, que consiste em tratar o problema de programação linear equivalente ao problema de programação inteira que se quer resolver. A estratégia consiste em considerar exatamente o mesmo problema, mas modificando as restrições que exigem que as variáveis assumam números inteiros por restrições que flexibilizem os valores factíveis (HILLIER; LIEBERMAN, 2013). Por exemplo, se uma variável  $x$  pode tomar somente os valores 0 ou 1, colocam-se as restrições que  $x \geq 0$  e  $x \leq 1$ . Dessa forma, o problema se torna um problema de programação linear usual e pode ser resolvido eficientemente pelo método Simplex. Os valores retornados podem não resolver o problema, mas a solução dada possivelmente contém em seu interior um conjunto não vazio de soluções inteiras para o problema. Percorrer esse espaço de estados limitado é mais eficiente do que percorrer o espaço inteiro inicial de soluções.

Um dos métodos de se percorrer o espaço de soluções é a técnica *branch-and-bound* (HILLIER; LIEBERMAN, 2013). O método *branch-and-bound* consiste em “ramificar” para depois avaliar progressivamente. O processo de ramificação (*branching*) é responsável por dividir o problema original em diversos subproblemas, cada vez menores, até que seja possível resolvê-los. A avaliação progressiva (*bounding*) consiste em encontrar a melhor solução nos subconjuntos gerados pelo processo de

ramificação, descartando aqueles subconjuntos que só podem oferecer soluções piores do que a melhor solução já encontrada.

O método *branch-and-bound*, quando aplicado a um problema NP-difícil, tem um tempo de execução que é exponencial no número de variáveis. Naturalmente que, nesse caso, a busca pela solução ótima torna-se inviável. O estudo do tempo de execução do algoritmo, por outro lado, pode dar pistas sobre a complexidade do problema subjacente, quando sua complexidade é desconhecida. Essa estratégia foi usada em (MOTTA, 2021) e indicou que o problema da seleção de acasalamentos, quando somente o índice econômico é levado em consideração, gerava um tempo de execução polinomial no número de animais. Os experimentos realizados abriram caminho para a construção da heurística gulosa ótima apresentada em (FERREIRA; YOKOO; MOTTA, 2021) e posteriormente usada em (FERREIRA, 2021) para mostrar que tanto o acasalamento entre iguais quanto o acasalamento compensatório são problemas computacionalmente tratáveis quando somente o índice de avaliação é usado.



### 3 MATERIAL E MÉTODOS

#### 3.1 Caracterização e fases da pesquisa

O trabalho pode ser caracterizado como de pesquisa bibliográfica e aplicada. As fases de execução do trabalho são as seguintes:

1. Revisão da literatura referente a estratégias de minimização da coancestralidade em rebanhos, com o objetivo de entender melhor o problema, avaliar o seu nível de formalização nos diferentes trabalhos e conhecer as soluções já propostas ou desenvolvidas.
2. Construção da definição formal do problema e busca por problemas similares na literatura relacionada à complexidade de algoritmos e complexidade de problemas.
3. Desenvolvimento dos algoritmos de acasalamentos com vistas à minimização de coancestralidade e maximização de índice econômico.
4. Desenvolvimento de um banco de dados baseado em grafos para armazenamento do *pedigree*.
5. Escrita do texto da monografia, escrita de artigos e publicação dos resultados.

As seções seguintes apresentam a descrição de cada fase listada acima.

#### 3.2 Revisão da literatura

A revisão da literatura foi feita por meio do método de revisão de escopo, em acordo aos procedimentos descritos em (FERREIRA, 2022) (adaptados de (KITCHENHAM, 2004) e (DERMEVAL; BITTENCOURT, 2020)). A seguir é apresentado o protocolo de revisão delineado, que consta das seguintes fases: (i) definição das questões de pesquisa, (ii) definição das *strings* de busca, (iii) explicitação dos critérios de inclusão e exclusão e (iv) seleção das fontes de pesquisa. Os resultados da execução do protocolo são apresentados ao final desta seção.

A partir dos objetivos do trabalho, foram definidas as seguintes questões de pesquisa para a revisão sistemática da literatura:

**Q1** Quais técnicas são utilizadas para minimizar a coancestralidade em sistemas de acasalamento?

Tabela 1 – Termos de pesquisa para a revisão da literatura

<b>Termo</b>	<b>Termos relacionados</b>	
Endogamia	inbreeding coefficient, consanguinity, inbreeding depression, coancestry	endogamia, consanguinidade, coancestralidade
Reprodução	breeding systems, mating systems	reprodução, sistemas de acasalamento
Minimizar	minimization, minimize, reduction, reduce, attenuate, decrease	minimizar, reduzir, atenuar, diminuir
Técnica	technique, method, strategy, scheme	técnica, método, estratégia, esquema

Fonte: Autor (2022)

**Q2** Existem descrições formais do problema da minimização da coancestralidade em sistemas de acasalamento?

A Tabela 1 apresenta os principais termos derivados das questões de pesquisa e seus respectivos sinônimos. Os sinônimos são necessários para que as buscas – restritas ao título e ao resumo dos trabalhos – consigam recuperar o maior número possível de trabalhos relacionados. Utilizando os termos que constam nessa tabela, foram desenvolvidas as seguintes *strings* de busca:

**String 1** - (inbreeding OR consanguinity OR coancestry) AND (technique OR method OR strategy OR scheme) AND (minimize OR reduce OR attenuate OR decrease) AND (breeding OR mating system).

**String 2** - (inbreeding OR consanguinity OR coancestry) AND (technique OR strategy) AND (minimize OR reduce) AND (breeding OR mating system).

**String 3** - (inbreeding coefficient OR inbreeding depression OR coancestry) AND (strategy OR scheme) AND (minimization OR reduction) AND (mating OR breeding systems).

**String 4** - (endogamia OR consanguinidade OR coancestralidade) AND (técnica OR método OR estratégia OR esquema) AND (minimizar OR reduzir OR atenuar OR diminuir) AND (reprodução OR sistema de acasalamento).

Estas *strings* foram desenvolvidas com o objetivo de obter o maior número de resultados capazes de responder a pelo menos uma das questões de pesquisa. Como pode ser observado, as *Strings* 2 e 3 são versões reduzidas da *String* 1; porém, a *String* 3 contém termos distintos, visando uma maior obtenção de resultados relacionados. As

versões reduzidas têm seu uso justificado pela necessidade de lidar com plataformas específicas, tais como a *ScienceDirect*<sup>1</sup>, que trabalham com um número limitado de conectores lógicos do tipo *AND* ou *OR* em seus mecanismos de busca. Para realizar buscas em Português foi criada a *String 4* que se trata de uma tradução da *String 1*.

A área de Melhoramento Animal em geral e de procedimentos para seleção de animais para reprodução e escolha de acasalamentos em particular é interdisciplinar. Trabalhos relacionados a esses temas podem ser encontrados em periódicos de Medicina Veterinária, Zootecnia, Genética, Estatística e Computação. A busca em fontes específicas, que chegam às centenas, poderia gerar mais resultados, mas com certeza haveria muitas publicações com as palavras-chave escolhidas mas que não estariam relacionados com o tema deste trabalho. Por essas razões, há trabalhos referenciados que não foram obtidos seguindo o protocolo de pesquisa, mas a partir de uma adaptação feita com base em algumas regras heurísticas. Após a revisão dos trabalhos, uma busca por fontes relevantes foi realizada na bibliografia, resultando em novos trabalhos identificados ao aprofundar as informações sobre autores específicos ou trabalhos previamente conhecidos. Os filtros utilizados nas buscas são relacionados à agricultura, ciência animal e genética, como pode ser visto na Tabela 2.

O critério definido para inclusão dos trabalhos se refere à capacidade de responder a pelo menos uma das questões de pesquisa preestabelecidas, ou seja, apresentar possíveis soluções para o problema. Já para os critérios de exclusão definidos temos que todo trabalho que não cumpra com o critério de inclusão deve ser descartado. Outro critério que exclui um trabalho da lista de relacionados é o uso de técnicas sem aplicabilidade com animais, ainda que aplicadas a outras espécies de seres vivos. Foram descartados trabalhos totalmente voltados para as áreas de genéticas e de uso de dados genômicos, sem se referir diretamente a esquemas ou organizações de acasalamento, além de trabalhos que não estivessem em língua portuguesa ou inglesa. Para trabalhos de um mesmo autor, lidando sobre um mesmo assunto, foi dada prioridade de inclusão para o mais recente, assim como para trabalhos clássicos que serviram de base para a maioria dos atuais, e que são comumente referenciados, a preferência se dá para as publicações mais recentes. A análise dos resultados da busca se iniciou com a revisão dos títulos dos trabalhos, avaliando sua pertinência com o tema em questão. Em seguida, foi feita a leitura dos resumos e, se necessário, dos trabalhos completos, para determinar se atendiam aos critérios de inclusão estabelecidos.

---

<sup>1</sup><https://www.sciencedirect.com/>

Tabela 2 – Resultados da aplicação do protocolo

<b>Chave Utilizada</b>	<b>Fonte</b>	<b>Filtro</b>	<b>Resultados</b>	<b>Trabalhos Selecionados</b>
String 1	Wiley Online Library	Published in: Animal Science Journal	41	1
String 3	Wiley Online Library	Subject: Animal Agriculture	134	2
String 1	Oxford Academic	Subject: Genetics and Genomics	67	1
String 2	ScienceDirect	Publication title: Animal	122	2
String 3	ScienceDirect	Publication title: Journal of Dairy Science	207	3
String 4	LocusUFV	–	72	1
–	Outras fontes	–	–	2

Fonte: Autor (2022)

Dentre os resultados obtidos nas pesquisas, a grande maioria não traz soluções para a redução da coancestralidade, mas dados que estimam a depressão endogâmica de um grupo específico, estudos de técnicas que trabalham com genotipagem, predição genômica e seleção por dados genômicos, além de trabalhos que apresentam técnicas para redução da coancestralidade em espécies de plantas. É possível perceber uma maior incidência de trabalhos relacionados ao assunto dentre os anos de 1990 e 2005, após este período o enfoque se torna maior em relação a métodos e análises relacionadas a genotipagem.

No Capítulo 4 são apresentados os trabalhos selecionados, que foram considerados relevantes de acordo com os critérios de inclusão. Sendo 10 (dez) destes trabalhos encontrados a partir das *strings* de busca e 2 (dois) a partir da busca por trabalhos de determinados autores reconhecidos por investigar técnicas para a redução da consanguinidade. Dentre os trabalhos encontrados através de outras fontes, 33 (trinta e três) são trabalhos relacionados a outros temas que não a redução da consanguinidade, mas que trazem conceitos de coancestralidade, consanguinidade, índice econômico, entre outros assuntos que são abordados neste trabalho.

### 3.3 Ferramental tecnológico

A linguagem de programação R<sup>2</sup> foi usada no desenvolvimento do trabalho, com o RStudio<sup>3</sup> servindo como IDE. Essa escolha permitiu o reúso dos pacotes já construídos no âmbito do projeto de pesquisa coordenado pela orientadora deste trabalho, com os algoritmos de otimização e tratamento de dados de entrada. Outra razão para escolha da linguagem R é o fato de ser usada no ambiente de trabalho da Embrapa Pecuária Sul, criando assim a possibilidade do trabalho desenvolvido ser utilizado pela EMBRAPA, sem necessidade de recodificação. Há margem para futuras otimizações, considerando a implementação dos dados referentes aos parentescos de cada animal diretamente no banco de dados. Essas otimizações serão consideradas futuramente.

Os pacotes utilizados para construção do algoritmo CoancestryMIN-Selection foram os seguintes:

**Rcpp** (EDDELBUETTEL et al., 2022): este pacote é uma interface entre o R e o C++, permitindo escrever códigos em C++ e utilizá-los diretamente no R. No entanto, é necessário compilar o código em C++ antes de usá-lo, o que pode ser feito com ferramentas externas, como o compilador C++ padrão ou ferramentas de desenvolvimento integrado (IDEs). A função *cppFunction* permite escrever códigos em C++ para então executá-los dentro do programa R. Essa função foi utilizada exclusivamente no desenvolvimento do algoritmo CoancestryMIN-Selection, visando a otimização de estruturas de repetição. A *cppFunction* recebe como argumento uma variável do tipo *string* que é interpretada como uma função em C++. Dado que C++ é uma linguagem compilada de tipagem estática e o R é uma linguagem interpretada de tipagem dinâmica, a primeira oferece a vantagem de gerar códigos com execução mais rápida, o que é especialmente importante na execução de estruturas de repetição dos algoritmos deste trabalho.

**pedigreemm** (BATES; VAZQUEZ, 2015): o pacote *pedigreemm* tem o propósito de ajustar modelos de efeitos mistos baseados em pedigree. As funções deste pacote que foram utilizadas no desenvolvimento do algoritmo CoancestryMIN-Selection, tiveram como propósito a obtenção dos coeficientes de parentescos entre cada animal da base de dados, incluindo a possível prole. A função *pedigree* nos permite construir um objeto da classe “pedigree”, que tem como argumentos: *sire*: um

---

<sup>2</sup><<https://www.r-project.org/>>

<sup>3</sup><<https://www.rstudio.com/>>

vetor numérico com a identificação dos pais/touros; *dam*: um vetor numérico com a identificação das mães/vacas; *label*: um vetor numérico que consta a identificação dos respectivos animais. A saída da função é um objeto do tipo “pedigree”. A partir do objeto “pedigree” pode-se usar a função *relfactor* para obter o fator de relacionamento de um pedigree. Esta função tem como argumento o objeto “pedigree” criado pela função *pedigree* e sua saída é uma matriz esparsa incompleta com os coeficientes de parentesco entre cada indivíduo. Para se obter a matriz completa de relacionamento, é necessário utilizar a função *crossprod* no objeto criado a partir da função *relfactor*. A função *crossprod* pertence à funcionalidade básica do R, recebendo como argumento duas matrizes e retornando a matriz resultante da multiplicação entre elas.

**lpSolve** (BERKELAAR et al., 2020): o *lpSolve* é um pacote para resolver problemas de programação linear, utilizando os métodos Simplex e a abordagem *branch-and-bound*. A principal e única função deste pacote que foi utilizada na construção do algoritmo CoancestryMIN-Selection chama-se *lp* e possui os seguintes argumentos: *direction*: é uma *string* indicando o tipo de problema, utiliza-se *max* para maximização ou *min* para minimização; *objective.in*: é um vetor numérico com os coeficientes da função objetivo; *const.mat*: é uma matriz com os coeficientes das restrições, sendo definido uma linha por restrição e uma coluna por variável; *const.dir*: é um vetor de caracteres contendo a direção das restrições, cada valor deve compor um operador de comparação como, por exemplo, “>=” (maior ou igual); *const.rhs*: é um vetor numérico contendo os valores do lado direito das restrições. No último argumento é possível especificar o tipo de variáveis do problema, tais como *all.bin* com valor *TRUE* para variáveis binárias. Sua saída esperada é um objeto do tipo “lp”, contendo informações como solução e status, sendo esse último responsável por indicar a eficiência em encontrar uma solução viável.

**Matrix** (BATES; MAECHLER; MAECHLER, 2017): o pacote *Matrix* apresenta uma vasta hierarquia de classes de matrizes, além de diversos métodos e funções para manipulação de dados em matrizes. Neste trabalho o pacote *Matrix* é utilizado para extrair a parte triangular superior de uma matriz por meio da função *triu*, dentro do processo de multiplicações de matrizes esparsas. A função *triu* recebe como argumento uma matriz esparsa e a retorna sua parte triangular superior. Sendo uma matriz triangular superior composta por termos que abaixo da diagonal principal

são iguais a zero e termos que acima da diagonal principal são números reais. A triangulação é usada porque a relação de parentesco é uma relação simétrica.

**data.table** (DOWLE et al., 2021): o pacote *data.table* tem o propósito de lidar com dados tabulares de maneira otimizada, especialmente para as operações de adição, modificação e exclusão de dados. Além de permitir criar variáveis do tipo “data.table” a partir de variáveis de tipos diferentes. Neste trabalho é utilizada a função *rbindlist*, responsável por retornar uma variável do tipo “data.table” a partir da entrada de várias listas de valores. Essa função se fez necessária dentro do processo de multiplicação entre matrizes esparsas.

**fst** (KLIK; COLLET, 2022): o pacote *fst* permite a serialização paralela de dados compactados usando o formato “fast-storage” (.fst). O formato .fst permite o acesso aleatório aos dados armazenados, além de diversas configurações de compactação usando os compressores *LZA* e *ZSTD*. As funções desse pacote utilizadas neste trabalho são referentes à escrita e à leitura de documentos do tipo .fst por meio das funções *write.fst* e *read.fst*, respectivamente. A utilização de documentos do tipo .fst permite a possibilidade de trabalhar com uma quantidade maior de dados, o que pode ser útil na realização de multiplicações entre matrizes esparsas com mais de 1 milhão de elementos, ou seja, com dimensões elevadas. A função *write.fst* recebe como argumentos a variável que se quer gravar em arquivo e diretório onde o arquivo deverá ser armazenado, sua saída é um arquivo .fst. A função *read.fst* possui os argumentos o nome do arquivo .fst que se quer ler e o diretório em que está armazenado, para então o retornar para variável desejada.

**dplyr** (WICKHAM; WICKHAM, 2022): o pacote *dplyr* é uma ferramenta com propósito de ser rápida e consistente para tratar dados tabulares como objetos, tanto na memória quanto fora dela. Este pacote foi utilizado para se obter o somatório de valores que pertencem a um mesmo índice. Foram usadas as funções *group\_by* responsável por agrupar por uma ou mais variáveis e *summarise* responsável por identificar o número de linhas idênticas. Estas funções são necessárias para se tratar os dados obtidos após realizar uma multiplicação entre matrizes esparsas. As funções *group\_by* e *summarise* funcionam juntas através do operador *pipe* (`%>%`), que é responsável por utilizar a saída de uma função como entrada da outra. A união das funções *group\_by* e *summarise* recebem como argumento uma matriz, especificando uma coluna de posições e outra de valores. Para então retornar uma nova matriz sem posições repetidas e como valor traz a soma dos valores que

possuíam a mesma posição.

**scales** (WICKHAM; SEIDEL, 2020): o pacote *scales* tem o propósito de realizar ajustes de escala, dado um conjunto de dados. Neste trabalho o pacote *scales* é utilizado com o intuito de realizar a normalização dos valores de um vetor. Isto é possível por meio do uso da função *rescales* que tem por argumento um vetor numérico junto dos valores mínimo e máximo que determinam a nova escala desejada. A saída esperada é um vetor com seus valores normalizados respeitando o mínimo e máximo estabelecidos anteriormente.

O ambiente computacional onde está sendo desenvolvido o algoritmo, além dos testes e validações, é um notebook com processador AMD Ryzen™ 7 3700U de 2.3 GHz; 8GB de memória RAM; Armazenamento SSD de 250GB. Com o sistema operacional Windows 11 Home de 64 bits.

A combinação das técnicas Simplex e *branch-and-bound* usadas para desenvolver o algoritmo CoancestryMIN-Selection, somado à aplicação do relaxamento para lidar com problemas de inteiros, pode resultar em uma solução ótima para o problema em questão, mas não há garantia de que sempre seja obtida uma solução ótima.

### 3.4 Fontes de dados

A base de dados utilizada neste trabalho, contendo informações do *pedigree* e índice econômico de cada animal, foi providenciada pela EMBRAPA em planilhas eletrônicas no formato *comma-separated-values* (.csv). O formato de arquivo .csv é utilizado para representar dados tabulares e possui seus valores separados por vírgulas dispostos em várias linhas, em que cada linha contém informações referentes a um único animal. Para realizar a importação dos dados foi utilizada a função *read.csv* pertencente ao R base. Logo após a importação, os dados são armazenados na variável chamada *dbase*, do tipo *dataframe*. Uma variável do tipo *dataframe* no R permite armazenar dados tabulares com diferentes tipos de dados, como por exemplo, inteiros e caracteres.

A base de dados é constituída por dados referente a 3.515 bovinos da raça Brangus, sendo 1.468 machos e 2.029 fêmeas, além de 18 animais com sexo indefinido. A Figura 1 apresenta a estrutura dos dados por meio de um recorte da base de dados.

Os dados correspondem as identificações únicas de cada animal e de seus respectivos pais e mães, configurando assim o seu *pedigree*. Além de trazer informações



Figura 1 – Extrato da base de dados

	A	B	C	D	E	F	G
1	ID	ID_Pai	ID_Mae	Safra	Sexo	IECC	Selecionado
2	6297	0	0	1990	MACHO	0.02	0
3	Ebano1778	0	0	1992	FEMEA	0.04	0
4	KARUMANSO	0	0	1992	MACHO	0.06	0
5	MissThun	0	0	1992	FEMEA	0.07	0
6	OLHOS1222	0	0	1992	FEMEA	0.09	0
7	OLHOS1675	0	0	1992	MACHO	0.11	0
8	Rancho	0	0	1992	MACHO	0.13	0
9	28027	0	25134AAB	1992	MACHO	0.15	0
10	11107	6178	Y227590AAB	1996	MACHO	0.17	0
11	10829	OLHOS1675	OLHOS1222	1997	MACHO	0.18	0
12	Ebano2153	0	0	1998	MACHO	0.2	0
13	H95	0	0	2000	FEMEA	0.22	0
14	Pacote	Fenomeno	Jandaia	2000	MACHO	0.24	0
15	112876	Ebano2153	Ebano1778	2001	MACHO	0.26	0

Fonte: Autor, (2022)

como a safra de cada animal, que se trata do ano de nascimento. O valor do IECC, ou índice econômico de ciclo completo, indica o lucro esperado em dólares americanos e a coluna “Selecionado” indica quais machos e quais fêmeas pretende-se acasalar.

## 4 TRABALHOS RELACIONADOS

### 4.1 Abordagens para minimização da coancestralidade

A primeira questão de pesquisa da revisão bibliográfica (Seção 3.2) visa descobrir quais são as técnicas correntemente utilizadas para minimizar a coancestralidade em sistemas de acasalamento. Os trabalhos encontrados que abordam esse problema estão descritos abaixo e são sumarizados na Tabela 3.

Nomura e Yonezawa (1996) comparam quatro métodos diferentes de organizações de grupos de acasalamento: um modelo circular e três modelos cíclicos (métodos *Falconer*, *HAN-rotational* e *Cockerham*), com o objetivo de encontrar qual deles melhor controla as taxas de endogamia de uma população. O modelo circular é um sistema, onde uma população é dividida em diferentes grupos para então os machos serem transferidos entre grupos vizinhos de forma circular. Os modelos cíclicos, por sua vez, possuem um padrão cíclico de deslocamento dos machos entre os grupos. A principal diferença entre os três métodos cíclicos citados ocorre na organização dos deslocamentos que serão realizados a cada ciclo. Nomura e Yonezawa (1996) dizem realizar “computações numéricas” para revelar que quando em sistemas circulares o número de grupos ( $m$ ) é maior que seis, a consanguinidade supera os valores do acasalamento ao acaso. Esse efeito é causado pelo acasalamento entre primos, dado que a cada geração as fêmeas sempre acasalam com machos vindos do mesmo grupo. É visto que há um acréscimo inicial na taxa de consanguinidade no método *Falconer*, porém nas gerações seguintes sofrem uma redução. O método *Falconer*, segue um padrão oscilante de aumento e diminuição das taxas de consanguinidade. Por fim é definido que para um número de grupos ímpares o método *HAN-rotational* é o mais recomendado. Enquanto que para números pares de grupos o método *Cockerham*. Ambos os métodos apresentaram menores variações em relação às taxas de endogamia ao longo das gerações. A escolha do método deve ser feita visando a maior estabilidade, dado que uma oscilação abrupta na taxa de endogamia do rebanho pode resultar em uma severa depressão por endogamia (*inbreeding depression*), como apresentado por Beilharz (1982 *apud* NOMURA; YONEZAWA, 1996, p. 147). É possível observar, ainda nesse mesmo trabalho, que quanto maior o número de grupos mais favorável é para os sistemas, dado que o tamanho efetivo da população ( $N_e$ ) aumenta. Porém, cada grupo deve ser grande o suficiente para seja garantida sua persistência. O tamanho efetivo de uma população é o fator que indica o número de indivíduos que

efetivamente contribuem para a variância amostral ou taxa de endogamia, porém desde que sejam acasalados sob a suposição de uma população ideal (FILHO, 1999). Nenhum dos modelos propostos é capaz de evitar a consanguinidade a longo prazo dentro de um grupo fechado. A única possibilidade é manter as taxas de consanguinidade do rebanho sob controle. A ocorrência de consanguinidade em um rebanho não é um fato que comprometa a população: caso ocorra de forma gradual, não causará nenhuma séria depressão por endogamia: alelos deletérios podem ser eliminados gradualmente por meio da seleção. Ainda segundo Lande e Barrowclough (1987 *apud* NOMURA; YONEZAWA, 1996, p. 155), quanto mais lento for o processo de endogamia, maior será a oportunidade de eliminação dos alelos deletérios.

Paiva (2010) traz em sua tese diversas simulações realizadas a partir do sistema computacional *GENESYS 2010*, buscando os efeitos de algumas técnicas no que diz respeito ao controle de endogamia em populações sob seleção baseada em *Best Linear Unbiased Prediction* (BLUP). BLUP pode ser traduzido como a melhor predição linear sem viés. Uma seleção baseada em BLUP visa aumentar a exatidão das avaliações genéticas e melhorar a resposta genética a curto prazo, levando em consideração todas as relações conhecidas de parentesco do indivíduo (VERRIER; COLLEAU; FOULLEY, 1993). Inicialmente, é visto um comparativo de desempenho quanto à redução das taxas de endogamia entre quatro métodos de acasalamento distintos, sendo eles: acasalamento com exclusão de irmãos completos e meio-irmãos, acasalamento de melhores com melhores, acasalamento de melhores com piores (acasalamento compensatório) e acasalamento ao acaso. O acasalamento ao acaso é também conhecido como acasalamento aleatório, sendo definido como o sistema em que qualquer indivíduo de uma população tem as mesmas chances de se acasalar e deixar descendentes, podendo ser considerado um sistema de simples implementação com baixo custo. Entretanto, esse sistema de acasalamento pode ser aplicado junto a um sistema de seleção, onde machos e fêmeas previamente selecionados, acasalariam de forma aleatória (ELER, 2017). Os resultados obtidos, ao analisar as 50 gerações seguintes à geração inicial mostraram aumento do valor fenotípico em todos os casos. O melhor resultado se dá no modelo de acasalamento compensatório a partir da 13ª geração, enquanto que o método de acasalamento entre melhores reprodutores apresentou a maior taxa de fixação de alelos desfavoráveis. A justificativa dá-se pelo fato de que ao acasalar os melhores animais, que por sua vez são aqueles que possuem a maior porcentagem de genes favoráveis já fixados, acabam restando somente genes desfavoráveis para fixação. O método que

manteve os melhores valores de endogamia ao longo das gerações foi o de acasalamento compensatório que também traz um menor esgotamento de fixação de alelos favoráveis e consegue manter por mais tempo o limite de seleção. Justifica-se essa situação pelo fato de que este método acasala os animais com valores genéticos mais distantes, sendo esperado que estes possuam pouco ou nenhum parentesco. Os resultados deste primeiro teste apontaram que o modelo de acasalamento compensatório é um dos melhores para manutenção da endogamia a longo prazo. O segundo teste realizado visa comprovar a eficácia do aumento no número de machos selecionados para controle do aumento nos níveis de endogamia. Para realização dos testes foram simulados dois cenários diferentes, o primeiro onde eram selecionados 20 machos e 200 fêmeas por geração, formando uma relação 1:10 e no segundo onde eram selecionados 40 machos e 200 fêmeas por geração, formando uma relação 1:5. Os tamanhos efetivos das populações eram 72,72 e 133,33 respectivamente. O aumento do número de machos em todos os casos atrasou o ponto em que todos os animais são endogâmicos, além de causar uma redução na fixação de alelos desfavoráveis e aumento nos valores fenotípicos. O melhor resultado foi aquele que usou o modelo de acasalamento compensatório junto do aumento no número de machos. A terceira simulação verifica os valores de endogamia em diferentes intensidades de seleção de fêmeas. Foram utilizados dois cenários de seleção, o primeiro onde 0% das fêmeas são obtidas por seleção e o segundo onde 100% das fêmeas são obtidas por seleção. O resultado comparativo demonstrou somente pequenas diferenças nos valores fenotípicos, e demais atributos levados em consideração. É visto que em populações fechadas o responsável por quase todo aumento na taxa de endogamia é o sexo usado em menor número. Paiva (2010) também aponta que a seleção feita sobre os machos reduz o número de famílias o que por sua vez ao longo do tempo torna todas as fêmeas aparentadas. No quarto e último teste vemos uma comparação de diferentes valores de herdabilidade aplicados ao BLUP, visando uma menor taxa de endogamia. Os valores comparados são 0,10 (valor real utilizado nos testes anteriores), 0,30, 0,50, 0,70 e 0,90. O único modelo de acasalamento utilizado desta vez foi o acasalamento ao acaso. Como resultado, foi obtida uma redução nas taxas de endogamia ao aumentar o valor de herdabilidade de 0,10 até o valor de 0,70, após isso os resultados se tornaram muito similares, mostrando haver um limite do quanto pode ser melhorado utilizando esta técnica. O aumento no valor da herdabilidade faz com que seja dado um peso maior para as informações do indivíduo e um peso menor para as informações de sua família. O problema de se ter famílias com pesos elevados é o aumento da tendência de serem escolhidas. Caso seja levado em

consideração somente o maior valor genético como critério de escolha, isto pode se tornar problemático. Ao utilizar muitos reprodutores de uma mesma família, o esperado é que ocorra um acréscimo endogâmico. Para o BLUP, quanto maior a herdabilidade menor é o peso das informações familiares, o que explica a queda nos valores de endogamia. Ainda Villanueva *et al.* (1994 *apud* PAIVA, 2010, p. 118), relataram aumentar a herdabilidade de 0,35 para 0,70, assim obtendo redução das taxas de endogamia. Nas simulações também foi constatado que aumentar o valor de herdabilidade de 0,10 para 0,30 foi suficiente para retardar o ponto em que todos os animais são endogâmicos em até duas gerações. O aumento do valor da herdabilidade também é capaz de propiciar um decréscimo da taxa de fixação de alelos desfavorecidos e causar aumento dos valores fenotípicos.

Honda, Nomura e Mukai (2005) revelam um estudo que busca uma solução para o problema de consanguinidade em raças de gado para uso comercial. Assume-se uma estrutura hierárquica de raças, com nível inferior e superior. No nível superior, genes favoráveis são acumulados por seleção artificial e o progresso genético alcançado é então transferido para o nível inferior, por meio do deslocamento dos machos. O sistema de acasalamento examinado foi o cruzamento rotacionado que consiste em alternar a raça do pai a cada geração, podendo ser utilizada duas ou mais raças. Deve-se sempre levar em consideração que as raças sejam semelhantes em características como tamanho corporal e produção de leite, sendo importante possuírem uma boa adequação do genótipo ao ambiente geral (FILHO, 1997). Ainda Honda, Nomura e Mukai (2005) desenvolveram uma equação de recorrência para o coeficiente de consanguinidade em fêmeas comerciais mantidas por cruzamento rotacionado com linhagens parcialmente isoladas. Testes foram realizados com diferentes números de linhagens ( $n$ ), diferentes números de machos ( $N_m$ ) e diferentes taxas de migração dos machos ( $d_m$ ), mas com um número de fêmeas ( $N_f$ ) fixo ( $= 200$ ) e taxa de migração das fêmeas ( $d_m$ ) fixa para todos os casos. Os resultados obtidos por Honda, Nomura e Mukai (2005) demonstram que o fator crítico para determinar a consanguinidade é o número de linhagens, visto que aumentar  $n$  reduz a taxa de endogamia, porém somente aumentá-lo até  $n \leq 4$ , pois após esta marca o efeito torna-se insignificante. Chega-se à conclusão de que se utilizados quatro ou cinco linhagens, o cruzamento rotacionado pode ser considerado um sistema eficaz não só para reduzir a endogamia de curto e longo prazo das fêmeas comerciais, com também auxiliar na manutenção da diversidade genética, independentemente do tamanho efetivo de cada linhagem.

Moreno et al. (2011) fazem uso de uma população de ratos domésticos para testar três combinações diferentes de métodos de seleção e acasalamento. Enquanto visa a redução das taxas de endogamia, também busca-se o melhoramento nas taxas de ganho de peso. O uso de ratos para esses testes se mostra útil dado que os resultados obtidos poderão ser melhor aproveitados para outros animais como ovelhas e porcos (HILL & CABALLERO, 2000 *apud* MORENO et al., 2011, p. 2). Para iniciar o experimento foi criada uma geração 00 com a cruz de 72 machos com 2 fêmeas cada, resultando em 144 ninhadas. A partir das ninhadas geradas foram selecionados 32 machos para compor cada um dos 3 grupos distintos a serem analisados pelas 17 gerações seguintes. Sendo eles o grupo A que utiliza o método de seleção individual junto do método de acasalamento aleatório, grupo B que utiliza o método de seleção ponderada junto do método de acasalamento aleatório e grupo C que utiliza o método de seleção individual junto do método de acasalamento de mínima coancestralidade. Originalmente foi proposto por Wright (1921 *apud* MORENO et al., 2011, p. 2) um método de minimização da endogamia e posteriormente aprimorado para o método de mínima coancestralidade por Nieto *et al.* (1986 *apud* MORENO et al., 2011, p. 2). Devemos levar em consideração que consanguinidade e coancestralidade são coisas diferentes ainda que estritamente relacionadas. Como afirmado por (MORENO et al., 2011), evitar o acasalamento entre indivíduos com algum parentesco, atrasa o aumento da consanguinidade, porém aumenta a taxa de coancestralidade, sendo esperado que a longo prazo cause aumento da consanguinidade. Para o grupo A foi feita a seleção individual de 8 machos dentre os 32, levando em conta o desempenho relativo ao ganho de peso entre os seus 21 e 42 dias. O acasalamento ocorreu de maneira aleatória, porém evitando o acasalamento entre irmãos, com uma intensidade de seleção dos machos no valor de 1,235 e tamanho efetivo da população ( $N_e$ ) de 19,8. Para o grupo B foram selecionados 12 machos, resultando em uma intensidade de seleção de 1,246 e tamanho efetivo de 31,5. Para o grupo C foram selecionados 8 machos e 16 fêmeas com menor grau de parentesco, com intensidade de seleção e tamanho efetivo idênticos ao grupo A. Por fim para análise dos resultados nas taxas de endogamia foi utilizado o programa endog (na versão v4.6). Segundo Moreno et al. (2011) foi possível perceber uma melhora na característica desejada já na primeira geração, demonstrando uma resposta considerável dos métodos de seleção. Todos os padrões de respostas aos métodos de seleção foram bastante similares. O grupo que melhor manteve a variabilidade genética foi o B, com as menores taxas de coancestralidade e consanguinidade, seguido pelo grupo C. Valores

mais baixos nas taxas de endogamia também resultaram em menores taxas de mortalidade se comparados com o grupo A que obteve os piores resultados.

Pryce, Hayes e Goddard (2012) realizam a comparação de três estratégias diferentes para redução das taxas de endogamia da progênie em planos de acasalamento. Cada estratégia faz uso de informações diferentes, sendo elas: o coeficiente de endogamia da linhagem/pedigree, as relações genômicas e os segmentos homozigóticos compartilhados. Avalia-se em cada uma a endogamia e a redução no ganho genético. Para realizar as simulações foram usadas amostras de DNA de 2.117 touros *Holstein*, dos quais foram feitas a devida genotipagem. Esses touros foram nascidos entre 1995 e 2005, procriados por 316 reprodutores diferentes e 1589 matrizes diferentes. Para os testes foram considerados um número fixo de 400 fêmeas, enquanto 320 machos foram selecionados aleatoriamente dos 2.117 disponíveis, a partir destes, foram selecionados 20 machos levando em conta os maiores valores de Angus Performance Register (APR). APR se trata de um ranking de lucro australiano, interpretado como lucro líquido em dólares por vaca ao ano. Assim foram estabelecidos 300x20 acasalamentos em potencial e, a partir deles, foi definida uma matriz de relacionamentos que determina para as fêmeas a um único par e a possibilidade de até 30 fêmeas diferentes para cada macho. Os resultados demonstraram que uma matriz de relacionamento genômico é mais eficaz para reduzir a endogamia esperada da progênie, do que utilizar as informações do pedigree para formar os planos de acasalamento. Analisando sob um viés econômico é perceptível que a perda que se dá ao usar a matriz genômica pode ser considerada pequena se comparada com o ganho que se tem pela redução da consanguinidade. Sendo que a elevação das taxas de endogamia afetam a lucratividade, pois prejudicam características relativas a predisposição física e as capacidades reprodutivas do animal (SMITH *et al.*, 1998 *apud* PRYCE; HAYES; GODDARD, 2012, p. 377). Porém, Pryce, Hayes e Goddard (2012) afirmam que uma aplicação a grandes criações não era algo viável, visto que as informações de pedigree são adquiridas “gratuitamente” e seu uso já confere um resultado “aceitável” nos planos de acasalamento.

Windig e Kaal (2008) comparam dois esquemas de acasalamento: o modelo circular (analisado anteriormente por Nomura e Yonezawa (1996)) e o modelo de prevenção máxima de consanguinidade que funciona tal como o método *Falconer* analisado anteriormente por Nomura e Yonezawa (1996). No método Falconer as combinações doador-receptor mudam a cada geração, de modo que o uso da mesma combinação seja adiado o máximo possível. A comparação busca definir qual deles

tem maior desempenho no controle das taxas de endogamia ( $\Delta F$ ) aplicados à raça de ovinos *Kempisch Heideschaap*. Para análise também foram simulados outros três casos que não utilizam nenhum método de rotação. O primeiro caso possui permissão de troca de animais entre todos os rebanhos. Isto é, todos os machos tinham igual probabilidade de acasalar com as fêmeas, independente do rebanho de origem. O segundo caso do qual não há nenhuma permissão de troca de animais entre rebanhos, cada rebanho utilizava os machos que ali eram gerados. O terceiro caso busca se aproximar ao máximo de uma situação real, onde existe a situação de troca, porém de forma limitada. Para isso, foi definida uma probabilidade de 5% dos machos serem utilizados fora dos seus rebanhos de origem. O método de seleção usado foi o aleatório (também conhecido como acasalamento ao acaso). As simulações foram feitas objetivando a maior similaridade possível com a população real de *Kempisch Heideschaap*. Cada animal da simulação recebeu informações como sexo, idade e condições de rebanho de um animal real. O tamanho total da população foi de 2160 animais que, posteriormente, foram divididos em 8 rebanhos. A proporção sexual usada foi de 1 macho para 5 fêmeas. Para todos os cenários foram reproduzidas 25 simulações para 50 gerações visando uma média final concisa. Os resultados obtidos por Windig e Kaal (2008) definem uma melhor performance na redução das taxas de endogamia utilizando o modelo circular. Essa se mostrou uma opção viável não apenas para reduzir o coeficiente de endogamia ( $F$ ) como também para igualar a variação da taxa de endogamia ( $\Delta F$ ) entre os rebanhos. Além de mostrar-se capaz de manter o  $F$  baixo a longo prazo, possuindo maior previsibilidade das taxas de endogamia ao longo do tempo.

Fernández, Toro e Caballero (2003) tem por objetivo mostrar um comparativo entre diferentes esquemas de acasalamento que tentam trazer uma redução nas taxas de endogamia, utilizando simulações feitas por computador. Os esquemas avaliados são *W* (modelo proposto por Wang (1997 *apud* FERNÁNDEZ; TORO; CABALLERO, 2003, p. 886)), *SBW* (modelo proposto por Sánchez-Rodríguez *et al.* (2003 *apud* FERNÁNDEZ; TORO; CABALLERO, 2003, p. 886)) e modelo de Minimização de Coancestralidade Global (*GC*), que é descrito por Fernández and Toro (1999 *apud* FERNÁNDEZ; TORO; CABALLERO, 2003, p. 886). Os dois primeiros modelos são considerados hierárquicos, pois fixam as contribuições dos indivíduos, enquanto o último organiza as contribuições de uma geração para outra, intencionando obter a mínima coancestralidade global dos pais. As simulações foram realizadas levando em consideração a robustez e aplicabilidade prática dos esquemas, ao longo de 50 gerações. Para cada geração foram calculados os



valores de coeficiente de endogamia e taxa de variação da endogamia. Indivíduos da primeira geração foram considerados não relacionados. O modelo de seleção escolhido foi o aleatório, sendo permitido o acasalamento de uma fêmea com mais de um macho ainda que isto seja de difícil implementação prática. Foram realizadas 1000 repetições para simulações do modelo hierárquico e 200 para o modelo *GC*. Os resultados evidenciaram um desempenho superior do modelo *GC* quanto a curto e médio prazo, sendo superado pelo modelo *SBW* quando se trata de longo prazo. Fernández, Toro e Caballero (2003) considera que 30-50 gerações são valores distantes e dificilmente requisitados, o que torna o modelo *GC* melhor em termos de aplicabilidade.

Nomura (1999) apresenta um estudo da base teórica do acasalamento compensatório, uma proposta de melhoria e resultados de simulações estocásticas comparando o método proposto a outros quatro. Toro *et al.* (1988 *apud* NOMURA, 1999, p. 356) desenvolveu o modelo de acasalamento de mínima coancestralidade, que mais tarde teve modificações sugeridas por Caballero *et al.* (1996 *apud* NOMURA, 1999, p. 356), das quais consistiam em unir o método de acasalamento compensatório ao modelo de coancestralidade mínima. Nesse modelo misto, os pares são formados seguindo os conceitos do acasalamento compensatório e posteriormente as fêmeas são trocadas de pares (sem que o modelo compensatório seja prejudicado), seguindo os conceitos da mínima coancestralidade. Então, seguindo essa lógica, é proposto um sistema de trocas diferente, baseado nos valores genéticos estimados ou *estimated breeding values* (EBV). A simulação abrangeu três tipos de seleção, sendo elas: por índice de seleção, seleção fenotípica e BLUP. Os métodos de acasalamento comparados são o modelo de acasalamento compensatório proposto (CM + ASEBV), acasalamento aleatório (RM), acasalamento compensatório (CM), acasalamento compensatório combinado ao acasalamento de mínima coancestralidade (CM + MC) e acasalamento de mínima coancestralidade (MC). Os testes foram executados 300 vezes para os modelos de seleção fenotípica e para o de índice de seleção, enquanto foram executadas 50 vezes para o modelo BLUP. Os resultados apontaram maiores tamanhos efetivos em populações que utilizavam acasalamento de mínima coancestralidade. É apresentado por Nomura (1999) que a seleção BLUP tende a conceder um maior peso às informações da família, tornando maior a probabilidade dos indivíduos selecionados serem intimamente relacionados. Neste caso, o método de mínima coancestralidade inclina-se a ser uma das melhores opções. Nomura (1999) afirma que a eficiência de CM + ASEBV é prejudicada pela falta de precisão dos dados EBV em uma seleção fenotípica. Porém, tem seu desempenho

aprimorado quando submetida a uma seleção BLUP com dados EBV precisos, além de apresentar a possibilidade de potencializar os efeitos do modelo de acasalamento compensatório.

NOMURA et al. (2002) apresentam uma comparação entre diferentes combinações de métodos de seleção e acasalamento, buscando avaliar a sua eficiência para ganhos genéticos para linhas fechadas de frango. Dois traços diferentes são levados em consideração, sendo o primeiro a melhoria desejada (peso corporal e idade em que se atinge maturidade sexual) e o segundo a taxa de consanguinidade. Os métodos de seleção avaliados são o DIS, que baseia-se no índice de seleção para ganho genético no tópico desejado e LPS, que consiste de uma combinação entre o índice familiar e técnica de programação linear. Os esquemas de acasalamento usados são acasalamento aleatório e acasalamento de mínima coancestralidade. Com isso as combinações testadas foram DIS + RM, DIS + MC, LPS + RM e LPS + MC. As seleções foram projetadas para atingir a meta de peso corporal de 2600g em 6 semanas e maturidade sexual em 180 dias. A estrutura populacional simulada foi baseada em linhas de frango de corte existentes, desenvolvidas no centro nacional de pecuária de *Hyogo*. Nas simulações desenvolvidas tem-se uma população discreta de 40 machos e 280 fêmeas, formando uma relação 1:7. Cada macho é responsável por gerar um total de 35 descendentes, formando um total de 1.400 descendentes aptos para seleção da qual definirá os próximos 40 pais e 280 mães da geração seguinte. Os resultados das simulações foram retirados em cima da média de 300 repetições realizadas para cada teste. É visto que o ganho genético entre os quatro esquemas é semelhante, constatando que o modelo de acasalamento de mínima coancestralidade não induz nenhum efeito vital sobre o progresso genético. Dentre os métodos de seleção aquele que demonstrou menor variância nos ganhos genéticos foi o LPS, demonstrando assim maior “robustez” contra possíveis erros em estimativas de parâmetros genéticos. Também é apurado por NOMURA et al. (2002) que independentemente dos métodos de seleção usados, o método de acasalamento MC foi capaz de reduzir a consanguinidade para aproximadamente 80% de RM, além de apresentar maior uniformidade nos coeficientes de endogamia individual. Ao final da comparação, dados os resultados o melhor dos quatro esquemas analisados para se utilizar em linhas fechadas de frango é o LPS + MC.

Liu, Henryon e Sørensen (2017) testam a hipótese de que estratégias de acasalamento utilizando dados genotípicos possuem um melhor desempenho na redução das taxas de endogamia. Isso quando comparados a métodos que utilizam somente

informações do pedigree. Para isso, foram feitas simulações estocásticas, comparando os sistemas de acasalamento de mínima coancestralidade e o sistema que visa minimizar a covariância entre as contribuições genéticas ancestrais (MCAC). Como ponto de referência também foram feitas simulações utilizando o sistema de acasalamento aleatório. Cinco esquemas reprodutivos diferentes foram testados, variando as estruturas familiares, com razões de acasalamento de 1, 2 ou 6 fêmeas/macho, tamanhos de ninhada de 10, 20 ou 60 descendentes e herdabilidades diferentes de 0,1 ou 0,4. As simulações foram divididas em três etapas, sendo a primeira etapa a geração de uma população fundadora, a segunda etapa a geração de uma população base e a terceira a geração de uma população selecionada. A população selecionada teve as 20 gerações seguintes simuladas e replicada 100 vezes. Os resultados obtidos por Liu, Henryon e Sørensen (2017) enfatizam a ideia de que acasalamentos planejados utilizando dados genotípicos propiciam taxas de endogamia mais baixas. É apresentado que MC e MCAC ao utilizarem informações genômicas atingiram uma taxa de endogamia de 6% a 22% menor do que ao utilizar dados do pedigree, sem comprometer os ganhos genéticos. Em comparação com o método RM foram obtidas taxas de endogamia de 20% a 44% menores utilizando MC e MCAC.

Henryon, Sørensen e Berg (2009), por meio de simulações estocásticas, realizaram a comparação dos sistemas de acasalamento MC e MCAC para seleções por truncamento. Foram feitas simulações com dois esquemas de acasalamento: hierárquico e fatorial. Com o hierárquico utiliza-se 20 machos e 120 fêmeas, formando uma relação 1:6, que resulta em 1.200 descendentes por geração. Para o modelo fatorial são selecionados 60 machos e 60 fêmeas, onde cada macho é acasalado com duas fêmeas e cada fêmea é acasalada com dois machos, resultando também em 1.200 descendentes por geração. Cada esquema foi replicado 100 vezes. Os resultados obtidos demonstraram um desempenho considerável para o método MCAC que apresentou de 4% a 8% menos consanguinidade se comparado com MC e até menos 28% se comparado com RM. MCAC obteve um ganho genético aproximado de MC. Mostra-se ainda que tanto MC quanto MCAC foram capazes de atrasar o início da endogamia para a 4<sup>o</sup> geração (no modelo hierárquico) e 5<sup>o</sup> geração (no modelo fatorial), enquanto que no modelo RM a consanguinidade poderia ser vista já na 2<sup>o</sup> geração. Com isto pode-se concluir que MCAC não só reduziu as taxas de endogamia como também permitiu ganho genético similar a MC.

Weigeland e Lin (2002) discutem a metodologia de contribuição ótima desenvolvida por Meuwissen (1997 *apud* WEIGELAND; LIN, 2002, p. 2377), aplicada

às cinco principais raças leiteiras dos EUA (*Ayrshire, Brown Swiss, Guernsey, Holstein e Jersey*). Busca-se o controle das taxas de consanguinidade, afetando minimamente o ganho genético. Foram utilizados dados fornecidos pelo Laboratório de Programas de Melhoramento Animal do Departamento de Agricultura dos Estados Unidos (USDA) para desenvolver uma matriz de relacionamento e um vetor de valores genéticos estimados para cada raça. Os resultados obtidos mostram que há perda genética ao se restringir a endogamia. Este fato aponta para necessidade de encontrar um equilíbrio entre o ganho genético e as taxas de endogamia desejadas. Ainda Weigeland e Lin (2002) afirmam que este equilíbrio é economicamente realizável e que severas restrições à endogamia podem ser problemáticas do ponto de vista econômico. Além de também apresentar a possibilidade de selecionar animais mais velhos como reprodutores.

A Tabela 3 sumariza os trabalhos encontrados em relação as abordagens que foram seguidas e ferramentas utilizadas. Como pode ser observado, a grande maioria dos trabalhos encontrados ((NOMURA; YONEZAWA, 1996), (PAIVA, 2010), (HONDA; NOMURA; MUKAI, 2005), (MORENO et al., 2011), (PRYCE; HAYES; GODDARD, 2012), (WINDIG; KAAL, 2008), (FERNÁNDEZ; TORO; CABALLERO, 2003), (NOMURA, 1999), (NOMURA et al., 2002), (LIU; HENRYON; SØRENSEN, 2017), (HENRYON; SØRENSEN; BERG, 2009)) busca encontrar meios para redução da consanguinidade e da coancestralidade, trazendo comparações entre métodos já existentes e comumente usados. Raramente é observada a proposta de uma nova solução como Nomura (1999) faz. Weigeland e Lin (2002), diferentemente dos demais trabalhos, busca encontrar o limiar entre os ganhos financeiros e perda de variabilidade genética, ressaltando a necessidade de se encontrar um limite para as restrições que devem ser impostas nos acasalamentos. É visto que tanto Fernández, Toro e Caballero (2003) quanto NOMURA et al. (2002) fazem citação ao algoritmo *simulated annealing* como um meio de encontrar uma melhor solução (se comparada com a utilização de programação linear) para a implementação prática do acasalamento de mínima coancestralidade.

## 4.2 Formalização do problema

Esta seção trata da resposta à segunda questão de pesquisa apresentada na Seção 3.1 – [Q2] Existem descrições formais do problema da minimização da coancestralidade em sistemas de acasalamento?

Nenhum dos trabalhos traz uma descrição formal do problema a ser resolvido,

Tabela 3 – Técnicas utilizadas

<b>Trabalho</b>	<b>Abordagem</b>	<b>Ferramenta/Método</b>
(FERNÁNDEZ; TORO; CABALLERO, 2003)	Comparativa	Não especificado
(WINDIG; KAAL, 2008)	Comparativa	Estimador genético WEDS
(WEIGELAND; LIN, 2002)	Analítica	Matriz de relacionamento e vetor de valores genéticos estimados
(HENRYON; SØRENSEN; BERG, 2009)	Comparativa	Simulação estocástica
(HONDA; NOMURA; MUKAI, 2005)	Comparativa	Equação de recorrência
(LIU; HENRYON; SØRENSEN, 2017)	Comparativa	Simulação estocástica
(MORENO et al., 2011)	Comparativa	Aplicação prática em ratos domésticos e utilização do programa endog (na versão v4.6)
(NOMURA; YONEZAWA, 1996)	Comparativa	Equação de recorrência
(NOMURA, 1999)	Comparativa	Simulação estocástica
(NOMURA et al., 2002)	Comparativa	Não especificado
(PAIVA, 2010)	Comparativa	Sistema computacional <i>GENESYS 2010</i>
(PRYCE; HAYES; GODDARD, 2012)	Comparativa	Ferramenta genômica <i>Bovine Genome Build 4.0</i>

Fonte: Autor (2022)

o que impede que abordagens analíticas sejam utilizadas e que uma solução seja, de fato, mostrada ótima para a questão. Exemplo de tratamento formal para problema de melhoramento é visto no trabalho de Motta (2021) que não lida com o mesmo problema, porém demonstra uma solução formal para o problema de otimização dos ganhos financeiros. Buscar uma descrição formal do problema e posteriormente definir sua complexidade, torna possível descobrir o seu limite de otimização e até mesmo encontrar uma solução ótima polinomial, dependendo de sua complexidade.

## 5 O ALGORITMO COANCESTRYMIN-SELECTION

### 5.1 Caracterização do problema

Os algoritmos apresentados neste capítulo são construídos a partir das relações de *coancestralidade*, *consanguinidade* e *índice de seleção*, conceitos-chave nos processos de melhoramento animal. A coancestralidade, ou relação de parentesco, é uma relação binária entre dois animais, enquanto que a consanguinidade é uma relação unária (ou um predicado) que diz respeito a um único animal. A consanguinidade de um animal é calculado a partir da coancestralidade existente entre seus pais. Enquanto que o índice de seleção é calculado a partir de uma média ponderada entre diversas características do indivíduo. No caso deste trabalho, utilizar-se-á o índice econômico de ciclo completo definido pela Embrapa Pecuária Sul. Essas relações são formalmente apresentadas nas Definições 1, 2 e 3, respectivamente.

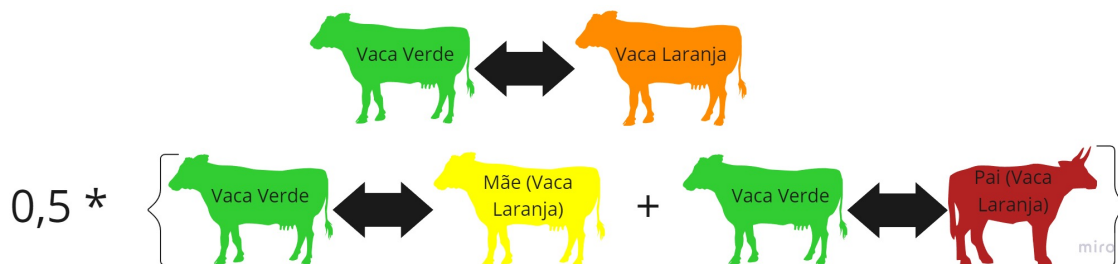
**Definição 1 (Coancestralidade)** (adaptado de (ELSTON; OLSON; PALMER, 2002) e (VLECK, 1993)) *Seja  $A$  um conjunto de animais, seja  $s : A \rightarrow \mathbb{N}$  a função que indica a safra de cada animal, e sejam  $\rho, \mu : A \rightarrow A \cup \{\perp\}$  as funções que retornam, para cada animal, seu pai e sua mãe. Se o pai ou a mãe não pertencem ao rebanho a função retorna o valor  $\perp$ . O valor da relação de coancestralidade, ou de parentesco, entre quaisquer dois animais  $x, y \in A$ , denotado por  $\psi(x, y)$ , é o valor retornado pela função  $\psi : A \times A \rightarrow \mathbb{R}^+$ , definida como se segue (assumindo  $s(x) \geq s(y)$ ):*

$$\psi(x, y) = \begin{cases} 0 & x \text{ e } y \text{ são indivíduos base} \\ 1 & x = y, x \text{ é um indivíduo base} \\ 1/2 & x \neq y, x \text{ é um indivíduo base e } y \text{ é seu filho} \\ 1/2[\psi(x, \rho(y)) + \psi(x, \mu(y))] & x \neq y \\ 1 + [1/2\psi(\rho(x), \mu(y))] & x = y \end{cases} \quad (1)$$

A coancestralidade, apresentada formalmente na Definição 1, é utilizada para calcular as matrizes de parentesco entre todo um grupo de indivíduos, como ilustrado na Figura 5. A dimensão da matriz de parentesco gerada sempre é o número de indivíduos elevado ao quadrado. Nesse esquema matricial se faz necessário que os indivíduos mais velhos e/ou que não possuem pais cadastrados, comumente chamados de “base”, ocupem

as primeiras posições da matriz (VLECK, 1993). Para implementação em algoritmo, é essencial que uma ordenação seja realizada de modo que indivíduos mais velhos ocupem as primeiras posições da matriz de coancestralidade. Um dos erros que se pode cometer é referenciar um pai que ainda não foi registrado, especialmente quando o algoritmo usado não foi preparado para lidar com essa situação.

Figura 2 – Ilustração da Equação (1) para  $x \neq y$



Fonte: Autor (2022)

Na Figura 2 busca-se ilustrar como se dá o cálculo da coancestralidade entre dois indivíduos diferentes, sendo estes a “Vaca Verde” e a “Vaca Laranja”. Supondo que a “Vaca Verde” seja mais velha que a “Vaca Laranja”. Para saber o coeficiente de parentesco entre as duas é necessário calcular o coeficiente de parentesco entre a “Vaca Verde” e a mãe da “Vaca Laranja” e somar com o coeficiente de parentesco entre a “Vaca Verde” e o pai da “Vaca Laranja”. Para então multiplicar por 0.5 e assim obter o valor de coancestralidade inicialmente procurado.

**Definição 2 (Consanguinidade)** *Seja  $A$  um conjunto de animais. O valor da consanguinidade, para qualquer animal  $x \in A$ , é dado pela função  $\Gamma : A \rightarrow \mathbb{R}^+$  cuja expressão é dada por:*

$$\Gamma(x) = \begin{cases} 0 & x \text{ é um indivíduo base} \\ \psi(\rho(x), \mu(x)) - 1 & \text{caso contrário} \end{cases} \quad (2)$$

*ou seja, o coeficiente de consanguinidade é calculado a partir da coancestralidade entre os pais do indivíduo  $x$ .*

A consanguinidade expressa o índice de parentesco entre os pais de um animal, e é calculado a partir das relações de parentesco existentes no rebanho.

A Figura 3 busca ilustrar a maneira que é calculada a consanguinidade de um indivíduo, segundo a Equação (2). Nesta figura temos que para descobrir a



Figura 3 – Ilustração da Equação (2)



Fonte: Autor (2022)

consanguinidade da “Vaca Verde” é necessário subtrair 1 do seu valor de coancestralidade entre si mesma, isto é, quando na Equação (1), temos  $x = y$ .

Figura 4 – Ilustração da Equação (1) dentro da Equação (2)



Fonte: Autor (2022)

Na Figura 4 é possível notar que a consanguinidade de um indivíduo nada mais é que a coancestralidade entre seus pais multiplicada por 0.5. Visto que o cálculo da coancestralidade de um indivíduo com si mesmo se dá pela coancestralidade entre seus pais, multiplicada por 0.5 e posteriormente somada a 1. Enquanto que sua consanguinidade é dada por este mesmo valor, porém subtraindo-se 1.

**Definição 3 (Índice de Seleção)** (FERREIRA; YOKOO; MOTTA, 2021) *Seja  $T$  um conjunto de características de animais. Um índice de seleção é um par  $\mathcal{I} = (T, w)$  onde  $w : T \rightarrow \mathbb{R}$  é a função de ponderação do índice.*

A função de ponderação do índice de seleção tem o propósito de determinar o peso a ser aplicado para cada característica inclusa no índice. Não há nenhum tipo de restrição quanto aos valores que a ponderação das características possam assumir. A vantagem do uso de índices nos processos de seleção é poder classificar os animais por meio de um único valor.

O índice de seleção usado neste trabalho é definido em (YOKOO et al., 2021) e contém os objetivos de seleção com seus valores econômicos em dólar americano (US\$): PAV (-0,24), OPG (-5,35), CARR (-20,88), taxa de prenhez (1,59), peso de carcaça quente (2,11) e FS (-3,14). A partir deste objetivos chegou-se a ponderação apresentada na Tabela 4. Note-se que os objetivos de seleção indicam o quanto (valor absoluto) uma característica é desejada (valor positivo) ou indesejada (valor negativo).

Tabela 4 – Índice econômico de ciclo completo (IEEC)

Objetivo de seleção	Critério de seleção	Ponderador
CARR	Contagem de carrapatos	17.13%
OPG	Contagem de ovos por grama de fezes	4,39%
PCQ	AOL	0.62%
	EG	13.70%
	EGP8	9.72%
PAV	Peso adulto da vaca	16.26%
PN	Peso ao nascer	20.79%
FS	Temperamento	17.39%

Fonte: (YOKOO et al., 2021)

**Definição 4 (Índice de Coancestralidade)** *Seja  $A = M \cup F \cup P$  um conjunto de animais, onde  $M$  é o conjunto de machos,  $F$  é o conjunto de matrizes e  $P$  é o conjunto de possíveis proles de acasalamentos entre pares  $(m, f) \in M \times F$ , com  $M \cap F \cap P = \emptyset$ ,  $\psi : A \times A \rightarrow \mathbb{R}^+$  a coancestralidade entre dois indivíduos. Sendo o índice de coancestralidade da possível prole do par  $(m, f) \in M \times F$ , denotado por  $\lambda : M \times F \rightarrow \mathbb{R}$ , calculado da seguinte forma:*

$$\lambda(m, f) = \sum_{i=1}^A \psi(z_{mf}, z_i) \quad (3)$$











*Sendo a possível prole do par  $(m, f) \in M \times F$ , denotada por  $z_{mf}$  e qualquer outro animal do rebanho, denotado por  $z_i$ .*

O índice de coancestralidade apresentado formalmente na Definição 4, se trata de um indicador de quão aparentado é um indivíduo com o restante do rebanho no qual ele poderá ser inserido.

Na Figura 5 busca-se ilustrar uma matriz de coancestralidade, considerando 3 indivíduos base, sendo eles o “Touro Vermelho”, a “Vaca Amarela” e o “Touro Azul”. A partir destes indivíduos há dois acasalamentos possíveis. Estes acasalamentos podem resultar na “Vaca Verde” ou na “Vaca Laranja”. É perceptível que cada valor desta matriz representa um grau de parentesco entre dois indivíduos. Sendo atribuído 0 para indivíduos não aparentados, 0.5 para indivíduos que possuem uma relação pai/filho ou mãe/filho e 0.25 para meio-irmãos. Enquanto que os valores  $\geq 1$  servem para indicar a consanguinidade de um indivíduo.

Possuindo os valores de coancestralidade entre cada indivíduo é possível calcular os índices de coancestralidade dos possíveis produtos e com isto buscar acasalamentos

Figura 5 – Ilustração da Equação (3)

Coefficiente de parentesco						Índice de coancestralidade
	1	0	0	0	0,5	X
	0	1	0	0,5	0,5	X
	0	0	1	0,5	0	X
	0	0,5	0,5	1	0,25	2,25
	0,5	0,5	0	0,25	1	2,25

Fonte: Autor (2022)

que resultem em indivíduos menos aparentados com o restante do rebanho. Na Figura 5 ambas as possibilidades resultam em indivíduos igualmente aparentados, cujo somatório de coancestralidade resultam em um mesmo valor.

Note-se que a relação de coancestralidade é uma relação simétrica. Ou seja,  $\psi(x, y) = \psi(y, x)$ . Dessa forma, é suficiente calcular somente os valores acima da diagonal principal da matriz que expressa esses valores.

**Definição 5 (Contribuição Individual)** (FERREIRA; YOKOO; MOTTA, 2021) *Seja A um conjunto de animais,  $\mathcal{I} = (T, w)$  um índice de seleção e  $v : T \times A \rightarrow \mathbb{R}$  uma função que retorna o valor de uma característica para cada animal no conjunto A. A função  $\iota^{\mathcal{I}} : A \rightarrow \mathbb{R}$  que para cada animal  $a \in A$ , retorna sua contribuição individual de acordo com o índice de seleção  $\mathcal{I}$ , é calculada como*

$$\iota^{\mathcal{I}}(a) = \sum_{k \in T} v(k, a) \cdot w(k) \quad (4)$$

*Quando o índice de seleção é entendido a partir do contexto, escrevemos  $\iota(a)$  em vez de  $\iota^{\mathcal{I}}(a)$ .*

A contribuição de acasalamento, apresentada formalmente na Definição 6, nada mais é que a contribuição individual esperada na prole entre um macho e uma fêmea. A contribuição de acasalamento é obtida através da média entre as contribuições econômicas individuais dos progenitores. Neste trabalho, é o IECC que está sendo utilizado como unidade de medida para mensurar a contribuição individual de cada indivíduo.

**Definição 6 (Contribuição de acasalamento)** (FERREIRA; YOKOO; MOTTA, 2021) *Seja  $A = M \cup F$  um conjunto de animais, onde  $M$  é o conjunto de machos e  $F$  é o conjunto de matrizes, com  $M \cap F = \emptyset$ ,  $\mathcal{I}$  o índice de seleção. A contribuição de acasalamento  $\pi : M \times F \rightarrow \mathbb{R}$  de um macho  $m \in M$  e de uma matriz  $f \in F$  é calculado da seguinte forma:*

$$\pi(m, f) = (\iota^{\mathcal{I}}(m) + \iota^{\mathcal{I}}(f))/2 \quad (5)$$

Onde  $\iota^{\mathcal{I}}$  é a contribuição individual de cada animal, dado um índice de seleção  $\mathcal{I}$ .

Quando somente o índice de seleção é levado em consideração, o valor da contribuição de acasalamento apresentada na Definição 6 é suficiente para que um algoritmo de otimização seja usado. Contudo, neste trabalho queremos também considerar a coancestralidade do rebanho. A Definição 7, a seguir, expressa a ideia de que o valor de um acasalamento deve levar também essa variável em consideração.

**Definição 7 (Contribuição ponderada de melhoramento)** *Seja  $A = M \cup F$  um conjunto de animais, em que  $M$  é o conjunto de machos e  $F$  é o conjunto de matrizes, com  $M \cap F = \emptyset$ . Seja  $\Pi_{|M| \times |F|}$  a matriz de contribuição, onde cada  $\pi_{ij}$  representa o índice de seleção esperado da prole resultante do par  $(m_i, f_j) \in M \times F$ , ou o valor  $\pi(m_i, f_j)$  (Definição 6). Seja  $\Lambda_{|M| \times |F|}$  a matriz de coancestralidade, em que cada  $\lambda_{ij}$  representa o índice de coancestralidade esperado da prole resultante do par  $(m_i, f_j) \in M \times F$ , ou o valor  $\lambda(m_i, f_j)$  (Definição 1). Sejam  $P_e, P_c \in \mathbb{R}$  o peso a ser aplicado no índice de seleção e o peso a ser aplicado no índice de coancestralidade, respectivamente. Seja  $\Gamma : M \times F \rightarrow \mathbb{R}^+$  a função que retorna consanguinidade esperada da prole resultante do par  $(m_i, f_j) \in M \times F$  (Definição 2). A contribuição ponderada de melhoramento  $v : M \times F \rightarrow \mathbb{R}$  de um macho  $m \in M$  e de uma matriz  $f \in F$  é calculado da seguinte forma:*

$$v(m, f) = \frac{\lambda(m, f) \cdot P_c - \pi(m, f) \cdot P_e}{P_c + P_e} \quad (6)$$

A Definição 7 une o índice de coancestralidade ao índice econômico esperado da possível prole, de forma que se possa realizar a otimização de ambos índices, permitindo a escolha da prioridade desejada que será dada para cada atributo, estabelecendo a importância relativa entre a contribuição econômica e a e minimização do índice de coancestralidade, dependendo das características do rebanho e objetivos de produção. Note-se que é possível determinar pesos que ignorem um desses aspectos, bastando considerar  $P_e = 0$  e  $P_c = 1$  para que somente a minimização do índice de coancestralidade

seja usado; trocando-se os valores, ignora-se a coancestralidade e considera-se apenas o índice de seleção.

A Definição 7 estabelece os valores de ponderação, mas ainda é necessário que os valores de entrada estejam normalizados ou que o elemento ponderador compense a diferença, para que a ponderação não seja enviesada, já que os valores absolutos do índice são usualmente muito maiores do que os valores de coancestralidade. A normalização consiste da operação em que, perante a multiplicação por uma constante adequada, se faz a norma de um vetor igual à unidade. De toda forma, a definição ainda pode ser usada, contanto que o usuário tome esse tipo de cuidado. O pacote de software que será construído terá o ferramental necessário para que os valores possam ser normalizados. A razão do sinal dos valores referentes a contribuição de acasalamento ser negativo é que o intuito é modelar o problema como uma minimização do índice de coancestralidade em conjunto com a contribuição de acasalamento. Como se quer o maior valor possível para a contribuição do índice, o problema equivalente é conseguir o menor valor possível da contribuição negativa.

**Definição 8 (Problema de minimização da contribuição ponderada de melhoramento)**

Seja  $A = M \cup F$  um conjunto de animais, sendo  $M$  e  $F$  o conjunto de machos e matrizes, respectivamente, de forma que  $M \cap F = \emptyset$ . Seja  $\Upsilon_{|M| \times |F|}$  a matriz de contribuições ponderadas de melhoramento, onde cada  $v_{ij}$  representa a média ponderada entre a contribuição e o índice de coancestralidade, que se é esperado da prole resultante do par  $(m_i, f_j) \in M \times F$ , ou o valor  $v(m_i, f_j)$  (Definição 7). Sejam  $t_{\max}, t_{\min} : M \rightarrow \mathbb{N}$ , respectivamente, as funções que indicam os limites máximo e mínimo de utilização de cada touro no processo de cruzamento. A saída do problema de minimização da contribuição ponderada de melhoramento, dada a entrada  $\Upsilon$ , é uma matriz binária  $\mathcal{B}^*$ , com dimensão  $|M| \times |F|$  onde cada elemento  $b^*_{ij}$  de  $\mathcal{B}^*$  é

$$b^*_{ij} = \begin{cases} 1 & \text{se o macho } i \text{ e a matriz } j \text{ devem acasalar} \\ 0 & \text{caso contrário} \end{cases} \quad (7)$$

sujeito às seguintes restrições:

$$\sum_{i=1}^{|M|} b^*_{ij} = 1, \text{ para cada matriz } j \in F \quad (8)$$

$$\sum_{j=1}^{|F|} b_{ij}^* \leq t_{\max}(i), \text{ para cada macho } i \in M \quad (9)$$

$$\sum_{j=1}^{|F|} b_{ij}^* \geq t_{\min}(i), \text{ para cada macho } i \in M \quad (10)$$

e de modo que, para qualquer outra matriz binária  $B$  com as mesmas dimensões, e que atenda às restrições acima, temos que

$$\sum_{i=1}^{|M|} \sum_{j=1}^{|F|} v_{ij} b_{ij} \geq \sum_{i=1}^{|M|} \sum_{j=1}^{|F|} v_{ij} b_{ij}^* \quad (11)$$

## 5.2 Modelo de programação linear/inteira

O problema apresentado na Definição 8 pode ser facilmente modelado como uma instância de um problema de programação inteira. Como essa classe de problemas é NP-difícil, a solução comumente usada é apresentar o problema como uma instância de um problema de programação linear, em que as restrições de valores inteiros são relaxadas para intervalos de interesse. A solução de problemas de programação linear em que as variáveis podem tomar valores reais usualmente leva menos tempo que problemas inteiros (HILLIER; LIEBERMAN, 2013).

O modelo em programação linear do problema de minimização da contribuição ponderada de melhoramento fará uso de um conjunto de variáveis  $x_{ij}$ , com  $i = 1, 2, \dots, |M|$  e  $j = 1, 2, \dots, |F|$  para os conjuntos de reprodutores  $M$  e matrizes  $F$  do rebanho já selecionados entre os animais com condições de reproduzir. Os valores dessas variáveis indicarão se o macho  $i \in M$  e a fêmea  $j \in F$  deverão ou não reproduzir entre si, caracterizando o problema como programação inteira e binária. A função objetivo do problema pode ser expressa como:

$$\text{MIN } y = \sum_{i=1}^{|M|} \sum_{j=1}^{|F|} v_{ij} x_{ij} \quad (12)$$

em que  $v_{ij}$  é o valor computado conforme a Definição 7.

As restrições resultam diretamente da definição do problema:

$$\sum_{i=1}^{|M|} x_{ij} = 1, \text{ para cada matriz } j \in F \quad (13)$$

$$\sum_{j=1}^{|F|} x_{ij} \leq t_{\max}(i), \text{ para cada macho } i \in M \quad (14)$$

$$\sum_{j=1}^{|F|} x_{ij} \geq t_{\min}(i), \text{ para cada macho } i \in M \quad (15)$$

em que a primeira restrição assegura que uma fêmea será acasalada com exatamente um macho, e que os limites máximos e mínimos de cada macho são obedecidos.

Para o relaxamento do problema, em vez de assumir-se que as variáveis  $x_{ij}$  poderão ter os valores 0 ou 1, exclusivamente, usa-se as seguintes restrições nos valores das variáveis, assegurando um valor entre esses dois limites:

$$x_{ij} \geq 0, \text{ para cada } i = 1, 2, \dots, |M| \text{ e } j = 1, 2, \dots, |F|, \text{ e} \quad (16)$$

$$x_{ij} \leq 1, \text{ para cada } i = 1, 2, \dots, |M| \text{ e } j = 1, 2, \dots, |F| \quad (17)$$

Note-se que, sem essas duas últimas restrições, o modelo não fica bem definido, pois valores sucessivamente menores das variáveis  $x_{ij}$  seriam melhores, fazendo o algoritmo entrar em loop.

O modelo de programação linear acima descrito apresenta  $|M||F|$  variáveis, com  $|F| + 2|M| + 2|M||F|$  restrições.

Para ilustrar a formalização do problema, suponha que possuamos um banco de dados referente à raça Brangus do qual são escolhidas 6 fêmeas (f1, f2, f3, f4, f5 e f6) e 2 machos (m1 e m2) para acasalar. Considerando um limite máximo de consanguinidade de 5,0% (0.05). A partir destas informações pode-se estimar 12 possíveis proles (produtos).

Os índices de coancestralidade ( $\Lambda$ ) de cada possível produto são apresentados na Tabela 5 e calculados conforme a Definição 4.

Os coeficientes de consanguinidade ( $\Gamma$ ) de cada possível produto podem ser vistos na Tabela 6 e foram calculados de acordo com a Definição 2. Os IECC estimados para cada possível produto são apresentados na Tabela 7. O IECC corresponde à contribuição individual de cada animal. Para estimar o IECC de um possível produto foi calculada uma média entre o IECC do pai e o IECC da mãe deste possível produto, como mostrado na

Tabela 5 – Exemplo do problema de acasalamento - índice de coancestralidade

	<b>f1</b>	<b>f2</b>	<b>f3</b>	<b>f4</b>	<b>f5</b>	<b>f6</b>
<b>m1</b>	33.79	31.80	31.99	32.47	39.66	35.58
<b>m2</b>	23.60	24.79	24.98	25.46	32.65	28.57

Fonte: Autor (2022)

Definição 6.

Tabela 6 – Exemplo do problema de acasalamento - coeficiente de consanguinidade

	<b>f1</b>	<b>f2</b>	<b>f3</b>	<b>f4</b>	<b>f5</b>	<b>f6</b>
<b>m1</b>	0.125	0	0	0	0	0
<b>m2</b>	0	0	0	0	0	0

Fonte: Autor (2022)

Tabela 7 – Exemplo do problema de acasalamento - IECC

	<b>f1</b>	<b>f2</b>	<b>f3</b>	<b>f4</b>	<b>f5</b>	<b>f6</b>
<b>m1</b>	12.350	12.345	12.340	12.340	12.320	12.320
<b>m2</b>	12.300	11.015	11.010	11.010	10.990	10.990

Fonte: Autor (2022)

Na Tabela 8 é vista a média ponderada de melhoramento de cada possível produto que foi obtida através da Definição 7, utilizando os valores do índice de coancestralidade do possível produto e do IECC estimado, ambos devidamente normalizados. Neste caso a normalização ajustou as faixas de valor entre  $-1$  e  $1$ .

É perceptível que um dos possíveis produtos na Tabela 8 possui como média ponderada de melhoramento o valor  $\infty$ . Este símbolo é usado para indicar que foi aplicado um alto valor como “penalidade”, pois seu coeficiente de consanguinidade ultrapassa o limite desejado. Na prática, é utilizado o maior valor inteiro disponível na representação pela máquina, no caso o valor constante associado (*Machine\$integer.max*) da linguagem R.

Os objetivos do modelo, de acordo com este exemplo em que temos  $|F| = 6$  e  $|M| = 2$ , tornam-se as seguintes expressões:



Tabela 8 – Exemplo do problema de acasalamento - média ponderada de melhoramento

	<b>f1</b>	<b>f2</b>	<b>f3</b>	<b>f4</b>	<b>f5</b>	<b>f6</b>
<b>m1</b>	$\infty$	-0.48574	-0.47023	-0.44034	0.02206	-0.23199
<b>m2</b>	-0.96324	0.05571	0.07122	0.10111	0.56351	0.30946

Fonte: Autor (2022)

$$\mathbf{MIN} y_c = \begin{aligned} & c_{11}x_{11} + c_{12}x_{12} + c_{13}x_{13} + c_{14}x_{14} + c_{15}x_{15} + c_{16}x_{16} + \\ & c_{21}x_{21} + c_{22}x_{22} + c_{23}x_{23} + c_{24}x_{24} + c_{25}x_{25} + c_{26}x_{26} \end{aligned} \quad (18)$$

Sendo a expressão 18 referente à minimização da coancestralidade.

$$\mathbf{MAX} y_e = \begin{aligned} & e_{11}x_{11} + e_{12}x_{12} + e_{13}x_{13} + e_{14}x_{14} + e_{15}x_{15} + e_{16}x_{16} + \\ & e_{21}x_{21} + e_{22}x_{22} + e_{23}x_{23} + e_{24}x_{24} + e_{25}x_{25} + e_{26}x_{26} \end{aligned} \quad (19)$$

Sendo a expressão 19 referente à maximização do IECC.

Para unir os dois objetivos representados nas expressões 18 e 19, é necessário inverter o sinal do objetivo que se quer maximizar, neste caso a expressão 19 referente ao IECC terá seu sinal invertido. Com isso tem-se duas funções que deseja-se minimizar. Pois ao inverter o sinal e minimizar valores negativos da função referente ao IECC, estaremos maximizando seus valores absolutos.

Porém cada expressão ainda possui faixas de valores distintas, o que torna necessário trabalhar com dados normalizados. A partir deste ponto é feita a aplicação de uma média ponderada entre os objetivos, aplicando os devidos “pesos” para cada atributo, sendo  $P_e$  o peso aplicado ao IECC e  $P_c$  o peso aplicado ao somatório de coancestralidade. Com isto obtém-se:

$$\begin{aligned}
& (-1 \cdot e_{11} \cdot P_e + c_{11} \cdot P_c) / (P_e + P_c) \cdot x_{11} + \\
& (-1 \cdot e_{12} \cdot P_e + c_{12} \cdot P_c) / (P_e + P_c) \cdot x_{12} + \\
& (-1 \cdot e_{13} \cdot P_e + c_{13} \cdot P_c) / (P_e + P_c) \cdot x_{13} + \\
& (-1 \cdot e_{14} \cdot P_e + c_{14} \cdot P_c) / (P_e + P_c) \cdot x_{14} + \\
& (-1 \cdot e_{15} \cdot P_e + c_{15} \cdot P_c) / (P_e + P_c) \cdot x_{15} + \\
\text{MIN } y = & (-1 \cdot e_{16} \cdot P_e + c_{16} \cdot P_c) / (P_e + P_c) \cdot x_{16} + \\
& (-1 \cdot e_{21} \cdot P_e + c_{21} \cdot P_c) / (P_e + P_c) \cdot x_{21} + \\
& (-1 \cdot e_{22} \cdot P_e + c_{22} \cdot P_c) / (P_e + P_c) \cdot x_{22} + \\
& (-1 \cdot e_{23} \cdot P_e + c_{23} \cdot P_c) / (P_e + P_c) \cdot x_{23} + \\
& (-1 \cdot e_{24} \cdot P_e + c_{24} \cdot P_c) / (P_e + P_c) \cdot x_{24} + \\
& (-1 \cdot e_{25} \cdot P_e + c_{25} \cdot P_c) / (P_e + P_c) \cdot x_{25} + \\
& (-1 \cdot e_{26} \cdot P_e + c_{26} \cdot P_c) / (P_e + P_c) \cdot x_{26}
\end{aligned} \tag{20}$$

Substituindo os valores da Tabela 5 e da Tabela 7 na expressão 20, considerando um peso de 0.5 tanto para o IECC ( $P_e = 0.5$ ) quanto para o índice de coancestralidade ( $P_c = 0.5$ ), o que significa dar igual prioridade para ambas funções, tem-se a expressão:

$$\begin{aligned}
& 2147483647 \cdot x_{11} + \\
& (-1 \cdot 0.9926 \cdot P_e + 0.02117 \cdot P_c) / (P_e + P_c) \cdot x_{12} + \\
& (-1 \cdot 0.9852 \cdot P_e + 0.04483 \cdot P_c) / (P_e + P_c) \cdot x_{13} + \\
& (-1 \cdot 0.9852 \cdot P_e + 0.10461 \cdot P_c) / (P_e + P_c) \cdot x_{14} + \\
& (-1 \cdot 0.9558 \cdot P_e + 1 \cdot P_c) / (P_e + P_c) \cdot x_{15} + \\
\text{MIN } y = & (-1 \cdot 0.9558 \cdot P_e + 0.49190 \cdot P_c) / (P_e + P_c) \cdot x_{16} + \\
& (-1 \cdot 0.9264 \cdot P_e + -1 \cdot P_c) / (P_e + P_c) \cdot x_{21} + \\
& (-1 \cdot -0.9632 \cdot P_e + -0.85180 \cdot P_c) / (P_e + P_c) \cdot x_{22} + \\
& (-1 \cdot -0.9705 \cdot P_e + -0.82814 \cdot P_c) / (P_e + P_c) \cdot x_{23} + \\
& (-1 \cdot -0.9705 \cdot P_e + -0.76836 \cdot P_c) / (P_e + P_c) \cdot x_{24} + \\
& (-1 \cdot -1 \cdot P_e + 0.12702 \cdot P_c) / (P_e + P_c) \cdot x_{25} + \\
& (-1 \cdot -1 \cdot P_e + -0.38107 \cdot P_c) / (P_e + P_c) \cdot x_{26}
\end{aligned} \tag{21}$$

Na expressão 21 é visto o valor 2147483647, este é o maior valor possível de ser associado a uma variável inteira na infraestrutura de hardware/software usada na execução deste trabalho. Como dito anteriormente, isso de faz necessário para aplicação da “penalidade” para os casos cuja consanguinidade ultrapassa o limite estabelecido.

As restrições do problema, neste exemplo, são instanciadas como

$$\begin{aligned}
x_{11} + x_{21} &= 1 \\
x_{12} + x_{22} &= 1 \\
x_{13} + x_{23} &= 1 \\
x_{14} + x_{24} &= 1 \\
x_{15} + x_{25} &= 1 \\
x_{16} + x_{26} &= 1 \\
x_{11} + x_{12} + x_{13} + x_{14} + x_{15} + x_{16} &\leq t_{\max}(1) \\
x_{21} + x_{22} + x_{23} + x_{24} + x_{25} + x_{26} &\leq t_{\max}(2) \\
x_{11} + x_{12} + x_{13} + x_{14} + x_{15} + x_{16} &\geq t_{\min}(1) \\
x_{21} + x_{22} + x_{23} + x_{24} + x_{25} + x_{26} &\geq t_{\min}(2)
\end{aligned} \tag{22}$$

Tomando como valores máximos de uso dos machos  $m_1$  e  $m_2$  como, respectivamente,  $t_{\max}(1) = 3$  e  $t_{\max}(2) = 4$  e seus respectivos valores mínimo como  $t_{\min}(1) = 0$  e  $t_{\min}(2) = 1$ , as restrições do problema tomam a seguinte forma:

$$\begin{aligned}
x_{11} + x_{21} &= 1 \\
x_{12} + x_{22} &= 1 \\
x_{13} + x_{23} &= 1 \\
x_{14} + x_{24} &= 1 \\
x_{15} + x_{25} &= 1 \\
x_{16} + x_{26} &= 1 \\
x_{11} + x_{12} + x_{13} + x_{14} + x_{15} + x_{16} &\leq 3 \\
x_{21} + x_{22} + x_{23} + x_{24} + x_{25} + x_{26} &\leq 4 \\
x_{11} + x_{12} + x_{13} + x_{14} + x_{15} + x_{16} &\geq 0 \\
x_{21} + x_{22} + x_{23} + x_{24} + x_{25} + x_{26} &\geq 1
\end{aligned} \tag{23}$$

Além das restrições acima, as restrições nos valores dos  $x_{ij}$  são as seguintes:

$$\begin{aligned}
x_{11} \geq 0 \quad x_{21} \geq 0 \quad x_{11} \leq 1 \quad x_{21} \leq 1 \\
x_{12} \geq 0 \quad x_{22} \geq 0 \quad x_{12} \leq 1 \quad x_{22} \leq 1 \\
x_{13} \geq 0 \quad x_{23} \geq 0 \quad x_{13} \leq 1 \quad x_{23} \leq 1 \\
x_{14} \geq 0 \quad x_{24} \geq 0 \quad x_{14} \leq 1 \quad x_{24} \leq 1 \\
x_{15} \geq 0 \quad x_{25} \geq 0 \quad x_{15} \leq 1 \quad x_{25} \leq 1 \\
x_{16} \geq 0 \quad x_{26} \geq 0 \quad x_{16} \leq 1 \quad x_{26} \leq 1
\end{aligned} \tag{24}$$

O modelo utilizado para solucionar este problema de programação linear descrito acima é similar ao utilizado por Motta (2021), que usava exclusivamente o índice econômico, com exceção de que existem agora restrições referentes à utilização mínima de cada macho e a consanguinidade da prole foi incluída no modelo. O método consiste em combinar técnicas de programação linear e *branch-and-bound* para transformar um problema de programação inteira binária em um problema de programação linear passível de ser executado em tempo polinomial. Utiliza-se primeiramente a técnica conhecida como “relaxamento” permitindo que as variáveis do problema assumam valores do conjunto [0,1]. Então, é possível utilizar o método Simplex (HILLIER; LIEBERMAN, 2013) para encontrar um espaço de soluções inteiras para o problema. Por fim, é possível utilizar a técnica *branch-and-bound* para encontrar as soluções inteiras dentro deste espaço de soluções “relaxadas”. Os métodos utilizados são melhor explicados no Capítulo 2, nas Seções 2.3 e 2.4.

Tabela 9 – Solução do exemplo do problema de acasalamento

	<b>f1</b>	<b>f2</b>	<b>f3</b>	<b>f4</b>	<b>f5</b>	<b>f6</b>
<b>m1</b>	0	0	0	1	1	1
<b>m2</b>	1	1	1	0	0	0

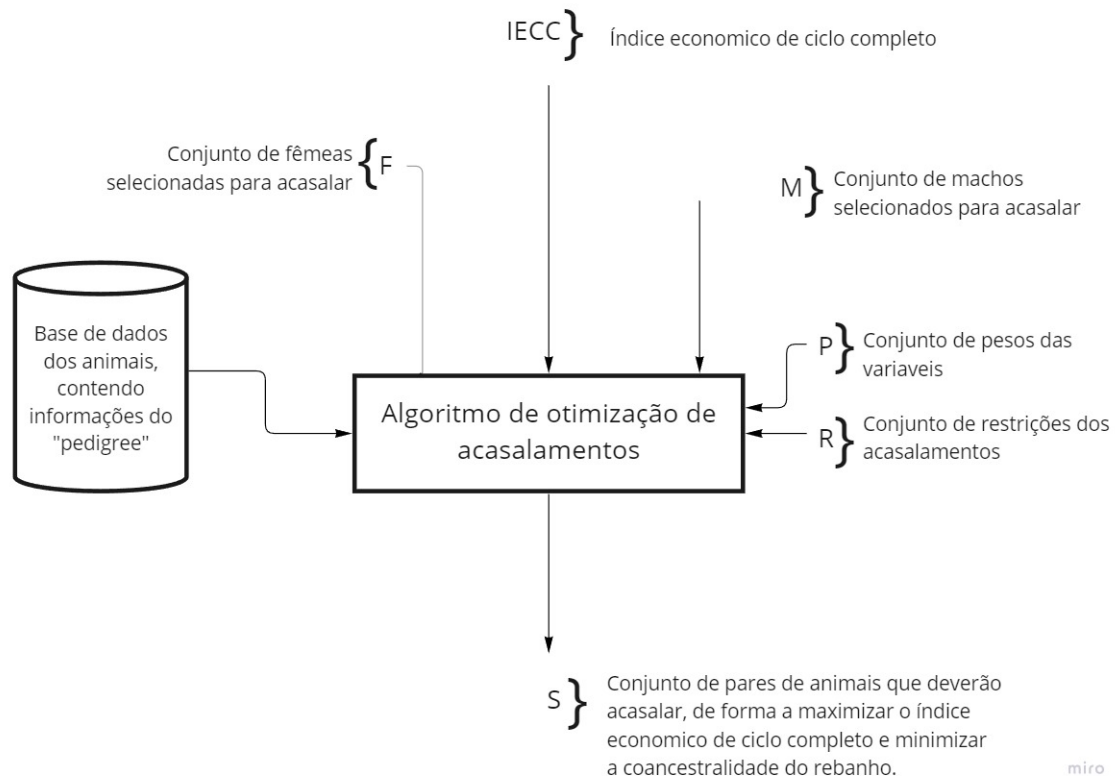
Fonte: Autor (2022)

Na Tabela 9 é possível visualizar a solução proposta para os acasalamentos, respeitando as restrições e minimizando o valor da função objetivo, através da indicação dos valores '1' para os pares m1f4, m1f5, m1f6, m2f1, m2f2 e m2f3. Enquanto os valores '0' indicam os pares que não deverão acasalar.

### 5.3 Codificação da solução

O algoritmo desenvolvido traz consigo a codificação do modelo de problema que foi expresso na Definição 8 e modelado e exemplificado na Seção 5.2. A Figura 6 traz consigo representações das entradas e da saída esperada.

Figura 6 – Esquema de entradas e saída do algoritmo CoancestryMIN-Selection



Fonte: Autor (2022), adaptado de (MOTTA, 2021)

Os problemas aqui tratados se referem à programação inteira binária. O que significa que o resultado esperado é uma resposta do tipo “sim-ou-não” para cada possibilidade de acasalamento. Para evitar a complexidade do problema associado à programação inteira, aplicou-se uma solução através do pacote *lpSolve* no R, que utiliza a técnica *Simplex* no modelo de programação linear relaxado junto da técnica *branch-and-bound* para a descoberta das soluções inteiras a partir do espaço de soluções restrito devolvido pelo algoritmo *simplex*.

Para problemas de decisão (ou problemas que requerem respostas do tipo “sim ou não”) é recomendado utilizar o argumento “all.bin=TRUE” ao chamar a função “lp” da biblioteca *lpSolve*. Isso fará com que todas as variáveis do problema sejam tratadas como binárias (0 ou 1), garantindo que a solução final seja uma solução inteira válida.

Quando o argumento “all.bin=TRUE” é utilizado na chamada da função “lp”, ela automaticamente habilita a opção de usar o *branch-and-bound*. Isso faz com que a função “lp” utilize o algoritmo simplex para resolver o problema inicial e, quando é encontrada uma solução fracionária, ele utiliza o *branch-and-bound* para encontrar a solução ótima inteira.

No caso da função “lp” da biblioteca *lpSolve*, a complexidade é de tempo  $O(n^2 * m)$ , onde  $n$  é o número de variáveis e  $m$  é o número de restrições do problema de programação linear. A complexidade se dá pelo fato de que a cada iteração, todas as restrições e todas as variáveis são avaliadas para determinar a melhor escolha de entrada e saída da base. Isso resulta em uma complexidade de  $O(n * m)$  para cada iteração, e como o número de iterações é limitado pelo número de variáveis no problema, a complexidade total é  $O(n^2 * m)$ .

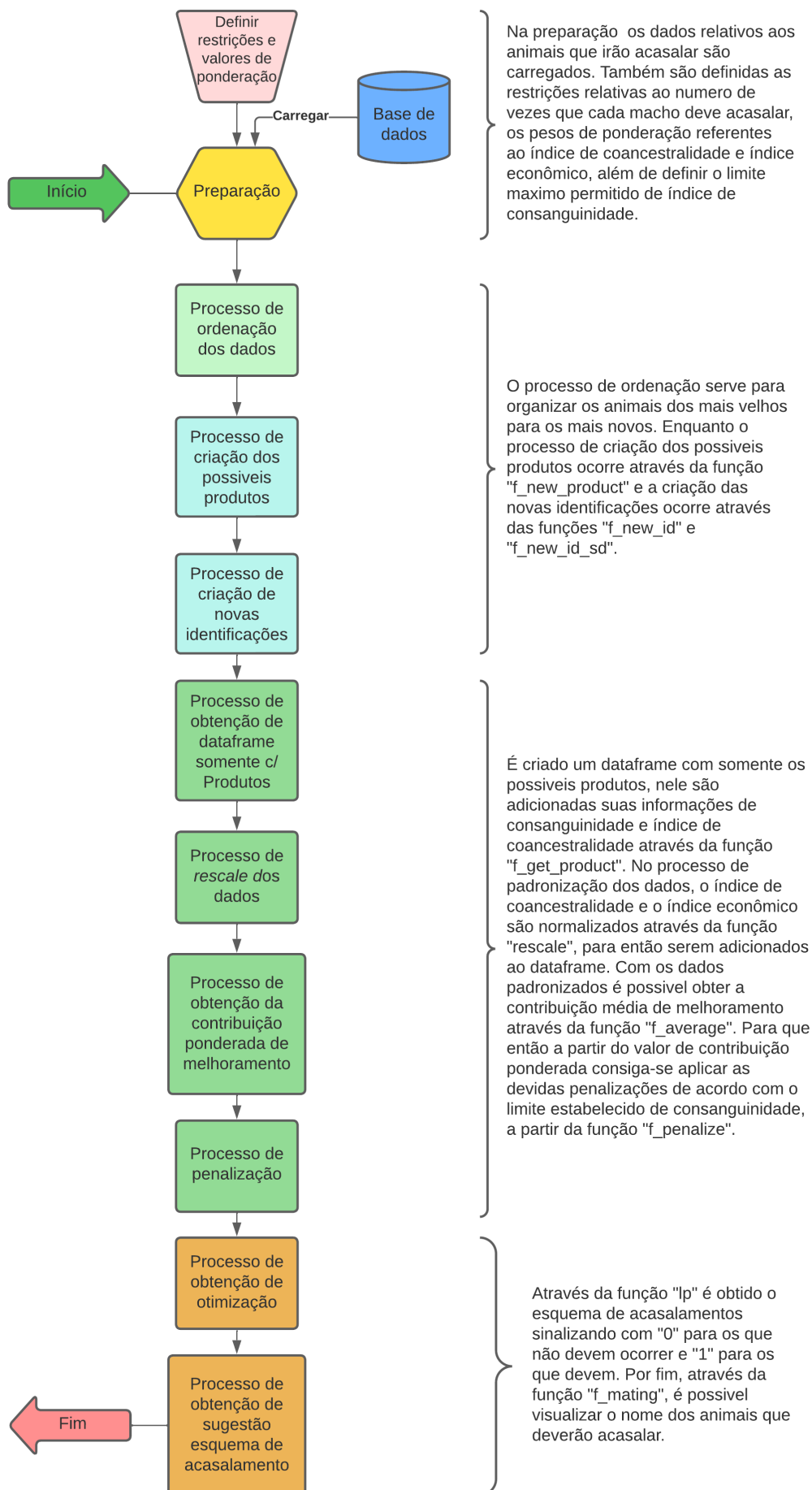
A função de complexidade  $O(n^2 * m)$  indica que o tempo de execução do algoritmo aumenta proporcionalmente à multiplicação do número de variáveis  $n$  elevado ao quadrado com o número de restrições  $m$ . Isso significa que, quanto maior o número de variáveis e restrições no problema, maior será o tempo de execução do algoritmo. No caso do problema da atribuição de pares para acasalamento, se temos  $f$  matrizes e  $r$  reprodutores, o número de variáveis do problema  $n = f * r$  e o número de restrições é  $f + 2r$ , como já visto na definição do problema. Dessa forma, temos uma complexidade total de  $O(f^2 * r^2 * (f + 2r)^2)$ . Mesmo considerando somente número de fêmeas (que é usualmente bastante superior ao número de machos), temos que a ordem de complexidade do algoritmo é um polinômio de grau 4. Com uma complexidade dessa ordem, o tamanho do problema que conseguimos tratar em um tempo aceitável é bastante limitado: um problema com cerca de 1000 animais animais pode ser resolvido em minutos, mas um problema com 10.000 animais precisaria de vários meses de execução. Como problemas com milhares de animais não são incomuns, tem-se a indicação de que otimizações são necessárias.

O algoritmo simplex para um problema de programação linear inteira relaxado e o algoritmo *branch-and-bound* é uma das estratégias mais utilizadas para resolver problemas de programação linear binária e é considerado como um dos métodos mais eficientes para encontrar soluções ótimas (AARDAL; WEISMANTEL; WOLSEY, 2000; SCHRIJVER, 1998; HORST; PARDALOS; THOAI, 1995; PAPADIMITRIOU, 1994). A estratégia, no caso do problema tratado neste trabalho, pelas suas características, limita o tamanho da entrada. Essa é uma indicação de que outras estratégias podem ser

necessárias (heurísticas ou meta-heurísticas) para que um algoritmo mais eficiente possa ser construído.

A Figura 7 traz um fluxograma que busca descrever o funcionamento do algoritmo desenvolvido, de forma a facilitar o entendimento. Os códigos desenvolvidos encontram-se no Apêndice A, para referência.

Figura 7 – Fluxograma do algoritmo CoancestryMIN-Selection





## 6 RESULTADOS E DISCUSSÃO

Os testes executados tem como objetivo verificar como o tempo de execução do algoritmo muda conforme o tamanho da entrada (ou seja o número de machos e fêmeas) aumenta. Para isso, o experimento consiste em medir o tempo de execução ao aumentar gradualmente o tamanho da entrada, apresentando os resultados graficamente. O procedimento ajudará a entender a complexidade média do tempo de execução do algoritmo implementado. Se a função tiver a forma de um polinômio, isso será um sinal importante de que o problema é resolvível computacionalmente.

A função “Sys.time” do pacote base do R foi utilizada para medir o tempo de execução do algoritmo nos experimentos. De acordo com a documentação oficial do R, essa função retorna a data e hora do sistema no momento em que é chamada, em formato de objeto "POSIXt". A função “Sys.time” foi escolhida por ser simples de usar. É suficiente registrar o tempo inicial e final e subtraí-los para estimar o tempo de execução. Também foi utilizada a função “mean” do pacote base do R, cuja função é calcular a média aritmética de um conjunto de números.

A base de dados para realização dos testes é a mesma detalhada anteriormente no Capítulo 3, na Seção 3.4.

Para a automatização dos testes foi criado um algoritmo de testes que pode ser consultado no Apêndice C. O algoritmo de testes possui dois *scripts* diferentes sendo um deles para testar o desempenho do algoritmo quando aumentando-se o numero de machos e se mantém fixo o numero de fêmeas e outro para a situação inversa, quando se mantém fixo o numero de machos e aumenta-se o numero de fêmeas.

Os *scripts* do algoritmo de testes geram arquivos do tipo *.csv* como saída. Nestes arquivos, encontra-se uma tabela com informações relevantes, incluindo o ponderador do índice econômico (Pe), o ponderador do índice de coancestralidade (Pc), a quantidade de animais utilizados (que é incrementada durante o teste) e o tempo médio de execução, expressado em segundos, como pode ser visto na Figura 8.

Na Figura 8 é visto a coluna “nfemale” que contem o numero de fêmeas utilizado e na coluna “mean\_time” temos a média do tempo de 10 execuções.

O propósito deste experimento era avaliar a eficiência do algoritmo CoancestryMIN-Selection. O tempo de execução do sistema foi testado em relação ao número de animais selecionados para acasalamento, com o objetivo de determinar o limite aproximado de animais a partir do qual o sistema torna-se inviável. Os resultados

Figura 8 – Exemplo da saída gerada pelo arquivo do algoritmo de testes

Pe	Pc	nfemales	mean_time
1	0	100	0.0213344573974609
1	0	200	0.102746796607971
1	0	300	0.312365674972534
1	0	400	0.491599917411804
1	0	500	0.512093305587769
1	0	600	0.67582004070282
1	0	700	1.02086951732635
1	0	800	1.27377510070801

Fonte: Autor (2023)

e os parâmetros usados nos testes do algoritmo CoancestryMIN-Selection podem ser encontrados na Figura 9 e na Figura 10. As colunas representam o número de machos, o número de fêmeas, o número de variáveis, o número de restrições e o tempo médio de execução por configuração Pe/Pc, respectivamente. Foram realizados 20 testes, cada um com 10 execuções diferentes, utilizando amostras distintas da base de dados.

Na Figura 9 é visto os resultados dos testes considerando um número fixo de machos, enquanto incrementa-se o número de fêmeas. Devido a limitações na base de dados utilizada, o número de fêmeas foi restrito a 2.000. Além disso, a escolha de não incrementar o número de machos foi tomada considerando que isso resultaria em um aumento significativo do número de variáveis, o que poderia exceder a capacidade de memória do computador responsável por realizar os testes.

A limitação na capacidade de memória do computador utilizado nos testes se deve quanto ao cálculo da matriz de parentesco entre todos os animais do rebanho, incluindo todos os possíveis descendentes gerados pelas combinações possíveis de fêmeas e machos selecionados. Seria mais eficiente se esses dados referentes ao coeficiente de parentesco já estivessem armazenados em uma base de dados, para que o algoritmo possa se concentrar somente na estimativa dos melhores pares de animais que devem ser acasalados.

Na Figura 10 é visto os resultados dos testes considerando um número fixo de fêmeas, enquanto incrementa-se o número de machos. Nestes experimentos, os valores foram estabelecidos de forma a capturar os pontos necessários para estimar a curva de desempenho na Figura 12. O número de fêmeas e machos utilizados nestes testes foi

Figura 9 – Resultados e parâmetros dos testes variando nº de fêmeas

Nº de MACHOS	Nº de FÊMEAS	Variáveis	Restrições	1/0	0.8/0.2	0.6/0.4	0.4/0.6	0.2/0.8	0/1
10	100	1000	120	0,02	0,02	0,02	0,03	0,02	0,02
10	200	2000	220	0,10	0,08	0,08	0,10	0,09	0,08
10	300	3000	320	0,31	0,19	0,25	0,20	0,18	0,21
10	400	4000	420	0,49	0,50	0,58	0,43	0,38	0,47
10	500	5000	520	0,51	0,64	0,78	0,80	1,11	0,65
10	600	6000	620	0,68	0,85	1,17	1,32	1,43	0,96
10	700	7000	720	1,02	1,05	1,58	1,81	1,98	1,26
10	800	8000	820	1,27	1,10	2,33	1,99	2,02	1,61
10	900	9000	920	1,44	1,56	3,57	2,45	2,78	2,83
10	1000	10000	1020	1,86	1,71	4,05	3,29	3,77	3,64
10	1100	11000	1120	2,17	1,77	7,05	4,31	3,84	4,25
10	1200	12000	1220	2,84	5,44	7,98	5,49	6,51	5,23
10	1300	13000	1320	3,10	6,26	8,03	7,50	9,18	7,52
10	1400	14000	1420	3,19	7,23	9,56	7,60	10,25	10,97
10	1500	15000	1520	3,36	9,31	12,95	8,49	11,18	22,26
10	1600	16000	1620	3,85	10,93	14,46	12,61	12,16	25,98
10	1700	17000	1720	4,55	13,99	15,10	13,56	13,64	37,35
10	1800	18000	1820	5,57	14,64	22,32	15,86	22,40	38,57
10	1900	19000	1920	6,35	15,28	26,34	16,49	26,19	39,94
10	2000	20000	2020	6,73	25,10	29,33	16,56	37,92	50,58

Fonte: Autor (2023)

limitado devido à viabilidade computacional, especialmente quando o tempo de execução ultrapassou 12 horas durante a tentativa de testar 45 machos para 200 fêmeas.

A Figura 11 apresenta a relação entre o tempo de execução e o incremento do número de fêmeas para cada configuração Pe/Pc. A configuração 0/1 prioriza a minimização da coancestralidade no rebanho, enquanto que a configuração 1/0 tem como objetivo principal a maximização do índice econômico. Observa-se que o tempo médio de execução aumenta à medida que os ponderadores são direcionados para minimizar a coancestralidade do rebanho.

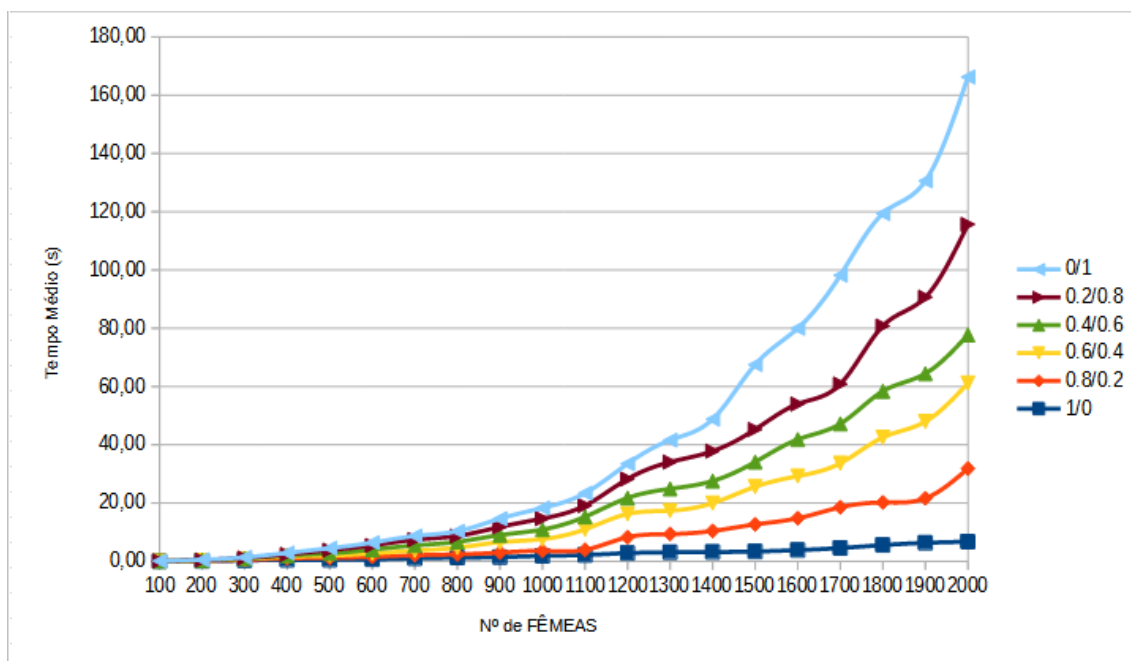
A Figura 12 apresenta a relação entre o tempo de execução e o incremento do número de machos para cada configuração Pe/Pc. Nota-se que, ao se priorizar a minimização da coancestralidade do rebanho, há um aumento no tempo médio de execução. Entretanto, as configurações 0/1 e 0.2/0.8 apresentam desempenho semelhante e praticamente idêntico.

Figura 10 – Resultados e parâmetros dos testes variando nº de machos

Nº de Machos	Nº de Fêmeas	Variáveis	Restrições	1/0	0.8/0.2	0.6/0.4	0.4/0.6	0.2/0.8	0/1
11	200	2200	222	0,10	0,10	0,09	0,12	0,11	0,11
12	200	2400	224	0,10	0,11	0,12	0,12	0,12	0,11
13	200	2600	226	0,13	0,12	0,13	0,13	0,13	0,12
14	200	2800	228	0,16	0,14	0,15	0,17	0,13	0,13
15	200	3000	230	0,19	0,14	0,15	0,17	0,14	0,14
16	200	3200	232	0,20	0,15	0,16	0,20	0,14	0,14
17	200	3400	234	0,24	0,16	0,17	0,20	0,16	0,14
18	200	3600	236	0,24	0,17	0,19	0,22	0,16	0,15
19	200	3800	238	0,27	0,18	0,22	0,24	0,21	0,17
20	200	4000	240	0,38	0,19	0,23	0,27	0,21	0,18
21	200	4200	242	0,40	0,21	0,24	0,41	0,24	0,18
22	200	4400	244	0,45	0,22	0,25	0,60	0,25	0,19
23	200	4600	246	0,45	0,24	0,26	0,79	0,28	0,21
24	200	4800	248	0,51	0,29	0,27	1,41	0,28	0,23
25	200	5000	250	0,63	0,30	0,28	1,45	0,33	0,27
26	200	5200	252	0,87	0,32	0,29	1,59	0,37	0,28
27	200	5400	254	0,88	0,33	0,31	1,67	2,83	0,29
28	200	5600	256	0,88	0,34	0,36	2,47	3,64	0,30
29	200	5800	258	1,05	0,37	0,37	5,23	4,25	0,32
30	200	6000	260	1,26	0,51	0,40	18,09	25,65	0,33

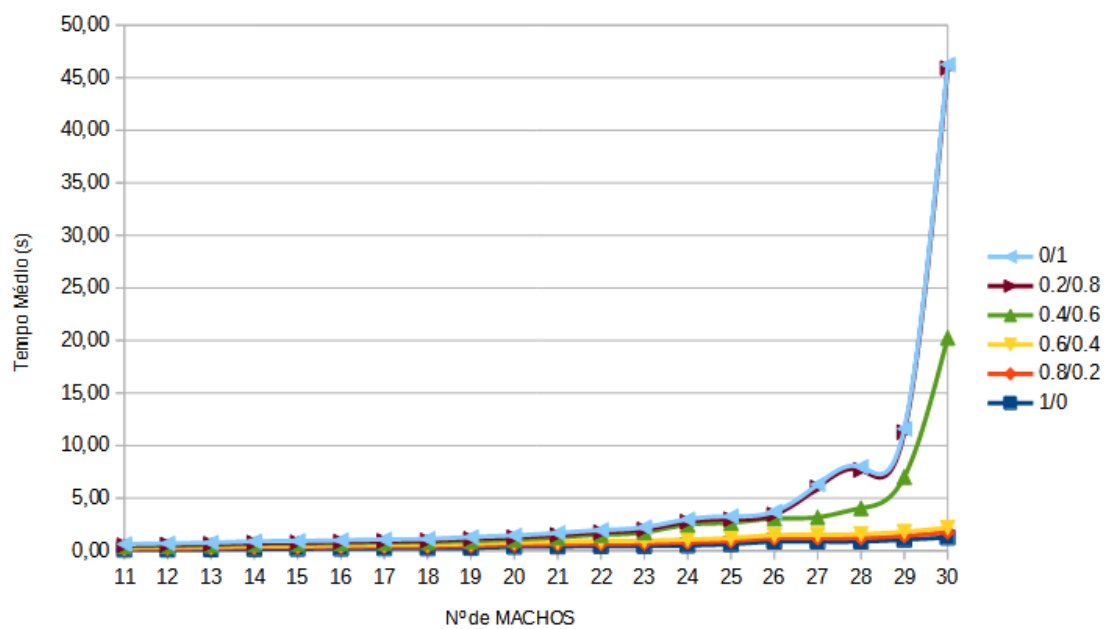
Fonte: Autor (2023)

Figura 11 – Resultados e parâmetros dos testes variando nº de fêmeas



Fonte: Autor (2023)

Figura 12 – Resultados e parâmetros dos testes variando n° de machos



Fonte: Autor (2023)

## 7 MODELO DE DADOS

### 7.1 Banco de dados baseado em grafos

O algoritmo CoancestryMIN-Selection está correntemente implementado de forma a receber os dados de entrada a partir de planilhas eletrônicas. Planilhas, contudo, passam a ter tratamento difícil quando a quantidade de dados é volumosa. Além dos problemas de desempenho, encontrar erros de inserção de dados em planilhas é uma atividade bastante custosa. Dessa forma, percebeu-se a necessidade de um banco de dados adequado para o armazenamento dos dados referente aos animais e suas relações de parentesco.

O algoritmo CoancestryMIN-Selection necessita de toda a base de dados relativa ao *pedigree* dos animais que irão acasalar. Os cálculos realizados para estimar o coeficiente de parentesco entre cada animal e o índice de coancestralidade da possível prole são realizados dentro do próprio algoritmo a cada execução. Porém, este não é um procedimento viável ou necessário para lidar com informações de milhares de animais, visto que as relações de parentesco entre os indivíduos não se alteram com o tempo. Assim, seria mais adequado que todas as relações de parentesco entre cada par de animais fossem registradas de maneira persistente, em um banco de dados. Também seria mais adequada a possibilidade de realizar a estimativa do índice de coancestralidade via requisição no próprio banco de dados.

Modelos de dados relacionais podem ser facilmente construídos para implementação dessa proposta. Contudo, as relações de coancestralidade são binárias e possuem a propriedade de simetria, o que torna o problema adequado para utilização do modelo de banco de dados baseado em grafos.

Grafos são entidades matemáticas formalmente definidas como um par ordenado  $G = (V, E \subseteq V \times V)$ , em que  $V$  é um conjunto de vértices e  $E$  é uma relação binária sobre o conjunto  $V$ , denominada conjunto de arestas. Grafos podem ser representados graficamente por nodos (vértices) e setas que os conectam (arestas). O fato de grafos expressarem relações entre elementos<sup>1</sup> serviu de motivação para o desenvolvimento de bancos de dados baseados em grafos, com os métodos usuais *Create*, *Read*, *Update*, e *Delete* (usualmente denotadas pela sigla CRUD). Nesse tipo de banco de dados o foco está nos relacionamentos, o que facilita a busca não só de elementos indexados por chaves,

---

<sup>1</sup>Relações entre múltiplos elementos também podem ser representados por hipergrafos.

como no caso dos bancos relacionais. Sistemas de gerenciamento de banco de dados relacionais exigem a criação de conexões entre as entidades utilizando chaves estrangeiras ou outros artifícios, que não favorecem a busca de relacionamentos, somente de elementos particulares do banco. Ao reunir os conceitos simples de nós e relacionamentos em estruturas conectadas, os bancos de dados de grafos nos permitem construir modelos que mapeiam de perto o domínio do problema. Os modelos criados a partir deste método geralmente são mais simples e mais expressivos do que aqueles produzidos por bancos de dados relacionais tradicionais e NOSQL (Not Only SQL) (ROBINSON; WEBBER; EIFREM, 2015).

## 7.2 Modelo do banco de melhoramento

O sistema de gerenciamento de banco de dados Neo4j<sup>2</sup> foi utilizado para o desenvolvimento de um modelo de banco de dados que atenda às demandas para o armazenamento de dados de animais.

O uso deste banco de dados visa auxiliar problemas existentes no armazenamento de dados de animais, tais como machos referenciados como mães de outros indivíduos, *pedigree* circular (caso em que um indivíduo é apontado como pai de seu ancestral) e indivíduos que possuem data de nascimento anterior a de seus pais. Embora nenhuma base esteja isenta de inserção errada de dados, é mais fácil identificar os problemas no momento da inserção, a tempo de correção pelo produtor dos dados. Além disso, este banco de dados também tem como objetivo armazenar informações precisas sobre o coeficiente de parentesco entre cada indivíduo.

A base para criação do banco de dados de melhoramento foram os dados fornecidos pela EMBRAPA Pecuária Sul em formato de planilha .csv, no formato já apresentado anteriormente na Seção 3.4. Para desenvolver o banco, foi utilizada a linguagem de consulta Cypher<sup>3</sup>, que é considerada por Robinson, Webber e Eifrem (2015) a linguagem de consulta de grafos mais fácil de aprender, além de uma ótima base para aprender sobre grafos.

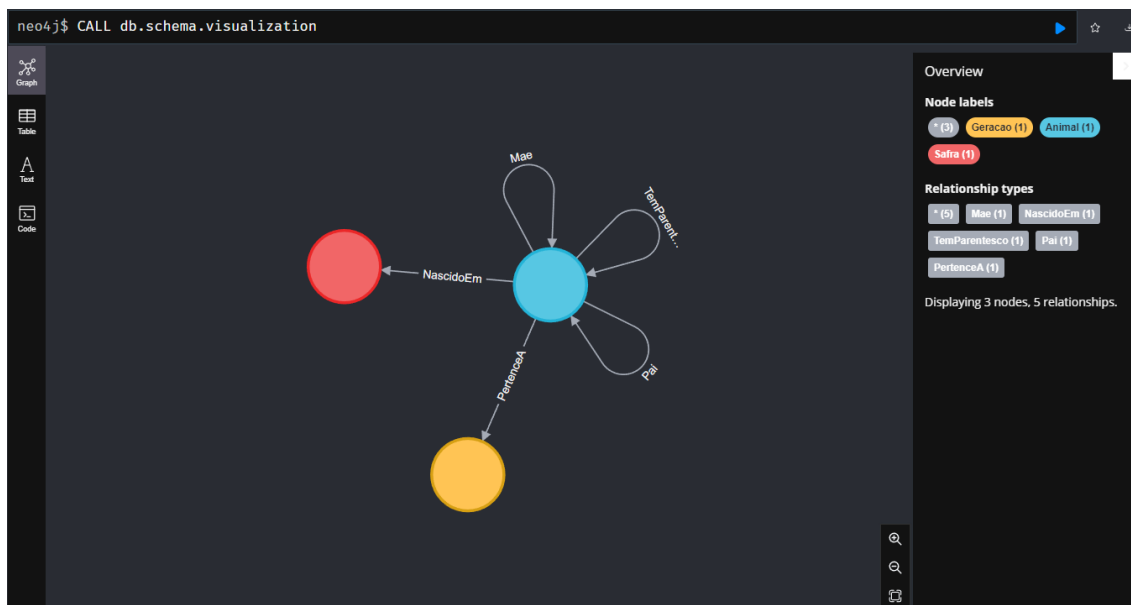
Na Figura 13 é possível visualizar as entidades e os relacionamentos existentes no modelo de banco de dados que foi desenvolvido. Existem três tipos de nós: Animal (apresentado na figura com a cor azul), Safra (apresentado na figura com a cor vermelha)

---

<sup>2</sup><<https://neo4j.com/>>

<sup>3</sup><<https://neo4j.com/developer/cypher/>>

Figura 13 – Modelo de dados



Fonte: Autor (2023)

e Geracao (apresentado na figura com a cor amarela). Os nodos podem ser enriquecidos com rótulos, assim como os arcos, que expressarão as relações no grafo. O nodo Animal possui as propriedades ID, IECC e Sexo, que são o equivalente a atributos no modelo relacional. A Safra tem uma única propriedade “Ano”. Geracao possui uma única propriedade “Numero”, cujo propósito é definir o número da geração à qual o animal pertence. Os nós do tipo “Animal” possuem relações (arcos) entre si que indicam sua paternidade, maternidade e coeficiente de parentesco.

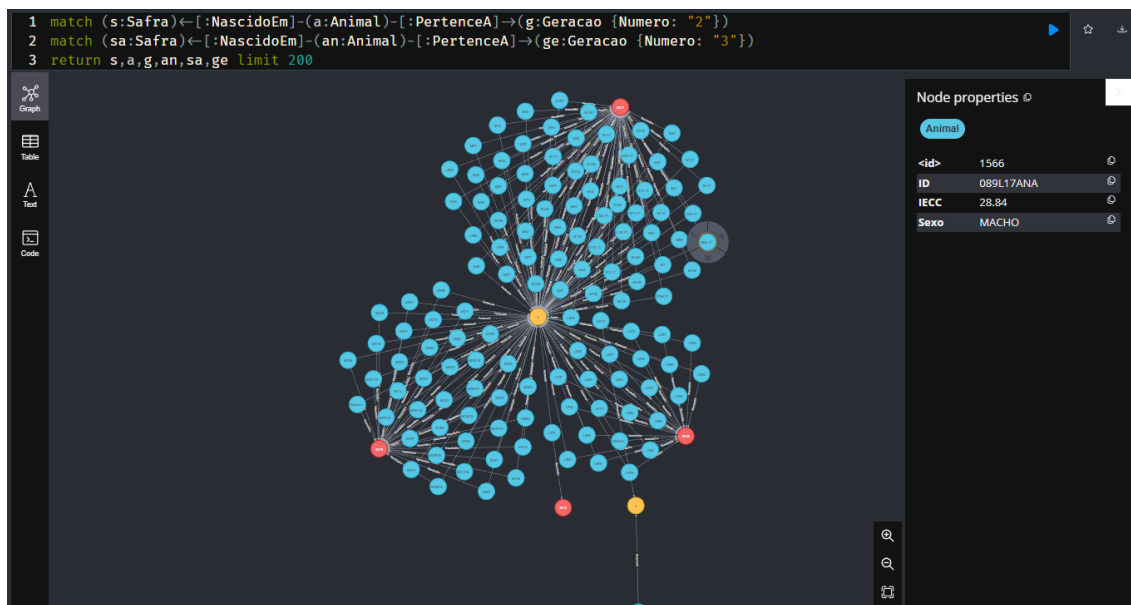
A Figura 14 traz uma amostra de como os nodos e as suas relações estão dispostos no banco de dados. Note-se que cada valor do banco é um nodo específico, o que torna a busca por valores e por ligações entre valores direta, diferentemente do modelo relacional.

Animais apontam para seus filhos a partir das relações “Pai” e “Mãe” indicando sua relação com o individuo apontado. Os animais apontam para sua geração e sua safra por meio das relações “PertenceA” e “NascidoEm” respectivamente. Informações relativas à paternidade dos animais e ao ano de seu nascimento foram adaptadas para o modelo baseado em grafos, passando a ser representados como relações e não como propriedades. As requisições desenvolvidas para elaboração deste banco de dados encontram-se no Apêndice B, para referência.

A Figura 15 traz uma amostra da disposição dos nodos do tipo “Animal” em relação aos nodos do tipo “Safra”, para uma melhor visualização da disposição e relacionamento entre os dados.



Figura 14 – Recorte do banco de dados populado com dados da EMBRAPA

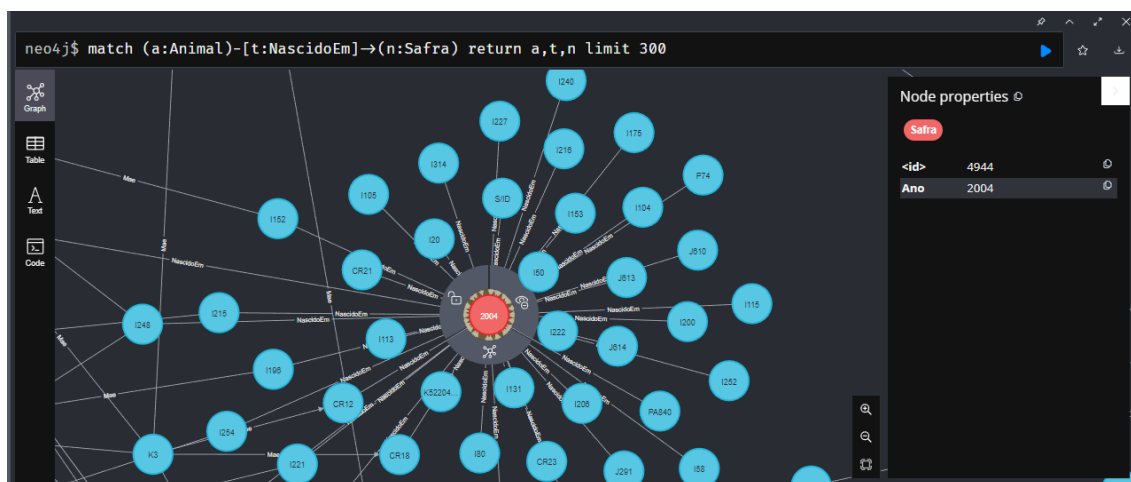


Fonte: Autor (2023)

Sendo o coeficiente de parentesco uma relação binária é desejável que estas informações sejam armazenadas em um banco de dados baseado em grafos, pois neste modelo as relações entre dois indivíduos são priorizadas diferentemente de um banco de dados relacional. Uma das maneiras para realizar o armazenamento do coeficiente de parentesco entre dois indivíduos em um banco de dados convencional, seria criar uma tabela de duas colunas para cada animal constando o ID do outro animal e seu valor de parentesco. Isto seria custoso no quesito de armazenamento, além de dificultar a inserção de novos animais. Novos animais inseridos no banco não só precisariam de sua própria tabela de referencia indicando seu parentesco com todos os demais animais como também precisariam ser inseridos nas tabelas dos animais já registrados. O uso de bancos de dados baseados em grafos é especialmente útil para armazenar relações de parentesco. Esta abordagem simplifica a adição de novos indivíduos ao banco de dados, pois é possível criar um novo relacionamento entre dois animais, definindo um coeficiente de parentesco entre eles.

Existem diversas vantagens associadas ao uso de bancos de dados relacionais baseados em grafos em comparação aos bancos de dados tradicionais. Em primeiro lugar, os grafos oferecem grande flexibilidade para lidar com relacionamentos entre entidades, possibilitando a criação de modelos de dados mais precisos. Além disso, os bancos de dados orientados a grafos podem lidar com grandes volumes de dados com uma maior facilidade, o que torna o processamento mais rápido. Por último, os bancos de dados

Figura 15 – Amostra do banco de dados - nodos do tipo “Animal” e “Saфра”



Fonte: Autor (2023)

relacionais baseados em grafos permitem a consulta e gerenciamento de várias fontes de dados de maneira simples e eficiente (DEMIR; ERKAN, 2017; ZHANG et al., 2018).

## 8 CONSIDERAÇÕES FINAIS

### 8.1 Conclusões

Os processos de melhoramento animal buscam as melhores técnicas de seleção e melhores arranjos no processo de acasalamento, visando melhorias genéticas e, por consequência, maiores ganhos financeiros. Contudo, um dos principais desafios é manter as taxas de consanguinidade do rebanho a um nível mínimo ao longo das gerações, para não comprometer nem a saúde dos animais e nem os ganhos financeiros.

Nos trabalhos relacionados fica evidenciada a importância da aplicação de técnicas que visem o controle da endogamia e coancestralidade. São comparados diferentes sistemas, em sua maioria buscando evitar o relacionamento entre irmãos completos e meio-irmãos, utilizando o sistema de acasalamento compensatório ou então fracionando o rebanho em diferentes grupos e rotacionando os machos entre eles. Sempre com o objetivo de se ter maior estabilidade das taxas de consanguinidade e coancestralidade ao longo do tempo.

Nenhum dos trabalhos vistos traz uma solução descrita de maneira formal. A maioria das afirmações se dá em relação a comparações feitas entre simulações de diferentes métodos. É mostrada a superioridade de um método sobre o outro, mas jamais a comprovação de que aquele método é efetivamente o melhor em todos os casos possíveis. A maioria dos pesquisadores responsáveis por abordar este tema geralmente possuem formação em áreas das Ciências da Vida (Biologia, Genética, Medicina Veterinária, Zootecnia e afins), justificando assim a comum ausência de definições formais. Formalizar esse problema permitirá abordagens mais rigorosas de investigação.

Neste trabalho são apresentadas formalmente as definições de coancestralidade, consanguinidade, índice econômico e dos problemas relacionados ao processo de seleção e sistemas de acasalamento. A formalização adequada para o problema do acasalamento permitiu o desenvolvimento de um algoritmo que visa a obtenção de uma solução ótima, resultando em uma recomendação de acasalamento que ocasione as próximas gerações de um rebanho terem um menor parentesco entre si e um maior valor econômico.

O algoritmo desenvolvido pode ser parametrizado para que o peso relativo entre o índice econômico (que deve ser maximizado) e a consanguinidade da nova geração (que deve ser minimizada) sejam balanceados por percentuais. Dessa forma, o algoritmo

pode ser usado para quaisquer uma das duas funções isoladas ou qualquer combinação delas. Os resultados dos testes realizados indicam que o problema da otimização do índice de coancestralidade apresenta uma complexidade superior à otimização do índice econômico.

Um banco de dados orientado a grafos foi desenvolvido com o intuito de melhorar o armazenamento dos dados referentes ao parentesco entre indivíduos. Como foi mencionado anteriormente, é a partir do coeficiente de parentesco que é possível estimar o índice de coancestralidade da prole potencial. Armazenar essas informações em um banco de dados específico permitirá que o algoritmo CoancestryMIN-Selection possa se concentrar exclusivamente no melhoramento animal e não precise gastar tempo tratando os dados previamente. Além disso, o uso do banco de dados orientado a grafos facilitaria a inserção de novos indivíduos e a gestão das relações de parentesco, permitindo uma análise mais precisa e eficiente do índice de coancestralidade.

Para a realização deste trabalho foi necessário estudar áreas completamente diferentes das proporcionadas na rotina acadêmica do curso de Engenharia de Computação, tais como conceitos básicos de genética, funcionamento de um sistema de acasalamento e de seleção, importância de se controlar as taxas de consanguinidade em um rebanho e demais conteúdos relacionados a agropecuária. É interessante ver como áreas tão distintas possam interagir entre si e com isso abrir a possibilidade de realizar avanços significativos na área das Ciências Agrárias.

## 8.2 Trabalhos futuros

A continuidade deste trabalho deverá seguir na pesquisa de métodos para se otimizar a capacidade de armazenamento de dados, referente às informações de *pedigree* de cada animal. Acredita-se que um banco de dados que comporte os valores de coancestralidade entre cada animal, possibilite obter índices de coancestralidade ( $\Lambda$ ) com maior precisão. No estudo da literatura, percebe-se que existe espaço para o desenvolvimento de algoritmos de cálculo de consanguinidade mais eficientes dos que os que estão sendo correntemente utilizados.

O modelo de banco de dados desenvolvido deverá ser integrado ao algoritmo de forma que a comunicação ocorra dentro do próprio ambiente R. Também deverão ser desenvolvidas as requisições para estimar os valores de índice de coancestralidade da possível prole. Trabalhar com estes dados armazenados em banco de dados e não

em *dataframes* do R como é feito atualmente, permitira manejar um número maior de informações o que resultará em uma maior precisão dos dados. Com o banco de dados finalizado, o algoritmo CoancestryMIN-Selection contará somente com a responsabilidade de realizar a estimativa dos pares que devem acasalar, abrindo mão da necessidade de realizar a estimativa do índice de coancestralidade da possível prole previamente.

Os tempos de execução do algoritmo indicam que a construção de algoritmos que não dependam do modelo de programação linear são necessários. O problema original possui um tamanho de entrada que corresponde à multiplicação do número de fêmeas pelo número de machos. Isso faz com que o grau de qualquer polinômio que descreva a complexidade do algoritmo de solução será multiplicado por dois. Polinômios de grau alto não são de utilidade prática, a menos que o tamanho da entrada seja bastante restrito. Dessa forma, entre os trabalhos futuros estão o desenvolvimento de algoritmos baseados em heurísticas que permitam um melhor desempenho, mantendo-se a solução ótima do problema, se isso for possível.

## REFERÊNCIAS

- AARDAL, K.; WEISMANTEL, R.; WOLSEY, L. A. **Branch-and-cut-and-price: An exact algorithm for the capacitated vehicle routing problem**. [S.l.]: Springer, 2000.
- BATES, D.; MAECHLER, M.; MAECHLER, M. M. Package ‘matrix’. **R package version**, p. 1–2, 2017.
- BATES, D.; VAZQUEZ, A. I. **pedigreemm: Pedigree-based mixed-effects models. R package version 0.3-3**. 2015.
- BERKELAAR, M. et al. Interface to lp\_solve v. 5.5 to solve linear/integer programs. **R package**, 2020.
- CORMEN, T. H. et al. **Algoritmos: teoria e prática**. 2. ed. [S.l.]: Editora Campus, 2002. 296 p.
- DEMIR, E. M.; ERKAN, G. Graph databases: A survey on technologies, solutions and applications. **Computer Science Review**, Elsevier, v. 24, p. 1–27, 2017.
- DERMEVAL, D.; BITTENCOURT, J. A. P. M. C. I. I. Mapeamento sistemático e revisão sistemática da literatura em informática na educação. In: \_\_\_\_\_. **Metodologia de Pesquisa Científica em Informática na Educação: Abordagem Quantitativa**. [S.l.]: CEIE/SBC, 2020. v. 2, cap. 3, p. 1–26.
- DOWLE, M. et al. Package data.table; extension of ‘data.frame’. **R (>= 3.1.0), MPL-2.0| file LICENSE**, 2021.
- EDDELBUETTEL, D. et al. Package ‘rcpp’. 2022.
- ELER, J. Teorias e métodos em melhoramento genético animal: sistemas de acasalamento. **Pirassununga: FZEA/USP**, 2017.
- ELSTON, R. C.; OLSON, J. M.; PALMER, L. **Biostatistical genetics and genetic epidemiology**. [S.l.]: John Wiley & Sons, 2002. v. 1.
- FERNÁNDEZ, J.; TORO, M.; CABALLERO, A. Fixed contributions designs vs. minimization of global coancestry to control inbreeding in small populations. **Genetics**, Oxford University Press, v. 165, n. 2, p. 885–894, 2003.
- FERREIRA, A. P. L. On the problem of compensatory mating in animal breeding. In: **Anais do VI Workshop-Escola de Informática Teórica (WEIT 2021)**. Bagé, RS: [s.n.], 2021. p. 96–103.
- FERREIRA, A. P. L. Metodologia científica. Programa de Pós-graduação em Computação Aplicada, Universidade Federal do Pampa. Notas de aula. 2022.
- FERREIRA, A. P. L.; YOKOO, M. J.-I.; MOTTA, B. E. T. On the problem of optimal mating in animal breeding. In: **Simposio Latinoamericano de Teoría Computacional, Proceedings of the XLVII Conferencia Latinoamericana de Informática (CLEI 2021)**. San José, Costa Rica: IEEE, 2021. p. 1–7. DOI 10.1109/CLEI53233.2021.9640023.

FILHO, K. E. O melhoramento genético e os cruzamentos em bovino de corte. **Embrapa Gado de Corte-Documents (INFOTECA-E)**, Campo Grande, MS: EMBRAPA-CNPGC, 1997., 1997.

FILHO, K. E. Melhoramento genético animal no Brasil: fundamentos, história e importância. **Embrapa Gado de Corte-Documents (INFOTECA-E)**, Campo Grande, MS: Embrapa Gado de Corte, 1999., 1999.

GAREY, M. R.; JOHNSON, D. S. **Computers and Intractability: A Guide to the Theory of NP-Completeness (Series of Books in the Mathematical Sciences)**. 1st. ed. [S.l.]: W. H. Freeman, 1979. ISBN 0716710455.

HENRYON, M.; SØRENSEN, A.; BERG, P. Mating animals by minimising the covariance between ancestral contributions generates less inbreeding without compromising genetic gain in breeding schemes with truncation selection. **Animal**, Elsevier, v. 3, n. 10, p. 1339–1346, 2009.

HILLIER, F. S.; LIEBERMAN, G. J. **Introdução à pesquisa operacional**. [S.l.]: McGraw Hill Brasil, 2013.

HONDA, T.; NOMURA, T.; MUKAI, F. Prediction of inbreeding in commercial females maintained by rotational mating with partially isolated sire lines. **Journal of Animal Breeding and Genetics**, Wiley Online Library, v. 122, n. 5, p. 340–348, 2005.

HORST, R.; PARDALOS, P.; THOAI, N. Branch-and-bound methods. **Handbook of Global Optimization**, Springer, v. 1, p. 431–495, 1995.

IEZZI, G. et al. **Fundamentos de matemática elementar**. [S.l.]: Atual, 1995.

KITCHENHAM, B. **Procedures for Performing Systematic Reviews**. Keele, 2004.

KLIK, M.; COLLET, Y. fst: lightning fast serialization of data frames for R. **R package version 0.9.8**, 2022.

LIU, H.; HENRYON, M.; SØRENSEN, A. Mating strategies with genomic information reduce rates of inbreeding in animal breeding schemes without compromising genetic gain. **Animal**, Cambridge University Press, v. 11, n. 4, p. 547–555, 2017.

MORENO, A. et al. Restricting inbreeding while maintaining selection response for weight gain in *Mus musculus*. **Journal of Animal Breeding and Genetics**, Wiley Online Library, v. 128, n. 4, p. 276–283, 2011.

MOTTA, B. E. T. da. **BrangusSelection: algoritmo de otimização para seleção de acasalamentos de bovinos com foco na maximização dos ganhos financeiros**. Dissertação (Mestrado) — Programa de Pós-graduação em Computação Aplicada, Universidade Federal do Pampa, 2021.

NOMURA, T. A mating system to reduce inbreeding in selection programmes: theoretical basis and modification of compensatory mating. **Journal of Animal Breeding and Genetics**, Wiley Online Library, v. 116, n. 5, p. 351–361, 1999.

NOMURA, T. et al. Optimization of selection and mating schemes in closed broiler lines. **Animal science journal**, Wiley Online Library, v. 73, n. 6, p. 435–443, 2002.

NOMURA, T.; YONEZAWA, K. A comparison of four systems of group mating for avoiding inbreeding. **Genetics Selection Evolution**, EDP Sciences, v. 28, n. 2, p. 141–159, 1996.

PAIVA, A. L. d. C. Avaliação de métodos de controle de endogamia utilizando dados simulados. Universidade Federal de Viçosa, 2010.

PAPADIMITRIOU, C. H. **Computational complexity**. [S.l.]: Addison-Wesley Longman Publishing Co., Inc., 1994.

PRYCE, J.; HAYES, B.; GODDARD, M. Novel strategies to minimize progeny inbreeding while maximizing genetic gain using genomic information. **Journal of dairy science**, Elsevier, v. 95, n. 1, p. 377–388, 2012.

ROBINSON, I.; WEBBER, J.; EIFREM, E. **Graph databases: new opportunities for connected data**. [S.l.]: "O'Reilly Media, Inc.", 2015.

SCHRIJVER, A. **Theory of linear and integer programming**. [S.l.]: John Wiley Sons, 1998.

SIMÕES, M. R. S. et al. Breeding objectives of brangus cattle in brazil. **Journal of Animal Breeding and Genetics**, v. 137, n. 2, p. 177–188, 2020. DOI 10.1111/jbg.12415. Disponível em: <https://onlinelibrary.wiley.com/doi/abs/10.1111/jbg.12415>.

VERRIER, E.; COLLEAU, J. J.; FOULLEY, J. L. Long-term effects of selection based on the animal model blup in a finite population. **Theoretical and Applied Genetics**, Springer, v. 87, n. 4, p. 446–454, 1993.

VLECK, L. van. **Selection Index and Introduction to Mixed Model Methods**. CRC-Press, 1993. ISBN 9780849387623. Disponível em: <https://books.google.com.br/books?id=mtsqAQAAMAAJ>.

WEIGELAND, K.; LIN, S. Controlling inbreeding by constraining the average relationship between parents of young bulls entering ai progeny test programs. **Journal of dairy science**, Elsevier, v. 85, n. 9, p. 2376–2383, 2002.

WICKHAM, H.; SEIDEL, D. scales: Scale functions for visualization. **R package version**, v. 1, n. 1, p. 678, 2020.

WICKHAM, H.; WICKHAM, M. H. Package 'dplyr'. **Obtido Httpscran Rproject Orgwebpackagesdplyrdplyr Pdf**, 2022.

WINDIG, J.; KAAL, L. An effective rotational mating scheme for inbreeding reduction in captive populations illustrated by the rare sheep breed kempisch heideschaap. **animal**, Elsevier, v. 2, n. 12, p. 1733–1741, 2008.

YOKOO, M. et al. Sumário genômico para características de difícil mensuração de animais da raça brangus-edição setembro de 2020. Bagé: Embrapa Pecuária Sul, 2021., 2021.

ZHANG, J. et al. Graph databases: Introduction and fundamentals. **Foundations and Trends® in Databases**, Now Publishers Inc, v. 12, n. 1, p. 1–133, 2018.



## APÊNDICE A – CÓDIGO-FONTE DOS PROGRAMAS DESENVOLVIDOS

Neste Apêndice encontram-se os códigos fonte que compõem o algoritmo desenvolvido.

```
base <- read.csv("ped_teste.csv", sep = ",", header =
  ↳ TRUE)

str(base)
dbase      <- base
dbase$ID   <- as.character (dbase$ID)
dbase$ID_Pai <- as.character (dbase$ID_Pai)
dbase$ID_Mae <- as.character (dbase$ID_Mae)
dbase$Safra <- as.integer   (dbase$Safra)
dbase$Sexo <- as.factor    (dbase$Sexo)
dbase$IECC <- as.vector    (dbase$IECC)
dbase$Selecionado <- as.integer (dbase$Selecionado)
str(dbase)
summary(dbase)
```

Como entrada do algoritmo é esperada uma base de dados onde deverá constar: *ID*: uma *string* contendo a identificação do animal; *ID\_Pai*: uma *string* contendo a identificação do pai, tendo como valor “0” caso não haja registro; *ID\_Mae*: uma *string* contendo a identificação da mãe, tendo como valor “0” caso não haja registro; *Safra*: um valor numérico contendo o ano de nascimento; *Sexo*: uma *string* contendo sexo do animal, sendo “MACHO” ou “FEMEA”; *IECC*: um valor numérico com o respectivo índice econômico do animal; *Selecionado*: um valor binário que serve como uma “flag” para indicar quais animais deverão acasalar.

```
males      <- subset(dbase, Sexo == "MACHO")      # Vetor
↳ onde serão armazenados somente os machos.
males      <- subset(males, Selecionado == "1")    # Aqui
↳ são separados somente os machos escolhidos para
↳ acasalar.
nmales     <- nrow(males)                          #
↳ Define o numero de machos escolhidos para acasalar.
```

```
females <- subset(dbase, Sexo == "FEMEA") # Vetor
↳ onde serão armazenados somente as femeas.
females <- subset(females, Selecionado == "1") # Aqui
↳ são separados somente as femeas escolhidas para
↳ acasalar.
nfemales <- nrow(females) #
↳ Define o numero de femeas escolhidas para acasalar.
```

A partir da base de dados, são criados dois *dataframes* distintos, contendo os machos e as fêmeas selecionadas para acasalar. Também são armazenadas em outras duas variáveis o número total de machos e de fêmeas que foram selecionados para acasalar.

```
Pe = 0.5 # Peso aplicado ao IECC.
Pc = 0.5 # Peso aplicado ao Coef. de parentesco.
lim_C = 0.05 # Limite de consanguinidade estipulado em
↳ 5%.
uso_min = 1 # Define o uso minimo de cada touro.
uso_max = 20 # Define o uso maximo de cada touro.
diretorio = "/fst"
```

São criadas variáveis com os pesos que imputam o grau de importância que será dado entre os fatores de índice econômico e coancestralidade. Além de variáveis para indicar as restrições que serão aplicadas quanto ao limite máximo de consanguinidade permitida, limite máximo de vezes que cada touro deve acasalar e limite mínimo de vezes que cada touro deve acasalar. Sendo as variáveis “uso\_min” e “uso\_max”, capazes de armazenar vetores indicando valores distintos de uso máximo e mínimo para cada touro, porém durante o desenvolvimento optou-se por utilizar os mesmos valores de limite para todos os touros. Também é criada a variável “diretório”, que será utilizada para definir o local de armazenamento de parte do processo de obtenção das informações de coancestralidade.

```
bigcrossprod <- function(m, nseg, path) { #Entrada: Matriz
↳ esparssa, n° de segmentos que se quer dividir o
↳ processo, e diretorio que será armazenado.
library(data.table)
```

```

library(Matrix)
library(fst)

jmax <- ncol(m)
# get the indices to split the columns of m into chunks
  → that will have
# approximately the same expense for crossprod
sumj <- cumsum(as.numeric(jmax:1))
dtj <- data.table(
  j = 1:jmax,
  int = sumj %/% (sumj[jmax]/nseg + 1)
)[
  , .(idx1 = min(j), idx2 = max(j)), int
][, idx1:idx2]
vapply(
  1:nseg,
  function(seg) {
    cat("\r", seg) # user feedback
    j1 <- dtj$idx1[seg]
    m2 <- as(triu(crossprod(m[, j1:dtj$idx2[seg], drop =
      → FALSE],
                                m[, j1:jmax])), "dgTMatrix")
    filepath <- file.path(path, paste0("dt", seg,
      → ".fst"))
    write.fst(
      data.table(
        i = attr(m2, "i") + j1,
        j = attr(m2, "j") + j1,
        v = attr(m2, "x")),
      filepath
    )
    filepath
  },
  character(1)

```

```
)
} #Saída: Arquivos ".fst" constando o resultado da
  ↳ multiplicação de matrizes.
```

A função *bigcrossprod* é responsável por realizar multiplicações entre matrizes. Sendo capaz de lidar com uma escala de dados maior do que a função *crossprod* pertencente ao R base. Para realização de suas operações faz uso das funções *cumsum*, *crossprod*, *file.path*, *on.exit* e *unlist* do R base. Também utiliza as funções *makeCluster*, *stopCluster* e *parLapply* do pacote *parallel*, para que seja possível implementar paralelismo ao processo de multiplicação. Além de utilizar as funções *triu* e *write.fst*, referentes aos pacotes *matrix* e *fst*, respectivamente. No contexto esta função possui como entrada uma matriz esparsa com os valores de parentesco entre cada animal, o número que se deseja segmentar o processo (o que também determina o número de multitarefas que serão executadas) e o diretório onde deverá ser armazenado os segmentos com os resultados da multiplicação entre as matrizes. Sua saída consiste em documentos do tipo “fst” que posteriormente deverão ser reagrupados para se obter a matriz com os coeficientes de parentesco de todos os animais.

```
# Função para criar e adicionar os possíveis novos
  ↳ produtos na base de dados.
f_new_product = function(db) { # Entrada: Base
  ↳ de dados
  tam      <- nmales*nfemales # Determina o
  ↳ numero de possíveis produtos.
  products = matrix(nrow = tam, ncol = 7) # Matriz onde
  ↳ serão armazenados os novos produtos.
  cont     = 0 # Contador usado
  ↳ para avançar as posições do vetor produtos.

  # Determina proximo ano para utilizar como estimativa de
  ↳ safra do possível produto.
  ano <- Sys.Date()
  ano <- format(ano, format = "%Y")
  ano <- as.numeric(ano) + 1
```

```

for (i in 1:nmales){
  for (j in 1:nfemales){
    cont = cont + 1
    products[cont,1] <- paste(males$ID[i],
      ↪ females$ID[j], sep = "") # Soma dos ID's dos
      ↪ pais para formar ID do produto.
    products[cont,2] <- males$ID[i]
      ↪ # ID do possivel pai.
    products[cont,3] <- females$ID[j]
      ↪ # ID da possivel mãe.
    products[cont,4] <- ano
      ↪ # Ano da possivel safra.
    products[cont,5] <- "P"
      ↪ # Sexo é definido com P.
    products[cont,6] <- (males$IECC[i] +
      ↪ females$IECC[j])/2 # Valor estimado de
      ↪ lucro.
    products[cont,7] <- 0
      ↪ # Valor de seleção que não será usado.
  }
}

df_p = data.frame(products)
names(df_p) = c("ID", "ID_Pai", "ID_Mae", "Safra",
  ↪ "Sexo", "IECC", "Selecionado")

db <- rbind(db,df_p)
  ↪ # Inclui os possiveis produtos na base de dados.
return (db)
  ↪ # Saída: Base de dados com o acréscimo dos possiveis
  ↪ produtos.
}

```

A função `f_new_product` recebe como argumento a base de dados após ter sido

ordenada de maneira crescente a partir da “safra”, para então retorná-la com o acréscimo de todos os possíveis produtos. A quantidade de possíveis produtos se dá pelo número de machos vezes o número de fêmeas, selecionados para acasalar. Para calcular o IECC de cada possível produto é feita uma média entre o IECC do pai e da mãe. O sexo dos possíveis produtos é definido como “P”.

```
# Função para criar novos ID's.
f_new_id = function(db) { # Entrada: Base de dados ordenada
  ↪ de acordo com a safra (animais mais velhos em
  ↪ primeiro).
  nid = 0 # Vetor onde serão armazenados os
  ↪ novos id's de cada animal.
  for (i in 1:nrow(db)) {
    nid[i] <- i # Cada novo id de cada animal vai
    ↪ sendo armazenado no vetor nid.
  }
  db <- cbind(db, nid) # Inclui o vetor nid como uma
  ↪ coluna na base de dados.
  return (db) # Saída: Base de dados com novos
  ↪ id's.
}
```

A função `f_new_id` é responsável por criar novas identificações numéricas para cada animal. Seu argumento esperado a base de dados após ter sido usada a função `f_new_product` e a saída é a base de dados com uma nova coluna de dados chamada “nid”, constando as novas identificações de cada animal.

```
# Função para encontrar os novos ID's de pai (sire) e mãe
  ↪ (dam) de cada animal.
f_new_id_sd = function(db) { # Entrada: Base de dados com
  ↪ novos id's de acordo com a safra.
  nids = 0 # Vetor onde serão armazenados
  ↪ os novos id's dos pais.
  nidd = 0 # Vetor onde serão armazenados
  ↪ os novos id's das mães.
```

```

# Cria uma tabela hash para otimizar o tempo de busca.
dict <- new.env(hash = TRUE, parent = emptyenv(), size =
  ↪ NA)

for (i in 1:nrow(db)) {
  dict[[db$ID[i]]] <- db$nid[i]
}

for (i in 1:nrow(db)){
  ifelse(is.null(dict[[db$ID_Pai[i]]]), nids [i] <- 0,
  ↪ nids [i] <- dict[[db$ID_Pai[i]])
  ifelse(is.null(dict[[db$ID_Mae[i]]]), nidd [i] <- 0,
  ↪ nidd [i] <- dict[[db$ID_Mae[i]])
}

db <- cbind(db,nids) # Inclui o vetor com o novo id dos
  ↪ pais (sire) como uma coluna na base de dados.
db <- cbind(db,nidd) # Inclui o vetor com o novo id das
  ↪ mães (dam) como uma coluna na base de dados.
return (db)          # Saída: Base de dados com novos
  ↪ id's de pais e mães.
}

```

A função *f\_new\_id\_sd*, tem o propósito de identificar o “nid” de cada pai e mãe de cada animal. A entrada esperada é a base de dados após uso da função *f\_new\_id* e sua saída é a base de dados com o acréscimo de duas novas colunas chamadas “nids” e “nidd”, referentes as novas identificações dos pais e das mães dos indivíduos, respectivamente.

```

# Função para calcular todos os coeficientes de parentesco
  ↪ e consanguinidade do rebanho.
f_get_product = function(db) { # Entrada: Base de dados e
  ↪ limite de consanguinidade.
  library(pedigreemm)
  library(data.table)

```

```

library(fst)
library(dplyr)

# Coluna 8 se refere ao novo id do animal, coluna 9 id
  ↳ do pai e coluna 10 id da mãe.
pedmm <- pedigree(sire = db[,9], dam = db[,10],
  ↳ label=db[,8])
U <- relfactor(pedmm)

# Cria a matriz de parentesco de todos com todos.
mp <- rbindlist(lapply(bigcrossprod(U, 50, diretorio),
  function(x) read.fst(x,
    ↳ as.data.table = TRUE)))

# Soma os valores de cada linha i
soma_p1 <- mp %>%
  group_by(i) %>%
  summarise(Total = sum(v))

# Soma os valores de cada linha j
soma_p2 <- mp %>%
  group_by(j) %>%
  summarise(Total = sum(v))

soma_p1 <- as.numeric(soma_p1$Total)
soma_p2 <- as.numeric(soma_p2$Total)

# Dataframe onde serão armazenados somente os possíveis
  ↳ produtos.
produtos <- subset(db, Sexo == "P")

# Obtem o total do somatório de parentesco de cada
  ↳ animal, descontando os valores 1's da
  ↳ consanguinidade.

```



```

Fc      <- inbreeding(pedmm)
CPar    <- ((soma_p1 + soma_p2) - 1) - Fc

Fc      <- Fc[produtos$nid[1]:length(Fc)]
CPar    <- CPar[produtos$nid[1]:length(CPar)]

produtos$Safra      <- NULL
produtos$Sexo       <- NULL
produtos$Selecioneado <- NULL
produtos$nid        <- NULL
produtos$nid1       <- NULL
produtos$nid2       <- NULL
produtos$IECC       <- as.numeric(produtos$IECC)

produtos <- cbind(produtos, CPar)
produtos <- cbind(produtos, Fc)

return(produtos)
} # Saida: Possíveis produtos com seus coeficientes de
↪ consanguinidade e seus somatórios de coeficientes de
↪ parentesco.

```

A função *f\_get\_product* cujo argumento é a base de dados após ter sido modificada pela função *f\_new\_id\_sd*. Dentro desta função usa-se as funções *pedigree* e *relfactor* do pacote *pedigreemm*. A função *pedigree* exige que os indivíduos sejam identificados por números, tendo em vista que geralmente os “ID’s” de cada animal se tratam de uma variável do tipo *string* se faz necessário o uso de identificações numéricas para cada animal, para isto foram desenvolvidas as funções *f\_new\_id* e *f\_new\_id\_sd*. A função *relfactor* é usada para obter o fator de relacionamento do “pedigree” que sozinho não determina a coancestralidade entre cada indivíduo. A partir do fator de relacionamento, realizando uma multiplicação de matrizes é possível obter a matriz de parentesco do rebanho. Para realizar essa multiplicação entre matrizes pode-se usar a função *crossprod* do R base, porém há limitações na quantidade de dados suportada. Para lidar com uma maior quantidade de dados, foi usada a função *bigcrossprod*, junto das funções *rbindlist* e

*read.fst*, pertencentes aos pacotes *data.table* e *fst*, respectivamente, além da função *lapply* do R base. A função *bigcrossprod* armazena seus resultados em arquivos do tipo “fst”, sendo necessário o uso das demais funções citadas para realizar a leitura dos resultados e a recomposição da matriz de parentesco. A partir da matriz de parentesco são utilizadas as funções *group\_by* e *summarise* do pacote *dplyr*, para se obter o valor do somatório de parentesco de cada animal. O coeficiente de consanguinidade de cada animal do rebanho é obtido através da função *inbreeding* do pacote *pedigreemm*. Então é criado um novo *dataframe* com somente os animais cujo sexo é “P”. Para então remover as colunas “Safrã”, “Sexo”, “Selecionado”, “nid”, “nids” e “nidd” e depois acrescentar as colunas com as informações de coancestralidade e consanguinidade destes animais. A saída da função *f\_get\_product* é um *dataframe* com somente os possíveis produtos.

```
# Função para calcular média ponderada
f_average = function(f1, f2, p1, p2){ # Entrada: Dois
  ↪ vetores f1 e f2, peso p1 relativo ao vetor f1 e peso
  ↪ p2 relativo ao vetor f2.
  med = ((f1 * p1) + (f2 * p2)) / (p1 + p2)
  return (med)
} # Saida: Média ponderada entre os dois valores,
  ↪ considerando que se quer uni-los em uma única
  ↪ expressão de minimização.
```

A função *f\_average* tem como propósito realizar a média ponderada entre dois vetores, possui como argumentos dois vetores e dois “pesos” (“p1” aplicado ao vetor “f1” e “p2” aplicado ao vetor “f2”). Os argumentos de peso permitem não somente determinar a importância de cada aspecto mas também permitem ignorar um deles e focar somente no outro. A saída da função *f\_average* é um vetor com a média ponderada entre os dois vetores de entrada. A função *f\_average* neste contexto é usada para se obter a média ponderada entre os valores de IECC e índice de coancestralidade.

```
# Função para aplicar penalidade de acordo com o limite de
  ↪ consanguinidade
f_penalize = function(Fc, med, lim_cons){ # Entrada:
  ↪ Valores de consanguinidade, médias ponderadas, limite
  ↪ de consanguinidade.
```

```

MAXINT = as.integer(.Machine$integer.max)           #
  ↪ Maior valor possível (usado como penalidade).
for(i in 1:length(Fc)){
  if(Fc[i] > lim_cons){
    med[i] <- MAXINT
  }
}

return(med)
} # Saida: Médias com a devida penalidade aplicada de
  ↪ acordo com o limite estabelecido de consanguinidade.

```

A função *f\_penalize* recebe como argumento dois vetores diferente e um limite máximo a ser aplicado. O primeiro vetor tem seus valores comparados com o limite determinado e caso ultrapasse este limite, o segundo vetor é penalizado ao receber o maior número inteiro possível de ser representado pelo R. Neste contexto é utilizado para penalizar a média ponderada de melhoramento de acordo com a consanguinidade.

```

#Função para criar a matriz de coeficientes das restrições
  ↪ das fêmeas utilizando o pacote Rcpp codificado em C++.
library(Rcpp)
cppFunction(
  "NumericMatrix f_coefficients_restrictions_females_c(int
  ↪ m, int f) {

  int nfemales = f;
  int nmales = m;
  NumericMatrix coef(nfemales, nfemales * nmales);

  for (int female = 0; female < f; female++){
    int k = 0;
    NumericVector v(nfemales * nmales);

    for (int i = 0; i < m; i++) {
      for (int j = 0; j < f; j++) {

```

```

        if (j == female){
            v[k] = 1;
        } else {
            v[k] = 0;
        }

        k = k+1;

    }
}

coef(female,_) = v;
}

return(coef);
}

"
)

```

```

#Função para criar a matriz de coeficientes das restrições
↪ dos machos utilizando o pacote Rcpp codificado em C++.
library(Rcpp)
cppFunction(
    "NumericMatrix f_coefficients_restrictions_males_c(int
    ↪ m, int f) {

        int nfemales = f;
        int nmales = m;
        NumericMatrix coef(nmales, nfemales * nmales);

        for (int male = 0; male < nmales; male++){
            int k = 0;

```

```

NumericVector v(nfemales * nmales);

for (int i = 0; i < nmales; i++) {
  for (int j = 0; j < nfemales; j++) {

    if (i == male){
      v[k] = 1;
    } else {
      v[k] = 0;
    }

    k = k+1;

  }
}

coef(male,_) = v;
}

return(coef);
}

"
)

```

As funções `f_coefficients_restrictions_males_c` e `f_coefficients_restrictions_females_c` são responsáveis por criar as matrizes de coeficientes de restrições dos machos e das fêmeas respectivamente. Estas funções são adaptações das funções homônimas desenvolvidas por Motta (2021).

```

# Função para mostrar a recomendação de acasalamentos.
f_mating <- function(db, sl){ # Entrada: Dataframe de
  ↪ produtos e solução.

  Macho      = 0

```

```

Femea      = 0
IECC       = 0
Fc         = 0
CPar       = 0
cont       = 0

for (i in 1:nrow(db)) {
  if(sl[i] == 1){
    cont <- cont + 1
    Macho   [cont] <- db$ID_Pai[i]
    Femea   [cont] <- db$ID_Mae[i]
    IECC    [cont] <- db$IECC[i]
    Fc      [cont] <- db$Fc[i]
    CPar    [cont] <- db$CPar[i]
  }
}

s <- data.frame(Macho, Femea, IECC, CPar, Fc)
colnames(s) <- c("Macho", "Fêmea", "IECC",
  ↪ "Coancestralidade", "Fc")
s$IECC          <- as.numeric(s$IECC)
s$Coancestralidade <- as.numeric(s$Coancestralidade)
s$Fc            <- as.numeric(s$Fc)
return(s)
} # Saída: Dataframe com as recomendações dos
↪ acasalamentos.

```

A função *f\_mating* recebe como argumento a base de dados dos possíveis produtos e a solução obtida, para retornar quais machos e fêmeas devem acasalar, além de informar uma estimativa de qual IECC, índice de coancestralidade e consanguinidade a possível prole terá.

```

dbase <- dbase[order(dbase$Safra, decreasing = FALSE,
  ↳ na.last = FALSE),] # Ordena a matriz de dados de
  ↳ acordo com a safra de cada animal para poder usarmos a
  ↳ biblioteca Pedigreemm.
dbase <- f_new_product(dbase)
  ↳ # Cria todos os possíveis produtos e adiciona a base
  ↳ total de animais.
dbase <- f_new_id      (dbase)
  ↳ # Cria uma nova identificação para os animais.
dbase <- f_new_id_sd  (dbase)
  ↳ # Encontra as novas identificações dos pais e mães que
  ↳ foram geradas no passo anterior.

```

Após serem criadas as funções necessárias, há uma ordem que deve ser seguida em sua execução para se obter o resultado desejado. Primeiro deve-se ordenar a base de dados de acordo com a “safra” de cada animal, de maneira que os mais velhos sejam referenciados primeiro. Logo após deve-se adicionar os possíveis produtos a base de dados através da função `f_new_product`. A partir da base completa com todos os animais o próximo passo é criar novas identificações únicas através da função `f_new_id` e encontrar os pais a partir dessas novas identificações com o uso da função `f_new_id_sd`.

```

produtos <- f_get_product(dbase)
  ↳ # Cria uma matriz somente com os possíveis produtos
  ↳ incluindo os dados de parentesco e consanguinidade de
  ↳ cada animal.

```

A partir da base de dados completa e atualizada com novas identificações, é usada a função `f_get_product` para criar um *dataframe* com somente os possíveis produtos, possuindo uma estimativa de IECC e índice de coancestralidade.

```

library(scales)
IECC_normal <- rescale(produtos$IECC, to = c(-1, 1))
  ↳ # Obtem os valores de IECC normalizados.
CPar_normal <- rescale(produtos$CPar, to = c(-1, 1))
  ↳ # Obtem os valores de Coancestralidade normalizados.

```

```

# Adiciona ao dataframe novas colunas com os valores de
↳ IECC e coeficiente de parentesco normalizados.
produtos <- cbind(produtos, IECC_normal)
produtos <- cbind(produtos, CPar_normal)

```

A função *rescales* da biblioteca *scales* é utilizada para obter os valores de IECC e índice de coancestralidade normalizados em uma faixa de valores entre  $-1$  e  $1$ .

```

Media <- f_average((produtos$IECC_normal * (-1)),
↳ produtos$CPar_normal, Pe, Pc) # Obtem a média
↳ ponderada de melhoramento.

# Adiciona ao dataframe nova coluna com a média ponderada
↳ entre os valores objetivo.
produtos <- cbind(produtos, Media)

# Aplica penalidade para todos os casos que excederem o
↳ limite de consanguinidade.
produtos$Media <- f_penalize (produtos$Fc, produtos$Media,
↳ lim_C)

# Limpa as variáveis que não serão mais usadas.
rm(IECC_normal, CPar_normal, Media)

```

A partir das variáveis de IECC e índice de coancestralidade é obtida uma média ponderada através da função *f\_average*. Neste caso leva-se em consideração que deseja-se minimizar os valores de coancestralidade e maximizar o IECC. Então os valores de IECC tem seu sinal invertido ao serem multiplicados por  $-1$ . Assim temos duas funções para serem minimizadas, pois ao se minimizar valores negativos, estamos maximizando seus valores absolutos.

```

# Cria um vetor com os coeficientes da função objetivo
↳ baseada na media ponderada entre Coancestralidade e
↳ IECC.
f.obj = produtos$Media

```



```

# Chamada da função para criar a matriz de coeficientes
↳ das restrições dos machos e das fêmeas.
crm <- f_coefficients_restrictions_males_c(nmales,
↳ nfemales)
crf <- f_coefficients_restrictions_females_c(nmales,
↳ nfemales)

# Cria a matriz dos coeficientes das restrições (machos +
↳ fêmeas).
# Cada linha da matriz corresponde aos coeficientes de uma
↳ única restrição.
f.con <- rbind(crm, crf)
f.con <- rbind(f.con, crm) # ((machos + fêmeas) + machos)
↳ pois os machos possuem a restrição >= além da <=.

# Atribui os sinais de igualdade / desigualdade
↳ correspondentes às restrições.
f.dir.males1 <- matrix("<=", nrow = 1, ncol = nmales)
↳ # Para os machos o sinal deve ser menor ou igual (cada
↳ macho pode acasalar <= número de acasalamentos
↳ definido pelo produtor).
f.dir.females <- matrix("=", nrow = 1, ncol = nfemales)
↳ # Para as fêmeas o sinal deve ser = (cada fêmea deve
↳ acasalar = 1).
f.dir.males2 <- matrix(">=", nrow = 1, ncol = nmales)
↳ # Para os machos o sinal deve ser maior ou igual (cada
↳ macho pode acasalar >= número de acasalamentos
↳ definido pelo produtor).

# Cria uma matriz única com os sinais (Junta as matrizes
↳ criadas acima).
f.dir <- cbind(f.dir.males1, f.dir.females)
f.dir <- cbind(f.dir, f.dir.males2)

```

```

f.rhs.males1  <- matrix(uso_max, nrow = 1, ncol = nmales)
  ↪ # Determina um numero maximo de vezes para todos os
  ↪ touros.
f.rhs.females <- matrix(1, nrow = 1, ncol = nfemales)
  ↪ # Cada vaca só pode acasalar uma única vez por
  ↪ solução.
f.rhs.males2  <- matrix(uso_min, nrow = 1, ncol = nmales)
  ↪ # Determina que cada touro seja usado pelo menos um
  ↪ tanto de vezes.

# Cria uma única matriz com os coeficientes do lado
  ↪ direito das restrições.
f.rhs <- cbind(f.rhs.males1, f.rhs.females)
f.rhs <- cbind(f.rhs, f.rhs.males2)

# Limpa as variaveis que não serão mais usadas.
rm(crm, crf, f.dir.males1, f.dir.females, f.dir.males2,
  ↪ f.rhs.males1, f.rhs.females, f.rhs.males2)
gc()

```

A partir do *dataframe* de possíveis produtos estar pronto. É definida a função objetivo que deve-se maximizar, neste caso é o vetor com as médias ponderadas que foi calculado anteriormente pela função *f\_average*. Logo após também é criada a matriz com os sinais de igualdade/desigualdade, além da matriz com os coeficientes do lado direito das restrições, da qual constam os números máximos e mínimos de uso para cada touro.

```

library(lpSolve)
lpSolve_r  <- lp("min", f.obj, f.con, f.dir, f.rhs,
  ↪ all.bin = TRUE)
sl         <- lpSolve_r$solution
produtos   <- cbind(produtos, sl)
  ↪ # Adiciona os resultados aos dados.
acasalamento <- f_mating(produtos, produtos$sl)
  ↪ # Armazena todos os acasalamentos recomendados.

```

Em posse de todas os argumentos requeridos para utilizar a função *lp* do pacote *lpSolve*, é possível obter a solução para função objetivo desejada.

```
# Armazena os resultados em arquivos ".csv".
write.csv(acasalamento, "resultado.csv", row.names =
  ↪ TRUE)
write.csv(summary(acasalamento), "resultado_sumario.csv",
  ↪ row.names = TRUE)
write.csv(produtos, "produtos.csv", row.names = TRUE)
```

Por fim a recomendação de acasalamento é armazenada em um arquivo “.csv”. Para este algoritmo, espera-se como entrada uma base de dados de animais de um rebanho, constando entre eles os machos e as fêmeas escolhidas para acasalar. É importante que contenha as informações sobre os pais de cada animal, o seu ano de nascimento e seu índice econômico. Também é esperado pelo algoritmo informações como número máximo e mínimo de vezes que cada macho pode acasalar. Além de ponderadores para os atributos que dizem respeito a coancestralidade e ao índice econômico. A partir deste conjunto de dados o algoritmo deve estimar o índice econômico e o índice de coancestralidade dos produtos que poderão ser gerados a partir dos machos e fêmeas selecionadas. Com estas informações, fazendo uso dos ponderadores configurados anteriormente, é estimada a contribuição ponderada de melhoramento relativa a cada possível produto. Então é realizado processo de programação linear que visa encontrar um esquema de acasalamento ótimo, retornando as indicações de quais pares devem acasalar, através de uma matriz binária, da qual o “0” serve para indicar que o acasalamento não deve ocorrer e “1” caso contrário.

## APÊNDICE B – REQUISIÇÕES DO BANCO DE DADOS DESENVOLVIDO

Neste Apêndice encontra-se as requisições utilizadas para desenvolver o banco de dados baseado em grafos.

```
LOAD CSV WITH HEADERS FROM 'file:///ped_teste.csv' AS line
CREATE (:Animal {ID: line.ID, Sexo: line.Sexo, IECC:
↪ line.IECC})
```

Inicialmente foram carregados os dados pertencentes a base de dados em arquivo .csv, criando assim os nós do tipo “Animal”.

```
LOAD CSV WITH HEADERS FROM 'file:///ped_teste.csv' AS line
MATCH (a:Animal {ID: line.ID})
MERGE (b:Animal {ID: line.ID_Mae})
ON CREATE SET b.ID = line.ID_Mae, b.Sexo = "FEMEA", b.IECC
↪ = "0"
ON MATCH SET a = a
```

Posteriormente foram registrados os animais que são referenciados na coluna “ID\_Mae” do arquivo, mas que não são referenciados na coluna “ID”.

```
LOAD CSV WITH HEADERS FROM 'file:///ped_teste.csv' AS line
MATCH (a:Animal {ID: line.ID})
MERGE (b:Animal {ID: line.ID_Pai})
ON CREATE SET b.ID = line.ID_Pai, b.Sexo = "MACHO", b.IECC
↪ = "0"
ON MATCH SET a = a
```

Também foram registrados os animais que são referenciados na coluna “ID\_Pai” do arquivo, mas que não são referenciados na coluna “ID”. Com isto o total de animais registrados passa a ser 4935.

```
LOAD CSV WITH HEADERS FROM 'file:///ped_teste.csv' AS line
MATCH (a:Animal {ID: line.ID})
```

```

MATCH (b:Animal {ID: line.ID_Mae})
WHERE b.ID <> "0"
CREATE (a) <- [t:Mae] - (b)

```

As relações de mãe e filho são criadas de maneira que as mães apontem para seus filhos.

```

LOAD CSV WITH HEADERS FROM 'file:///ped_teste.csv' AS line
MATCH (a:Animal {ID: line.ID})
MATCH (b:Animal {ID: line.ID_Pai})
WHERE b.ID <> "0"
CREATE (a) <- [t:Pai] - (b)

```

As relações de pai e filho também são criadas de maneira que os pais apontem para seus filhos.

```

LOAD CSV WITH HEADERS FROM 'file:///ped_teste.csv' AS line
MERGE (b:Safra {Ano: line.Safra})
ON CREATE SET b.Ano = line.Safra
ON MATCH SET b = b

```

A partir das safras registradas no banco de dados são criados os nós do tipo “Safra”.

```

CREATE (:Safra {Ano: "0"})

```

É criado um nó do tipo “Safra” cuja propriedade “Ano” é 0. Este nó deverá ser apontado por aqueles animais que não possuem safra especificada.

```

LOAD CSV WITH HEADERS FROM 'file:///ped_teste.csv' AS line
MATCH (a:Animal {ID: line.ID})
MATCH (b:Safra {Ano: line.Safra})
CREATE (a) - [t:NascidoEm] -> (b)

```

É criada a relação “NascidoEm” entre os nós que representam animais e os nós que representam sua safra.

```

MATCH (s:Safra {Ano: "0"})
MATCH (a:Animal)
WHERE NOT (a)-[:NascidoEm]->(:Safra)
CREATE (a)-[:NascidoEm]->(s)

```

Para os animais que não possuem safra é criada a relação “NascidoEm” com o nó do tipo “Safra” cujo ano é 0.

```

CREATE (:Geracao {Numero: "0"})
CREATE (:Geracao {Numero: "1"})
CREATE (:Geracao {Numero: "2"})
CREATE (:Geracao {Numero: "3"})
CREATE (:Geracao {Numero: "4"})
CREATE (:Geracao {Numero: "5"})

```

São criados 6 nós do tipo “Geracao” para determinar a qual geração pertence cada animal.

```

MATCH (a:Animal)
MATCH (b:Geracao {Numero: "0"})
WHERE NOT (a)<-[:Mae]-() AND NOT (a)<-[:Pai]-()
CREATE (a)-[:PertenceA]->(b)

```

Todos animais que não possuem indicação de pai e mãe, passam a apontar para o nó do tipo “Geracao”, cuja propriedade “Numero” é igual a 0, através da relação “PertenceA”.

```

MATCH (g:Geracao {Numero: "1"})
MATCH (p:Animal)-[:PertenceA]->(:Geracao {Numero: "0"})
MATCH (m:Animal)-[:PertenceA]->(:Geracao {Numero: "0"})
MATCH (m)-[:Mae]->(a:Animal)<-[:Pai]- (p)
CREATE (a)-[:PertenceA]->(g)

MATCH (g:Geracao {Numero: "1"})
MATCH (ac:Animal)-[:PertenceA]->(:Geracao {Numero: "0"})

```

```

MATCH (ac)-[:Mae]->(a:Animal)
WHERE NOT (a)<-[:Pai]-()
CREATE (a)-[:PertenceA]->(g)

MATCH (g:Geracao {Numero: "1"})
MATCH (ac:Animal)-[:PertenceA]->(:Geracao {Numero: "0"})
MATCH (ac)-[:Pai]->(a:Animal)
WHERE NOT (a)<-[:Mae]-()
CREATE (a)-[:PertenceA]->(g)

```

A partir dos filhos de pais pertencentes a geração numero 0 são criadas as relações que indicam os animais pertencentes a geração numero 1.

```

MATCH (g:Geracao {Numero: "2"})
MATCH (p:Animal)-[:PertenceA]->(:Geracao {Numero: "0"})
MATCH (m:Animal)-[:PertenceA]->(:Geracao {Numero: "1"})
MATCH (m)-[:Mae]->(a:Animal)<-[:Pai]-()
CREATE (a)-[:PertenceA]->(g)

MATCH (g:Geracao {Numero: "2"})
MATCH (p:Animal)-[:PertenceA]->(:Geracao {Numero: "1"})
MATCH (m:Animal)-[:PertenceA]->(:Geracao {Numero: "0"})
MATCH (m)-[:Mae]->(a:Animal)<-[:Pai]-()
CREATE (a)-[:PertenceA]->(g)

MATCH (g:Geracao {Numero: "2"})
MATCH (p:Animal)-[:PertenceA]->(:Geracao {Numero: "1"})
MATCH (m:Animal)-[:PertenceA]->(:Geracao {Numero: "1"})
MATCH (m)-[:Mae]->(a:Animal)<-[:Pai]-()
CREATE (a)-[:PertenceA]->(g)

MATCH (g:Geracao {Numero: "2"})
MATCH (ac:Animal)-[:PertenceA]->(:Geracao {Numero: "1"})
MATCH (ac)-[:Mae]->(a:Animal)
WHERE NOT (a)<-[:Pai]-()

```

```

CREATE (a)-[:PertenceA]->(g)

MATCH (g:Geracao {Numero: "2"})
MATCH (ac:Animal)-[:PertenceA]->(:Geracao {Numero: "1"})
MATCH (ac)-[:Pai]->(a:Animal)
WHERE NOT (a)<-[:Mae]-()
CREATE (a)-[:PertenceA]->(g)

```

A partir dos filhos de pais pertencentes a geração numero 1 são criadas as relações que indicam os animais pertencentes a geração numero 2.

```

MATCH (g:Geracao {Numero: "3"})
MATCH (p:Animal)-[:PertenceA]->(:Geracao {Numero: "0"})
MATCH (m:Animal)-[:PertenceA]->(:Geracao {Numero: "2"})
MATCH (m)-[:Mae]->(a:Animal)<-[:Pai]-()
CREATE (a)-[:PertenceA]->(g)

MATCH (g:Geracao {Numero: "3"})
MATCH (p:Animal)-[:PertenceA]->(:Geracao {Numero: "2"})
MATCH (m:Animal)-[:PertenceA]->(:Geracao {Numero: "0"})
MATCH (m)-[:Mae]->(a:Animal)<-[:Pai]-()
CREATE (a)-[:PertenceA]->(g)

MATCH (g:Geracao {Numero: "3"})
MATCH (p:Animal)-[:PertenceA]->(:Geracao {Numero: "1"})
MATCH (m:Animal)-[:PertenceA]->(:Geracao {Numero: "2"})
MATCH (m)-[:Mae]->(a:Animal)<-[:Pai]-()
CREATE (a)-[:PertenceA]->(g)

MATCH (g:Geracao {Numero: "3"})
MATCH (p:Animal)-[:PertenceA]->(:Geracao {Numero: "2"})
MATCH (m:Animal)-[:PertenceA]->(:Geracao {Numero: "1"})
MATCH (m)-[:Mae]->(a:Animal)<-[:Pai]-()
CREATE (a)-[:PertenceA]->(g)

```



```

MATCH (g:Geracao {Numero: "3"})
MATCH (p:Animal)-[:PertenceA]->(g:Geracao {Numero: "2"})
MATCH (m:Animal)-[:PertenceA]->(g:Geracao {Numero: "2"})
MATCH (m)-[:Mae]->(a:Animal) <-[:Pai]- (p)
CREATE (a)-[:PertenceA]->(g)

MATCH (g:Geracao {Numero: "3"})
MATCH (ac:Animal)-[:PertenceA]->(g:Geracao {Numero: "2"})
MATCH (ac)-[:Mae]->(a:Animal)
WHERE NOT (a) <-[:Pai]- ()
CREATE (a)-[:PertenceA]->(g)

MATCH (g:Geracao {Numero: "3"})
MATCH (ac:Animal)-[:PertenceA]->(g:Geracao {Numero: "2"})
MATCH (ac)-[:Pai]->(a:Animal)
WHERE NOT (a) <-[:Mae]- ()
CREATE (a)-[:PertenceA]->(g)

```

A partir dos filhos de pais pertencentes a geração numero 2 são criadas as relações que indicam os animais pertencentes a geração numero 3.

```

MATCH (g:Geracao {Numero: "4"})
MATCH (p:Animal)-[:PertenceA]->(g:Geracao {Numero: "0"})
MATCH (m:Animal)-[:PertenceA]->(g:Geracao {Numero: "3"})
MATCH (m)-[:Mae]->(a:Animal) <-[:Pai]- (p)
CREATE (a)-[:PertenceA]->(g)

MATCH (g:Geracao {Numero: "4"})
MATCH (p:Animal)-[:PertenceA]->(g:Geracao {Numero: "3"})
MATCH (m:Animal)-[:PertenceA]->(g:Geracao {Numero: "0"})
MATCH (m)-[:Mae]->(a:Animal) <-[:Pai]- (p)
CREATE (a)-[:PertenceA]->(g)

MATCH (g:Geracao {Numero: "4"})
MATCH (p:Animal)-[:PertenceA]->(g:Geracao {Numero: "1"})

```

```

MATCH (m:Animal)-[:PertenceA]->(:Geracao {Numero: "3"})
MATCH (m)-[:Mae]->(a:Animal)<-[:Pai]- (p)
CREATE (a)-[:PertenceA]->(g)

MATCH (g:Geracao {Numero: "4"})
MATCH (p:Animal)-[:PertenceA]->(:Geracao {Numero: "3"})
MATCH (m:Animal)-[:PertenceA]->(:Geracao {Numero: "1"})
MATCH (m)-[:Mae]->(a:Animal)<-[:Pai]- (p)
CREATE (a)-[:PertenceA]->(g)

MATCH (g:Geracao {Numero: "4"})
MATCH (p:Animal)-[:PertenceA]->(:Geracao {Numero: "2"})
MATCH (m:Animal)-[:PertenceA]->(:Geracao {Numero: "3"})
MATCH (m)-[:Mae]->(a:Animal)<-[:Pai]- (p)
CREATE (a)-[:PertenceA]->(g)

MATCH (g:Geracao {Numero: "4"})
MATCH (p:Animal)-[:PertenceA]->(:Geracao {Numero: "3"})
MATCH (m:Animal)-[:PertenceA]->(:Geracao {Numero: "2"})
MATCH (m)-[:Mae]->(a:Animal)<-[:Pai]- (p)
CREATE (a)-[:PertenceA]->(g)

MATCH (g:Geracao {Numero: "4"})
MATCH (p:Animal)-[:PertenceA]->(:Geracao {Numero: "3"})
MATCH (m:Animal)-[:PertenceA]->(:Geracao {Numero: "3"})
MATCH (m)-[:Mae]->(a:Animal)<-[:Pai]- (p)
CREATE (a)-[:PertenceA]->(g)

MATCH (g:Geracao {Numero: "4"})
MATCH (ac:Animal)-[:PertenceA]->(:Geracao {Numero: "3"})
MATCH (ac)-[:Mae]->(a:Animal)
WHERE NOT (a)<-[:Pai]- ()
CREATE (a)-[:PertenceA]->(g)

```

```

MATCH (g:Geracao {Numero: "4"})
MATCH (ac:Animal)-[:PertenceA]->(:Geracao {Numero: "3"})
MATCH (ac)-[:Pai]->(a:Animal)
WHERE NOT (a)<-[:Mae]-()
CREATE (a)-[:PertenceA]->(g)

```

A partir dos filhos de pais pertencentes a geração numero 3 são criadas as relações que indicam os animais pertencentes a geração numero 4.

```

MATCH (g:Geracao {Numero: "5"})
MATCH (p:Animal)-[:PertenceA]->(:Geracao {Numero: "0"})
MATCH (m:Animal)-[:PertenceA]->(:Geracao {Numero: "4"})
MATCH (m)-[:Mae]->(a:Animal)<-[:Pai]- (p)
CREATE (a)-[:PertenceA]->(g)

```

```

MATCH (g:Geracao {Numero: "5"})
MATCH (p:Animal)-[:PertenceA]->(:Geracao {Numero: "4"})
MATCH (m:Animal)-[:PertenceA]->(:Geracao {Numero: "0"})
MATCH (m)-[:Mae]->(a:Animal)<-[:Pai]- (p)
CREATE (a)-[:PertenceA]->(g)

```

```

MATCH (g:Geracao {Numero: "5"})
MATCH (p:Animal)-[:PertenceA]->(:Geracao {Numero: "1"})
MATCH (m:Animal)-[:PertenceA]->(:Geracao {Numero: "4"})
MATCH (m)-[:Mae]->(a:Animal)<-[:Pai]- (p)
CREATE (a)-[:PertenceA]->(g)

```

```

MATCH (g:Geracao {Numero: "5"})
MATCH (p:Animal)-[:PertenceA]->(:Geracao {Numero: "4"})
MATCH (m:Animal)-[:PertenceA]->(:Geracao {Numero: "1"})
MATCH (m)-[:Mae]->(a:Animal)<-[:Pai]- (p)
CREATE (a)-[:PertenceA]->(g)

```

```

MATCH (g:Geracao {Numero: "5"})
MATCH (p:Animal)-[:PertenceA]->(:Geracao {Numero: "2"})

```

```

MATCH (m:Animal)-[:PertenceA]->(:Geracao {Numero: "4"})
MATCH (m)-[:Mae]->(a:Animal)<-[:Pai]- (p)
CREATE (a)-[:PertenceA]->(g)

MATCH (g:Geracao {Numero: "5"})
MATCH (p:Animal)-[:PertenceA]->(:Geracao {Numero: "4"})
MATCH (m:Animal)-[:PertenceA]->(:Geracao {Numero: "2"})
MATCH (m)-[:Mae]->(a:Animal)<-[:Pai]- (p)
CREATE (a)-[:PertenceA]->(g)

MATCH (g:Geracao {Numero: "5"})
MATCH (p:Animal)-[:PertenceA]->(:Geracao {Numero: "3"})
MATCH (m:Animal)-[:PertenceA]->(:Geracao {Numero: "4"})
MATCH (m)-[:Mae]->(a:Animal)<-[:Pai]- (p)
CREATE (a)-[:PertenceA]->(g)

MATCH (g:Geracao {Numero: "5"})
MATCH (p:Animal)-[:PertenceA]->(:Geracao {Numero: "4"})
MATCH (m:Animal)-[:PertenceA]->(:Geracao {Numero: "3"})
MATCH (m)-[:Mae]->(a:Animal)<-[:Pai]- (p)
CREATE (a)-[:PertenceA]->(g)

MATCH (g:Geracao {Numero: "5"})
MATCH (p:Animal)-[:PertenceA]->(:Geracao {Numero: "4"})
MATCH (m:Animal)-[:PertenceA]->(:Geracao {Numero: "4"})
MATCH (m)-[:Mae]->(a:Animal)<-[:Pai]- (p)
CREATE (a)-[:PertenceA]->(g)

MATCH (g:Geracao {Numero: "5"})
MATCH (ac:Animal)-[:PertenceA]->(:Geracao {Numero: "4"})
MATCH (ac)-[:Mae]->(a:Animal)
WHERE NOT (a)<-[:Pai]- ()
CREATE (a)-[:PertenceA]->(g)

```

```

MATCH (g:Geracao {Numero: "5"})
MATCH (ac:Animal)-[:PertenceA]->(:Geracao {Numero: "4"})
MATCH (ac)-[:Pai]->(a:Animal)
WHERE NOT (a)<-[:Mae]-()
CREATE (a)-[:PertenceA]->(g)

```

A partir dos filhos de pais pertencentes a geração numero 4 são criadas as relações que indicam os animais pertencentes a geração numero 5.

```

MATCH p=(a:Animal)-[*]->(:Geracao {Numero: "0"})
WHERE a.ID = a.ID
CREATE (a)-[:TemParentesco {Valor: 1}]->(a)

```

Por fim está sendo criada a relação “TemParentesco” que tem como única propriedade a variável “Valor” utilizada para indicar o coeficiente de parentesco entre indivíduos. Atualmente esta relação existe somente para indicar o valor de parentesco entre os indivíduos da geração numero 0.

## APÊNDICE C – ALGORITMO DE TESTES

Neste Apêndice encontram-se os códigos fonte que compõem o algoritmo de teste desenvolvido. O principal ponto deste algoritmo são seus dois *scripts* utilizados para realizar os testes desejados. A maioria de suas funções já foram apresentadas anteriormente no Apêndice A.

```
results_m <- data.frame(Pe = numeric(0), Pc = numeric(0),
  ↪ nmales = numeric(0), mean_time = numeric(0))
Pe_values <- c(1, 0.8, 0.6, 0.4, 0.2, 0)
Pc_values <- c(0, 0.2, 0.4, 0.6, 0.8, 1)
nmales_values <- seq(from = 11, to = 30, by = 1)
```

Presume-se que todas as funções descritas e apresentadas no Anexo Apêndice A tenham sido carregadas antes de iniciar o uso dos *scripts*.

Os *scripts* possuem dependência de duas variáveis para o seu funcionamento: “results\_m” no caso do *script* que varia o número de machos e “results\_f” no caso do *script* que varia o número de fêmeas. Essas variáveis são do tipo *dataframe* e responsáveis por armazenar o peso aplicado ao índice econômico (Pe), o peso aplicado ao índice de coancestralidade (Pc), o número de machos ou fêmeas utilizados, e a média do tempo de execução. Além disso, há as variáveis “Pe\_values” e “Pc\_values”, que são vetores que armazenam diferentes configurações para “Pe” e “Pc”, respectivamente. Por fim, há a variável “nmales\_values” ou “nfemales\_values”, dependendo do *script*, que é um vetor com o número de animais para os quais se deseja realizar os testes.

```
for (x in 1:length(Pe_values)) {
  Pc = Pc_values[x]
  Pe = Pe_values[x]
  for (nmales in nmales_values) {
    times <- numeric(10)
    for (i in 1:10) {
      base <- read.csv("ped_teste.csv", sep = ",", header
        ↪ = TRUE)

      dbase <- base
```

```

dbase$ID          <- as.character (dbase$ID)
dbase$ID_Pai     <- as.character (dbase$ID_Pai)
dbase$ID_Mae     <- as.character (dbase$ID_Mae)
dbase$Saфра     <- as.integer   (dbase$Saфра)
dbase$Sexo       <- as.factor   (dbase$Sexo)
dbase$IECC       <- as.vector   (dbase$IECC)
dbase$Selecionado <- as.integer
  ↪ (dbase$Selecionado)

males <- subset (dbase, Sexo == "MACHO") #
  ↪ Vetor onde serão armazenados somente os machos.
males <- males[sample(nrow(males), size =
  ↪ nmales), ]
males <- males[order(males$IECC, decreasing =
  ↪ TRUE, na.last = FALSE),]

females <- subset (dbase, Sexo == "FEMEA") #
  ↪ Vetor onde serão armazenados somente as femeas.
females <- females[sample(nrow(females), size =
  ↪ 200), ]
females <- females[order(females$IECC, decreasing
  ↪ = TRUE, na.last = FALSE),]

nfemales <- nrow(females) #
  ↪ Define o numero de femeas escolhidas para
  ↪ acasalar.

MAXINT = as.integer(.Machine$integer.max) #
  ↪ Maior valor possivel (usado como penalidade).

lim_C = 0.05 # Limite de consanguinidade em 5%
uso_min = 1 # Define uso minimo de cada touro
uso_max = 20 # Define uso maximo de cada touro

```

```
diretorio = "C:/Users/dias_/Downloads/fst"
```

A primeira estrutura de repetição “for” é utilizada para percorrer todas as combinações de “Pe” e “Pc” que se deseje. O segundo laço de repetição utilizado é responsável por incrementar o numero de animais utilizados e definir o vetor “times” responsável por armazenar os diferentes tempos de execução de uma mesma configuração. O terceiro *loop* é responsável por medir os tempos de 10 execuções, para que em seguida, seja calculada a média dos tempos.

O código dentro dos laços de repetição é responsável por carregar a base de dados, definir as restrições e escolher os machos e fêmeas que deverão ser utilizados a partir da função “sample” que permite separar uma amostra aleatória dos dados.

```
dbase <- dbase[order(dbase$Safra, decreasing = FALSE,
  ↳ na.last = FALSE),] # Ordena a matriz de dados de
  ↳ acordo com a safra de cada animal para poder usarmos a
  ↳ biblioteca Pedigreemm.
  dbase <- f_new_product(dbase)
  ↳ # Cria todos os possiveis produtos e adiciona a
  ↳ base total de animais.
dbase <- f_new_id      (dbase)
  ↳ # Cria uma nova identificação para os animais.
dbase <- f_new_id_sd  (dbase)
  ↳ # Encontra as novas identificações dos pais e
  ↳ mães que foram geradas no passo anterior.

produtos <- f_get_product(dbase)
  ↳ # Cria uma matriz somente com os possiveis
  ↳ produtos incluindo os dados de parentesco e
  ↳ consanguinidade de cada animal.

library(scales)
IECC_normal <- rescale(produtos$IECC, to = c(0, 1))
CPar_normal <- rescale(produtos$CPar, to = c(0, 1))
```



```
# Adiciona ao dataframe novas colunas com os valores
  ↪ de IECC e coeficiente de parentesco
  ↪ normalizados.
produtos <- cbind(produtos, IECC_normal)
produtos <- cbind(produtos, CPar_normal)

Media <- f_average((produtos$IECC_normal * (-1)),
  ↪ produtos$CPar_normal, Pe, Pc)

# Adiciona ao dataframe nova coluna com a média
  ↪ ponderada entre os valores objetivo.
produtos <- cbind(produtos, Media)

# Aplica penalidade para todos os casos que
  ↪ excederem o limite de consanguinidade de 5%.
produtos$Media <- f_penalize (produtos$Fc,
  ↪ produtos$Media, lim_C)

rm(IECC_normal, CPar_normal, Media)

# Cria um vetor com os coeficientes da função
  ↪ objetivo baseada na media ponderada entre o
  ↪ somatorio de coancestralidade e IECC.
f.obj = produtos$Media

# Chamada da função para criar a matriz de
  ↪ coeficientes das restrições dos machos e das
  ↪ fêmeas.
crm <- f_coefficients_restrictions_males_c(nmales,
  ↪ nfemales)
crf <- f_coefficients_restrictions_females_c(nmales,
  ↪ nfemales)
```

```

# Cria a matriz dos coeficientes das restrições
→ (machos + fêmeas).
# Cada linha da matriz corresponde aos coeficientes
→ de uma única restrição.
f.con <- rbind(crm, crf)
f.con <- rbind(f.con, crm) # ((machos + fêmeas) +
→ machos) pois os machos possuem a restrição >=
→ alem da <=.

# Atribui os sinais de igualdade / desigualdade
→ correspondentes às restrições.
f.dir.males1 <- matrix("<=", nrow = 1, ncol =
→ nmales) # Para os machos o sinal deve ser
→ menor ou igual (cada macho pode acasalar <=
→ número de acasalamentos definido pelo produtor).
f.dir.females <- matrix("=", nrow = 1, ncol =
→ nfemales) # Para as fêmeas o sinal deve ser =
→ (cada fêmea deve acasalar = 1).
f.dir.males2 <- matrix(">=", nrow = 1, ncol =
→ nmales) # Para os machos o sinal deve ser
→ maior ou igual (cada macho pode acasalar >=
→ número de acasalamentos definido pelo produtor).

# Cria uma matriz única com os sinais (Junta as
→ matrizes criadas acima).
f.dir <- cbind(f.dir.males1, f.dir.females)
f.dir <- cbind(f.dir, f.dir.males2)

f.rhs.males1 <- matrix(uso_max, nrow = 1, ncol =
→ nmales) # Determina um numero maximo de
→ vezes para todos os touros.
f.rhs.females <- matrix(1, nrow = 1, ncol =
→ nfemales) # Cada vaca só pode acasalar
→ uma única vez por solução.

```

```

f.rhs.males2 <- matrix(uso_min, nrow = 1, ncol =
  ↪ nmales)      # Determina que cada touro seja
  ↪ usado pelo menos um tanto de vezes.

# Cria uma única matriz com os coeficientes do lado
  ↪ direito das restrições.
f.rhs <- cbind(f.rhs.males1, f.rhs.females)
f.rhs <- cbind(f.rhs, f.rhs.males2)

# Limpa as variáveis que não serão mais usadas.
rm(crm, crf, f.dir.males1, f.dir.females,
  ↪ f.dir.males2, f.rhs.males1, f.rhs.females,
  ↪ f.rhs.males2)
gc()

```

Dentro dos laços é realizado todo o processo necessário para obtenção do coeficiente da função objetivo para execução da função “lp”.

```

library(lpSolve)

start_time <- Sys.time()
lpSolve_r <- lp("min", f.obj, f.con, f.dir, f.rhs,
  ↪ all.bin = TRUE)
end_time <- Sys.time()

times[i] <- end_time - start_time

}
mean_time <- mean(times)
results_m <- rbind(results_m, data.frame(Pe = Pe, Pc =
  ↪ Pc, nmales = nmales, mean_time = mean_time))
write.csv(results_m, "AnaliseMachos.csv", row.names =
  ↪ TRUE)
}
}

```

A medição do tempo de execução é feita subtraindo o tempo inicial de execução, obtido através da função “Sys.time”, armazenado na variável ”start\_time”, do tempo final de execução, armazenado na variável “end\_time”. Esta medição é armazenada no vetor “times” que foi criado anteriormente.

Após a conclusão da terceira iteração do *loop*, a média dos tempos de execução é calculada com a função “mean”. Em seguida, as informações desejadas são armazenadas no *dataframe* “results\_m” (ou “results\_f”, dependendo da versão) e um arquivo .csv é gerado com essas informações.

A seguir, estão algumas partes do *script* específico para os testes com variação no número de fêmeas. As seções selecionadas correspondem às que diferem do *script* que testa a variação no número de machos.

```
nfemales_values <- seq(from = 100, to = 2000, by = 100)
```

```
for (nfemales in nfemales_values) {
```

```
males      <- subset(dbase, Sexo == "MACHO")           # Vetor
  ↳ onde serão armazenados somente os machos.
  males     <- males[sample(nrow(males), size = 10), ]
  males     <- males[order(males$IECC, decreasing =
  ↳ TRUE, na.last = FALSE), ]
  nmales    <- nrow(males)                             #
  ↳ Define o numero de machos escolhidos para
  ↳ acasalar.

  females   <- subset(dbase, Sexo == "FEMEA")         #
  ↳ Vetor onde serão armazenados somente as femeas.
  females   <- females[sample(nrow(females), size =
  ↳ nfemales), ]
  females   <- females[order(females$IECC, decreasing
  ↳ = TRUE, na.last = FALSE), ]
```

```
mean_time <- mean(times)
  results_f <- rbind(results_f, data.frame(Pe = Pe, Pc =
  ↳ Pc, nfemales = nfemales, mean_time = mean_time))
```

```
write.csv(results_f, "AnalyseFemeas.csv", row.names =  
  → TRUE)  
}
```