Optimal algorithms for mating systems using selection indexes*

Ana Paula Lüdtke Ferreira[1][0000−0001−7057−9095] and Renato Sayyed de Souza[1]

Programa de Pós-graduação em Computação Aplicada
Universidade Federal do Pampa, Bagé/RS, Brazil
anaferreira,renatosayyed.aluno@unipampa.edu.br

Abstract. Animal breeding involves directing the next generation's genetics and phenotypic traits towards animal health and financial profits for the production system, using essentially two strategies: the selection and the mating systems. The mating system aims to provide a set of sire-dam pairs, producing the best animals according to some selection criteria and a mating strategy for the animals chosen for reproduction. This paper presents an R-package to compute the best pairing scheme to obtain optimal polynomial-time solutions for either mating between peers or compensatory mating. The algorithm for optimal mating runs in worst-case $O(nm)$-time while the compensatory mating takes $O(m^3)$, where n is the number of sires and m is the number of dams.

Keywords: Animal breeding · Mating systems · Assignment problem.

1 Introduction

The growth of the world population, estimated at 9.8 billion by 2050 (https://population.un.org/wpp/), will determine a greater demand for food, implying new challenges for livestock production systems, in the sense of producing more with fewer resources in a scenario where water availability and land will be increasingly scarce. One way for producers to meet the demand for meat and maintain competitiveness in the international market is to improve the herds' genetic potential to produce better results [12].

Animal breeding involves directing the next generation's genetics and phenotypic traits towards animal health and financial profits for the production system, using essentially two strategies: the selection and the mating systems [11][10]. The selection process chooses the best animals to reproduce, increasing the frequency of desirable characteristics and decreasing the undesirable ones [18][6]. The selection process often relies on a selection index — a set of measurable traits weighted by their relative importance, allowing the expression of an animal value by a single number [20][1][23]. The mating system aims to provide the best possible combination of sire-dam pairs, producing the best animals according to the index selection and the defined mating strategy [21].

* This work was partially funded by FAPERGS, grant number 21/2551-0000958-6.

The mating system configured so that the best males reproduce with the best females and, consequently, the worst males reproduce with the worst females is called mating between peers. That strategy obtains exceptional animals regarding the selection index. On the other hand, a compensatory or corrective mating strategy seeks to bring more homogeneity within the population by pairing individuals differing in performance. Mating between peers increases the endogamy within the herd, while the compensatory mating strategy produces more genetically diverse individuals [3].

Algorithms for finding the best mating strategy often rely upon some metaheuristic approach [24], particularly genetic algorithms [4][16][2][5][15]. We have recently proven that optimal exact polynomial-time algorithms can solve both the mating between peers and the compensatory mating problems. The first problem by an optimal greedy strategy [13] and the latter by a polynomial-time reduction to the minimum assignment problem in weighted bipartite graphs [14]. The compensatory mating problem solution relies on the output of the optimal mating algorithm, assuring that the compensatory strategy is also optimal in the sense of having the maximum possible solution value.

This paper presents the algorithms derived from those two previous theoretical results. Specifically, we have built an R-package to compute the best pairing scheme to obtain optimal mating. The package builds optimal pairing schemes for both mating between peers and compensatory mating. The algorithms assume the existence of a selection index, but the actual calculation is unimportant. The algorithms receive as input: (i) a spreadsheet containing all animals selected for breeding and their respective trait values, (ii) a spreadsheet containing the values of all animal pairs' coancestry, (iii) a selection index consisting of a list of the considered traits and their respective weights, and (iv) the maximum number of times each male is allowed to mate. The algorithm returns the list of female-male pairings that maximizes the next generation's expected value accordingly to the established mating goal. The algorithm for optimal mating runs in $O(nm)$-time while the compensatory mating takes $O(mn + n^2 logn)$ worst-case running time, where n is the number of sires and m is the number of dams.

The rest of the paper is organized as follows: Section 2 formalizes the concepts of selection index, mating pair contribution, and the optimal compensatory mating assignment problem. Section 3 presents the main package features, as well as the set of tools we have used to develop the R package. Section 4 presents and discusses the results, and Section 5 concludes the paper.

2 Problem formalization

A *selection process* intends to elect the animals to become the next generation parents [26][23][25][10]. The selection process analyses the herd's genetic/economic values and phenotypes of genetic/economic interest. The value of each animal is computed from a selection index, chosen accordingly to the elicited breeding objectives. A *selection index* expresses the relative reward/punishment of each trait to consider in the breeding ahead. Formally, a *selection index* is a pair $\mathcal{I} = (T, w)$ where T is a set of animal traits and $w : T \to \mathbb{R}$ is the index weighting function.

Given a selection index $\mathcal{I}$, the *individual value* of each animal $a \in A$ is the sum of its individual characteristics' values weighted by the relative importance of each characteristic accordingly to $\mathcal{I}$. Formally, an individual value can be computed by the summation given in (1), where $\nu : T \times A \to \mathbb{R}$ outputs $a \in A$'s value for each trait $t \in T$. When the selection index is understood from the context, we write $\iota(a)$ instead of $\iota^{\mathcal{I}}(a)$.

$$\iota^{\mathcal{I}}(a) = \sum_{k \in T} \nu(k, a) \cdot w(k) \tag{1}$$

The mating contribution is the average of both parent's individual values, except when they have a co-ancestry above an acceptable level. The breed of related animals shrinks genetic diversity and leads to undesirable health issues. Inbreeding strategies are helpful to fix a desirable phenotype within a population, but it is usually undesirable in the mating process. Animal breeding programs control the degree of kinship amongst animals, making them available to producers and researchers. As in the case of traits, databases or spreadsheets maintain animal kinship information. We assume that the producer can choose whether inbreeding is acceptable and, if so, to what degree. Definition 1 inserts that idea into the calculation of mating pairs expected offspring contribution value.

Definition 1 (Mating contribution). *Let $A = S \cup D$ be a set of animals, where S is the set of sires and D is the set of dams, with $S \cap D = \emptyset$, $\mathcal{I}$ be a selection index, $r : A \times A \to \mathbb{R}^+$ be the degree of which two animals are related, and l be the maximum inbreeding threshold. The* mating contribution *$\pi : S \times D \to \mathbb{R}$ of a sire $s \in S$ and a dam $d \in D$ is computed as follows:*

$$\pi(s, d) = \begin{cases} (\iota^{\mathcal{I}}(s) + \iota^{\mathcal{I}}(d))/2 & r(s, d) \leq l \\ -\infty & r(s, d) > l \end{cases} \tag{2}$$

where $\iota^{\mathcal{I}}$ is the individual contribution of each animal, given selection index $\mathcal{I}$.

The mating contribution gives the expected offspring trait values of each dam/sire pair, which is the average of both parents' contributions unless the couple's kinship is above the acceptable inbreeding threshold.

The above definitions allow us to define the problems we intend to tackle.

Definition 2 (Optimal mating assignment problem).
Input*: A tuple $(S, D, \mathcal{I}, \pi, \max)$ where S is a set of sires, D is a set of dams, $\mathcal{I}$ is a selection index, $\pi : S \times D \to \mathbb{R}$ is a contribution function computed as in Definition 1, and $\max : S \to \mathbb{N}$ is a function expressing the use limit of each sire in the mating process.*
Output*: A function $b^* : D \to S$ obeying the use limit for each sire, i.e. for each $s \in S$ we have that $|\{d \mid b^*(d) = s\}| \leq \max(s)$, and such that the sum*

$\sum_{d \in D} \pi(b^*(d), d)$ *is maximal, i.e., for each other function* $b : D \to S$ *obeying the sires' use limit we have that*

$$\sum_{d \in D} \pi(b(d), d) \leq \sum_{d \in D} \pi(b^*(d), d) \tag{3}$$

Definition 3 (Optimal compensatory mating problem).
Input*: A tuple* (S, D, π, b^*) *where* S *is a set of sires,* D *is a set of dams,* π *is the contribution function (Eq. (2)), and* $b^* : D \to S$ *is an optimal mating function given as the solution of the optimal mating assignment problem (Definition 2).*
Output*: A function* $b^c : D \to S$ *with the same total value as* b^**, i.e.,* $\sum_{k \in D} \pi(b^c(k), k) = \sum_{k \in D} \pi(b^*(k), k)$*, and such that the solution variance is minimum. I.e, for any other function* $b' : D \to S$ *with* $\sum_{k \in D} \pi(b'(k), k) = \sum_{k \in D} \pi(b^*(k), k)$ *we have that*

$$\frac{1}{|D|} \sum_{d \in D} (\pi(b'(d), d) - \mu_{b'})^2 \leq \frac{1}{|D|} \sum_{d \in D} (\pi(b^c(d), d) - \mu_{b^c})^2 \tag{4}$$

[13] presents and discusses several theoretical results on the problem of optimal mating, including necessary and sufficient conditions for a result to exist. A greedy heuristic for the problem is proven to be correct and optimal, provided a solution to the problem exists. An algorithm resulting from that heuristic can perform in $O(|M|.|F|)$ polynomial-time and takes advantage from the contribution of animals to the future offspring being an arithmetic mean. I.e., given two mating pairs $(s_1, d_1), (s_2, d_2) \in S \times D$, we have that $\pi(s_1, d_1) + \pi(s_2, d_2) = \pi(s_2, d_1) + \pi(s_1, d_2)$, as long as the maximal accepted inbreeding threshold is not exceeded between pairs (s_2, d_1) and (s_1, d_2).

[14] analyses and discusses theoretical results concerning the optimal compensatory mating problem. Particularly, we have shown that the result from the optimal mating problem can be the basis for a solution for the compensatory mating problem while retaining the solution's optimality. The article proves that any solution involving the same males has a constant average value, regardless of the females involved in reproduction. Thus, we can compute the minimum variance of possible reproduction schemes without having to recalculate the average of each of them. This result allows us to model the problem as an instance of the assignment problem in weighted bipartite graphs, which has a polynomial algorithmic solution. The corresponding reduction is proven correct in the text of the article.

A reduction [7] from problem A to problem B defines an alternative algorithm for solving problem A using the solution for problem B. Therefore, we have a polynomial-time algorithm to solve the problem of compensatory mating, using an algorithm for the assignment problem in weighted bipartite graphs. The R package presented in this paper contains an implementation for the theoretical result previously presented.

3 The OMA package

The OMA (for *optimal mating assignment*) package implements algorithmic solutions for the problems listed in Definitions 2 and 3. The user can invoke two functions for calculating the mating system between equals and the compensatory mating system. The functions are, respectively, OptimalMatingSelection and CompensatoryMating.

The functions receive each selected animal's value, respecting the selection index and the matrix of co-ancestry between any two animals. Since that information has an external source, the functions "build_mating_values" and "b" build the necessary data structures from CSV files.

The user interface package functions are the following:

LoadCoancestryValues

Parameters:
CSVFile Animals' ancestrality data file name. The file must have the CSV format (*comma-separated values*) with three columns: identification of the animal, identification of the mother, and identification of the father.
CSVsep File separation character (colon or semicolon).
CSVHeader Indication whether the file has a header (true or false).

The LoadCoancestryValues function receives the file containing the animals' co-ancestrality data as input. It returns a two-dimensional data frame containing the calculated co-ancestrality values for each pair of animals. The calculation process uses the R package 'pedigree' [8]. If the user wants another calculation, he can build the co-ancestry matrix in the same format returned by this function and use it further as needed.

LoadTraitValues

Parameters:
FileCSV Animals' trait values file name. The file must have the CSV format (*comma-separated values*) with as many columns as needed, being the first the identification of the animal. The other columns should contain the trait values identified in the file header. Those names will be used to identify the traits belonging to the selection index.
CSVsep File separation character (colon or semicolon).

The file with the values of each measured trait of the animals must have a header identifying the traits' names. The selection index can contain any subset of existing traits with the desired weighting. If the trait name in the selection index does not exist in the file, the function issues a warning to the user, ignoring it in the selection index. The function returns a data frame containing the file contents.

LoadSelectedAnimals

Parameters:

FileCSV List of animals selected for reproduction. The file must have the CSV format (*comma-separated values*) containing the identification of each animal, its sex, and the maximum number of times it should be used for reproduction.

CSVsep File separation character (colon or semicolon).

CSVHead Indication whether the file has a header (true or false).

sexID Array of two positions indicating how male and females are designated in the file's second column.

The LoadSelectedAnimals identifies the actual animals chosen for reproduction. Notice that the user can maintain genetic information and perform trait measurements for all their animals. However, not all animals typically participate in the breeding process because they are too old, too young, or do not have the necessary qualities. The last parameter, **sexID**, makes easier to use the function without having to alter the original files. The function returns a list containing each animal's identification, sex, and reproductive limit, in the form of an R data structure.

he reproductive limit must exist to assure that physical limits are not extrapolated and maintain the herd inbreeding limits at check.

BuildContributionMatrix

Parameters:

SelIndex The selection index is a data frame containing the trait name and relative weight (a real number). The trait name must belong to the list of trait values loaded by the function LoadTraitValues.

SelectedAnimals List of animals selected for reproduction, using the same identification from the loaded files, identified as male of female, with their reproductive limit.

TraitValues Data frame returned by the function LoadTraitValues or a similar structure provided in another way by the user.

AncestryMatrix The ancestry matrix is the value returned from the function LoadCoancestryValues or a similar structure provided in another way by the user.

InbreedingThreshold Threshold value for related individuals.

The BuildContributionMatrix function organizes the data correctly so that the greedy heuristics can assure the optimal mating solution. It also builds the data structure to recuperate the original animals' identification to present a solution that makes sense to the end-user.

The role of this function is to organize the data so that the algorithms perform efficiently. First, the algorithm performs the computation of individual values, for all animals, according to the selection index (Eq. (1)). The animals are then separated into males and females and ordered by the computed selection index value, from the highest to the lowest.

The contribution matrix organizes the males in lines and the females in columns, according to their order. The contribution matrix is filled with the expected contribution value of each pair from the data passed in the last three parameters (Definition 1), ensuring that related animals above the kinship limit will not belong to the solution. This array, with the contribution values of each pair, is the data returned by the algorithm.

OptimalMatingSelection

Parameters:

ContributionMatrix A $n \times m$ matrix, where n is the number of sires and m is the number of dams, returned from function BuildContributionMatrix, or built in a way that assures the sires and dams are ordered by their individual values.

maxUsage A n-element vector containing each sire maximum use, in the same order the sires appears in the first parameter.

The OptimalMatingSelection function implements the greedy heuristic presented in [13]. The algorithm traverses the contribution matrix, assigning each dam to a sire, starting from the dam with the highest value (position 1) in the direction of the one with the lower value. The sire chosen is always the one with the highest selection value that has not yet reached its limit and is unrelated to the dam.

If the algorithm fails in founding a sire for a dam, it searches for the second-best possible choice by scanning all dams already allocated to a sire. If the best-valued allocated sire matches the left-out dam, the algorithm changes the allocations. If there is a solution to the problem (i.e., if it is possible to pair any female to some male, respecting their maximum use), the algorithm returns the best possible solution.

The algorithm OptimalMatingSelection returns a m-sized array, with the sire index chosen for each dam.

CompensatoryMating

Parameters:

ContributionMatrix A $n \times m$ matrix, where n is the number of sires and m is the number of dams, returned from function BuildContributionMatrix, or built in a way that assures the sires and dams are ordered by their individual values.

OptimalAssignment The solution returned by the algorithm OptimalMatingSelection.

The CompensatoryMating algorithm begins by counting the number of times each sire is used for reproduction, according to the its second parameter. Then it builds a bipartite graph [9], with the set of dams as its left side. The right side contains the same number of nodes as the left side, with sires appearing the same number of times they appear in the optimal solution. The bipartite graph is complete, i.e., all left nodes are connected to each of the right nodes. The graph edge is labelled with the value $(\pi_{ij} - \mu)^2$, where π_{ij} is the pair (j, i) mating value, and μ is the solution average computed from the contribution matrix.

The assignment problem in weighted bipartite graphs consists on finding the surjective set of arcs with domain in the left side and codomain in the right side whose sum is minimum. [14] shows that is equivalent to minimize the mating solution's variance.

The problem is solved with R package RcppHungarian [22], which applies the the Kuhn-Munkres algorithm (also known as the Hungarian method) [17,19]. The algorithm complexity $O(mn + n^2 \log n)$, with n being the number of dams $|D|$ and $m = |D|^2$. Therefore, the complete procedure is cubic in the number of dams.

The algorithm CompensatoryMating returns a $|D|$-sized array, with the sire index chosen for each dam.

4 Results and discussion

A database containing 1366 animals (Brangus cattle) with 511 males and 855 females served to test the algorithms. The EMBRAPA Southern Livestock provided all data.

We have conducted the tests within the R Studio IDE (https://www.rstudio.com/), using version R interpreter version 4.1.3 for Windows. The hardware used is a AMD Ryzen 7 2700X 8 cores 16 threads 3.7 GHz, 16 GB DDR4 2666 MHz RAM, Microsoft Windows 11 x64 operating system. Python's libraries Seaborn v. 0.11.2 and Matplotlib v. 3.5.1 produced the graphics presented below.

The algorithms' average execution time calculation runs them with the same input five times in a row. Although the algorithms are deterministic, we did that to ensure machine idiosyncrasies did not take part in the produced values.

The first experiment fixed the number of males and varied the number of females, adding 30 females to each run. The second experiment did the opposite, fixing the number of females while varying the number of males, incrementing 20 to each execution. Fig. 1 shows the results for the OptimalMatingSelection. The image on the left shows the execution time changing as the number of females increases, while the one on the right shows the execution time changing as the number of males increases.

The portrayed function on the left suggests a linear growth in the execution time if we add more males to a fixed number of females. A linear function is consistent with the algorithm operation: sires with lower values will possibly never be chosen if added. Correspondingly, adding quality sires makes the algorithm find them first, passing immediately to the next female. In both cases, the execution time should not change much.

The graphic on the right shows a quadratic function, as expected from the theoretical results. Adding more females makes the contribution matrix bigger, and the recalculations from the kinship restrictions are dependent on the number of females. Notice that, unlike the case with more males, adding more females makes the last need to search for males with lower contributions values. Hence, the algorithm traverses more lines than in the settings with fewer dams. The maximum reproduction limit plays a significant role in the execution time: the bigger the value, the fewer matrix lines need to be considered.

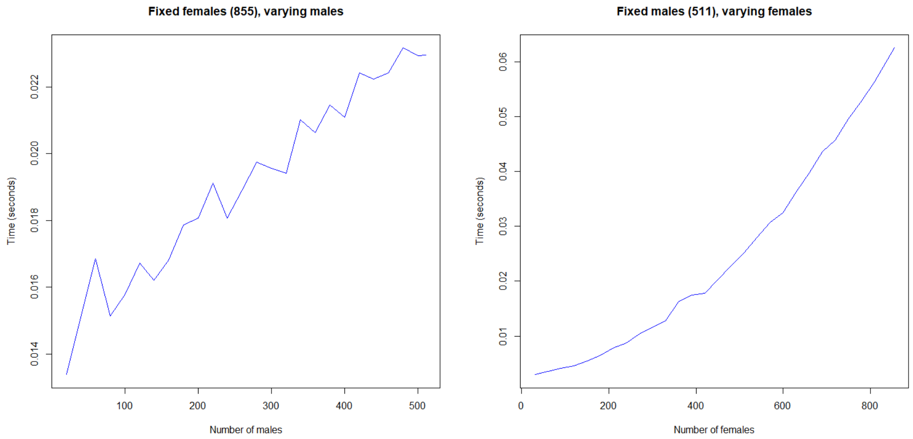

Fig. 1. OptimalMatingSelection average execution time

Fig. 2 portrays the execution time values for the CompensatoryMating algorithm, using the same method described above. The image on the left shows how the execution time varies as the number of males increases, while the one on the right shows the variation with adding more females.

As predicted by the algorithm complexity analysis, the execution time varies only with the number of females. The linear variation in the number of males reflects the time needed to execute the optimal mating algorithm, whose solution serves as input to the compensatory mating program. However, as the number of females increases, the execution time shows the expected cubic $O(n^3)$ form.

Although polynomial, a cubic-time algorithm runs slowly compared to linear or quadratic ones. An $O(n^2)$-algorithm can compute ten thousand elements in a matter of seconds, while a $O(n^3)$ one would take several minutes to do the same. If the input size goes to hundreds of thousands, the quadratic algorithm would still take seconds to run, albeit the cubic would need several days. However, the algorithms we have presented here are exact, different from meta-heuristic approaches that do not assure the optimal result. Given the economic impact of an optimal algorithm, processing of huge herds' data could still occur in time for a proper, optimal mating strategy.

5 Conclusion

Livestock production systems form the basis of the economic activity in many South American regions, including Argentina, Brazil, and Uruguay. Meat quality,

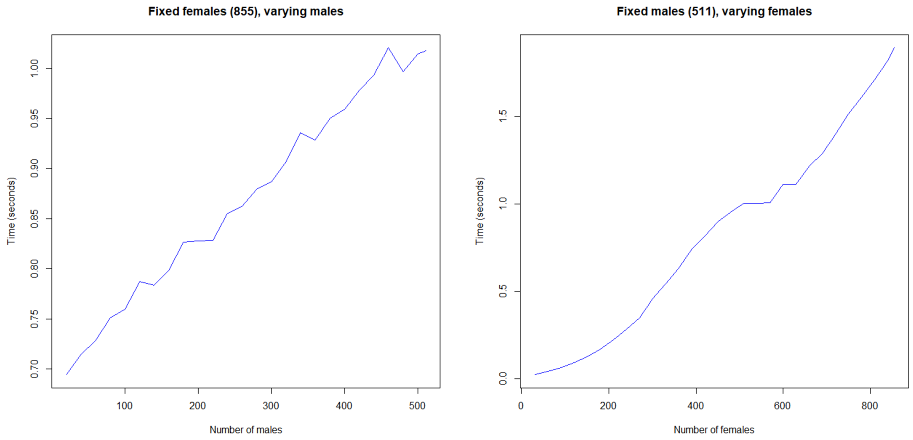

Fig. 2. CompensatoryMating average execution time

animal health, and financial profits depend on genetic and phenotypic characteristics, maintained and evolved throughout generations by employing animal breeding systems. The selection and the mating systems are at the core of animal breeding.

We have presented an R package with exact algorithms for optimal and compensatory mating, given a set of animals chosen for reproduction. The algorithms rely on a selection index, which can be customized to meet the breeding goals. A limitation on the co-ancestry level between mating pairs meets inbreeding restrictions requirements.

The algorithm for optimal mating tends to provide a solution reflecting mating between peers. A compensatory strategy with the same selection value as the former appears on the algorithm for compensatory mating. Both algorithms have a polynomial worst-time execution, ensuring no meta-heuristic approach is necessary.

The compensatory mating algorithm uses the Kuhn-Munkres algorithm. We are currently investigating the existence of a more efficient algorithm to solve it by analyzing the problem characteristics. We are also studying the co-ancestry minimization problem using the same theoretical approach to build efficient algorithms to combine the multi objectives of breeding in the same framework.

References

1. Abreu, U.G.P.d., Sonohata, M.M., Lopes, P.S.: Definição de pesos econômicos e de índices de seleção para sistemas de produção. In: Rosa, A.d.N., Martins, E.N., Menezes, G.R.d.O., Silva, L.O.C.d. (eds.) Melhoramento Genético Aplicado em Gado de Corte. Programa Geneplus-Embrapa, vol. 1, chap. 11. Embrapa Gado de Corte (2013)

2. Barreto Neto, A.D., et al.: Estrutura populacional e otimização de esquemas de acasalamento em ovinos com uso de algoritmos evolucionários (2014)
3. Cardoso, F.F.: Ferramentas e estratégias para o melhoramento genético de bovinos de corte. Embrapa Pecuária Sul-Documentos (INFOTECA-E) (2009)
4. Carvalheiro, R., Queiroz, S.A.d., Kinghorn, B.: Optimum contribution selection using differential evolution. Revista Brasileira de Zootecnia **39**(7), 1429–1436 (2010)
5. Carvalho, T., Santos, N., Lira, W., Oliveira, P.A., Neto, P.S., Lindenberg, J., Rabêlo, R.: Um sistema de informação para melhoramento genético de caprinos e ovinos. In: Anais Principais do XII Simpósio Brasileiro de Sistemas de Informação. pp. 100–107. SBC (2016)
6. Charlesworth, D., Willis, J.H.: The genetics of inbreeding depression. Nature Reviews Genetics **10**, 783–796 (2009)
7. Cormen, T.H., Leiserson, C.E., Rivest, R.L.: Introduction to Algorithms. MIT Press (2001)
8. Coster, A.: Package 'pedigree' (Feb 2015), https://cran.r-project.org/web//packages/pedigree/pedigree.pdf
9. Diestel, R.: Graph Theory. Graduate Texts in Mathematics, Springer (2005)
10. Dunne, F.L., Evans, R.D., Kelleher, M.M., Walsh, S.W., Berry, D.P.: Formulation of a decision support tool incorporating both genetic and non-genetic effects to rank young growing cattle on expected market value. Animal **15**, 100077 (2021)
11. Falconer, D.S., Mackay, T.F.C.: Introduction to Quantitative Genetics. Longman Group Ltd., Harlow, 4 edn. (1996)
12. FAO: Food and agriculture organization of the united nations (2009) how to feed the world 2050. Discussion paper prepared for Expert Forum: 12–13 October (2009)
13. 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 (Nov 2021), http://doi.org/10.1109/CLEI53233.2021.9640023
14. Ferreira, A.P.L.: On the problem of compensatory mating in animal breeding. In: Workshop-School on Theoretical Computer Science (WEIT). pp. 96–103 (2021), https://doi.org/10.5753/weit.2021.18928
15. Fontoura, D.C.N., Camargo, S.S., de Almeida Torres Junior, R.A., Carvalho, H.G., Cardoso, F.F.: Optimizing mate selection: a genetic algorithm approach. In: Proceedings of the ICAR Conference. ICAR Technical Series no. 24, Prague, CZ (2020)
16. Kinghorn, B.P.: An algorithm for efficient constrained mate selection. Genetics Selection Evolution **43**(1), 4 (2011)
17. Kuhn, H.W.: The Hungarian method for the assignment problem. Naval Research Logistics Quarterly **2**, 83–97 (1955)
18. Mi, M.P., Chapman, A.R., Tyler, W.J.: Effect of mating system on production traits in dairy cattle. J. Dairy Sci. **48**, 77–84 (1965)
19. Munkres, J.: Algorithms for the assignment and transportation problems. Journal of the Society for Industrial and Applied Mathematics **5**(1), 32–38 (March 1957)
20. Nieto, L.M., Alencar, M.M.d., Rosa, A.d.N.: Critérios de seleção. In: Rosa, A.d.N., Martins, E.N., Menezes, G.R.d.O., Silva, L.O.C.d. (eds.) Melhoramento Genético Aplicado em Gado de Corte. Programa Geneplus-Embrapa, vol. 1, chap. 10. Embrapa Gado de Corte (2013)
21. Rosa, A.d.N., Menezes, G.R.d.O., Egito, A.A.d.: Recursos genéticos e estratégias de melhoramento. In: Rosa, A.d.N., Martins, E.N., Menezes, G.R.d.O., Silva, L.O.C.d. (eds.) Melhoramento Genético Aplicado em Gado de Corte. Programa Geneplus-Embrapa, vol. 1, chap. 2. Embrapa Gado de Corte (2013)

22. Silverman, J., Ma, C., Buehren, M.: Package 'RcppHungarian' (Jan 2022), https://cran.r-project.org/web/packages/RcppHungarian/RcppHungarian.pdf
23. Simões, M.R.S., Leal, J.J.B., Minho, A.P., Gomes, C.C., MacNeil, M.D., Costa, R.F., Junqueira, V.S., Schmidt, P.I., Cardoso, F.F., Boligon, A.A., Yokoo, M.J.: Breeding objectives of brangus cattle in brazil. Journal of Animal Breeding and Genetics **137**(2), 177–188 (2020). https://doi.org/10.1111/jbg.12415, https://onlinelibrary.wiley.com/doi/abs/10.1111/jbg.12415
24. Sörensen, K., Glover, F.: Metaheuristics. Encyclopedia of operations research and management science **62**, 960–970 (2013)
25. Weigel, K.A., VanRaden, P.M., Norman, H.D., Grosu, H.: A 100-year review: Methods and impact of genetic selection in dairy cattle—from daughter–dam comparisons to deep learning algorithms. J. Dairy Sci. **100**(12), 10234–10250 (2017)
26. Yokoo, M.J.I., de Oliveira Seno, L., Oliveira, L.C., da Costa, P.U., da Silva, G.M., Suñé, R.W., Cardoso, F.F.: Economic selection index in small rural dairy farms. The Journal of Dairy Research **86**(1), 25–33 (2019)