Genet. Sel. Evol. 39 (2007) 3–25
© INRA, EDP Sciences, 2006
DOI: 10.1051/gse:2006031

Original article

Optimisation of contribution of candidate parents to maximise genetic gain and restricting inbreeding using semidefinite programming

(*Open Access publication*)

Ricardo PONG-WONG*, John A. WOOLLIAMS

Roslin Institute (Edinburgh), Roslin, Midlothian, EH25 9PS, Scotland, UK

(Received 19 February 2006; accepted 18 August 2006)

Abstract – An approach for optimising genetic contributions of candidates to control inbreeding in the offspring generation using semidefinite programming (SDP) was proposed. Formulations were done for maximising genetic gain while restricting inbreeding to a preset value and for minimising inbreeding without regard of gain. Adaptations to account for candidates with fixed contributions were also shown. Using small but traceable numerical examples, the SDP method was compared with an alternative based upon Lagrangian multipliers (RSRO). The SDP method always found the optimum solution that maximises genetic gain at any level of restriction imposed on inbreeding, unlike RSRO which failed to do so in several situations. For these situations, the expected gains from the solution obtained with RSRO were between 1.5–9% lower than those expected from the optimum solution found with SDP with assigned contributions varying widely. In conclusion SDP is a reliable and flexible method for solving contribution problems.

inbreeding restriction / optimisation of contributions / semidefinite-programming

1. INTRODUCTION

Response to selection in modern breeding schemes has been substantially increased due to the continuous improvement in the methods of genetic evaluation and high intensity of selection. This trend is expected to accelerate even more with the animal model BLUP rapidly becoming the standard in commercial selection schemes and the potential shown by the new generation of evaluation methods using DNA information [8, 9, 16, 19, 21, 22]. In parallel, the increase in reproduction capacity due to techniques such as AI (artificial

* Corresponding author: ricardo.pong-wong@bbsrc.ac.uk

Article published by EDP Sciences and available at http://www.edpsciences.org/gse or http://dx.doi.org/10.1051/gse:2006031

insemination) and MOET (multiple ovulation and embryo-transfer) has allowed for a higher intensity of selection previously not possible (*e.g.* [23]).

Despite the success in improving selection response, many of these techniques promote an increase in the rate of inbreeding (ΔF) through a higher probability of coselection of relatives [2, 25]. The increase in breeding risks arising from a high ΔF in a population has been reviewed (*e.g.* [26]). Therefore whilst inbreeding is unavoidable in finite populations, it is important to manage ΔF. Hence, the aim of selection should be to achieve the maximum rate of progress (ΔG) but restricted to a ΔF specified by the breeder.

Woolliams and Thompson [25] developed the theory where both ΔG and ΔF can be expressed as a function of the genetic contribution of parents and more distant ancestors. Under this framework, the genetic contribution of candidates to the next generation can be optimised taking into account their breeding values and relationships to maximise ΔG with ΔF controlled to a predefined level. This approach contrasts with traditional truncation selection where all selected candidates are expected to have equal numbers of offspring (contribution) in the next generation. The optimisation of contributions can be implemented using Lagrangian multipliers [6,7,11,15,27] and this approach has been shown to offer substantial improvement in ΔG (~20%) when compared to other forms of selection at the same rate of ΔF. Others (*e.g.* [13]) have advocated an approach to the solution using evolutionary algorithms.

Despite the potential of the methods, to date authors have only used approximate methods to solve the optimisation problem and these do not guarantee to find the optimum solution. In this study, we present a new method for optimising contribution to maximise ΔG with a restriction on inbreeding using semidefinite programming. The method can also be applied to minimise ΔF in conservation programmes where ΔG is not important. We illustrate this method with worked examples to compare results with the method proposed by Meuwissen [15].

2. MATERIALS AND METHODS

2.1. Theory and notation

Formulating the selection problem requires the expected breeding values of the set of n candidates for selection, given by $\mathbf{g}_t$, at a given round of selection t, and a vector to describe their potential genetic contribution to the next generation, $\mathbf{c}_t$. Since this paper will be concerned with describing events at a single time point the subscript t will be ignored. For a given candidate i, c_i refers to the fraction of genes that it contributes to the gene pool in the next generation. Therefore, $c_i \geq 0$ with $c_i = 0$ for a candidate that is not selected to be a parent, and with the total contributions summing to 1. In this formulation the expected genetic merit attained in the next generation is given by $\mathbf{c}^T\mathbf{g}$. In diploid species each sex contributes half the genes to the gene pool so that $\Sigma c_i = 0.5$ where the sum is over all individuals of a single sex, and c_i represents half the fraction of offspring obtained from this parent. Let $s_i = 1$ for a male and 0 for a female and $d_i = 1 - s_i$ then these restrictions are $\mathbf{c}^T\mathbf{s} = 0.5$ and $\mathbf{c}^T\mathbf{d} = 0.5$ with $c_i \geq 0$.

Although in principle it is possible that the total contribution originating from a given sex is from a single candidate (*i.e.* $c_i = 0.5$), because of biological limits to reproductive capacity or management policies the maximum contribution per each candidate may be restricted to a value lower than 0.5. The vector containing the maximum contribution for each candidate is denoted as $\mathbf{m}$. The maximum contribution m_i may vary across candidates and $m_i = 0.5$ for candidates with no restriction.

The restriction in inbreeding is controlled through consideration of the group coancestry of the selected parents, where the group coancestry is the inbreeding coefficient that would result from completely random mating among the parents, including selfing and given the contribution $\mathbf{c}$. For a set of contributions $\mathbf{c}$ among the candidates, the group coancestry is given by $\mathbf{c}^T\mathbf{A}\mathbf{c}/2$, where $\mathbf{A}$ is the Wright's numerator relationship matrix. If the group coancestry at time t was F_t, then due to the constraint to the target $\Delta\mathrm{F}$ it is required that $\mathrm{F}_{t+1} = 1 - (1 - \Delta\mathrm{F})(1 - \mathrm{F}_t)$. Since the paper is concerned only with obtaining the solution in one generation we will simply denote the target F_{t+1} by F^*.

Hence, the problem is to find the optimum genetic contributions, equivalent to the proportion of offspring, for candidates $\mathbf{c}$ in the form:

$$\begin{aligned} &\text{Maximise:} && \mathbf{c}^T\mathbf{g} && \text{(1a)}\\ &\text{Subject to:} && \mathbf{c}^T\mathbf{A}\mathbf{c}/2 \leq \mathrm{F}^* && \text{(1b)}\\ & && \mathbf{c}^T\mathbf{s} = 0.5 && \text{(1c)}\\ & && \mathbf{c}^T\mathbf{d} = 0.5 && \text{(1d)}\\ & && \mathbf{c} \geq \mathbf{0} && \text{(1e)}\\ & && \mathbf{c} \leq \mathbf{m} && \text{(1f)} \end{aligned}$$

where $\mathbf{0}$ is a column vector of zeros. Constraint (1b) is the restriction for the inbreeding to be less or equal to the predefined value; (1c) and (1d) are to ensure that the sum of contributions of candidates of a given sex is 0.5; (1e) and (1f) restrict the contributions to be within the valid parameter space. If no candidate has a restriction on their maximum contribution, *i.e.* $m_i = 0.5$

for all candidates, then restriction (1f) becomes redundant. The optimisation problem (1) is said to be feasible if there is at least one solution that satisfies all restrictions. Then the feasible solution which maximises $\mathbf{c}^T\mathbf{g}$ is the optimum solution.

2.2. Optimisation using semidefinite programming (SDP)

Semidefinite programming is a well-established methodology for optimising convex problems, where the space of all feasible solutions is convex. Since the intersection of convex sets is also convex, an optimisation problem is convex when its objective function and all its constraint functions are convex. A function f(x) is convex if it satisfies $\lambda f(x_1) + (1-\lambda)f(x_2) \geq f(\lambda x_1 + (1-\lambda)x_2)$, for any pair of values x_1 and x_2, and $0 < \lambda < 1$. For an extensive explanation of the methodology, a survey of applications and efficient algorithms, see Vandengerghe and Boyd [20] and Boyd and Vandengerghe [5].

Semidefinite programming minimises a linear objective function subject to a linear matrix inequality (LMI) [20]. Maximisation problems can be easily converted to this form by multiplying the objective function by -1. The standard form for a semidefinite programming problem is the following:

$$\text{Minimise:} \qquad \mathbf{a}^T\mathbf{x} \tag{2a}$$

$$\text{Subject to:} \qquad \mathbf{Y} \geq \mathbf{0},\ \mathbf{Y} = \mathbf{Y}_0 + \sum_{i=1}^{k} \mathbf{Y}_i x_i \tag{2b}$$

where $\mathbf{a}$ is the vector of "cost", $\mathbf{x}$ is the vector of k variables to be optimised, x_i is the i element of $\mathbf{x}$, $\mathbf{Y}$ is a positive semidefinite matrix with $k + 1$ affine matrices ($\mathbf{Y}_i$, $i = 0, 1, 2, \dots k$). The matrix inequality $\mathbf{Y} \geq \mathbf{0}$ means that $\mathbf{Y}$ is positive semidefinite.

To recast the optimisation problem in (1) into the semidefinite programming form we need to construct the LMI from the set of constraints. The matrix $\mathbf{Y}$ is constructed as a block diagonal matrix with the original constraints of the problem as the diagonal blocks. For illustration, assume the optimisation problem in (2) has the following two constraints: $\mathbf{b}^T\mathbf{x} + b_0 \geq 0$ and $-\mathbf{x}^T\mathbf{Q}\mathbf{x} + q_0 \geq 0$, where the first one is a linear constraint and the second is a (non-linear) quadratic constraint with $\mathbf{Q}$ being a positive definite matrix. Before the quadratic constraint can be included into the LMI, this equation is, first, transformed into a linear expression using *Schur complement* [20]. The linear form of $-\mathbf{x}^T\mathbf{Q}\mathbf{x} + q_0 \geq 0$ is equal to:

$$\begin{bmatrix} \mathbf{Q}^{-1} & \mathbf{x} \\ \mathbf{x}^T & q_0 \end{bmatrix} \geq \mathbf{0}. \tag{3}$$

Hence, the LMI containing the two inequality constraints is constructed by joining them in a single block diagonal matrix equal to:

$$\mathbf{Y} = \begin{bmatrix} \mathbf{b}^T\mathbf{x} + b_0 & \\ & \begin{bmatrix} \mathbf{Q}^{-1} & \mathbf{x} \\ \mathbf{x}^T & q_0 \end{bmatrix} \end{bmatrix} \geq \mathbf{0}, \tag{4a}$$

with the $m + 1$ affine matrices of $\mathbf{Y}$ defined by:

$$\mathbf{Y}_0 = \begin{bmatrix} b_0 & \\ & \begin{bmatrix} \mathbf{Q}^{-1} & \mathbf{0} \\ \mathbf{0} & q_0 \end{bmatrix} \end{bmatrix}, \quad \mathbf{Y}_i = \begin{bmatrix} b_i & \\ & \begin{bmatrix} \mathbf{0} & \mathbf{I}_i \\ \mathbf{I}_i^T & 0 \end{bmatrix} \end{bmatrix}, \quad i = 1, 2, ... \, k \tag{4b}$$

where $\mathbf{I}_i$ is the i column of an identity matrix of rank k, and b_i is the i element of the vector $\mathbf{b}$. The elements outside the diagonals blocks are zeros. Additional constraints of this form can easily be included into the problem by adding an extra block to the matrix diagonal.

An advantage of casting an optimisation problem as semidefinite programming is that they can be solved very efficiently using approaches such as interior point algorithms. These algorithms introduced by Karmarkar [12] for linear programming and later extended for semidefinite programming [17,18] are interactive methods, which search for the optimum solution within the strictly feasible parameter space. Currently, several general purpose programmes for solving SDP are already available (*e.g.* SDPA [10], CSDP [4], DSDP [3], SDPSOL [28]). Hence, from a practical point of view, the key process for using semidefinite programming is demonstrating the problem is convex and reformulating it in the standard form represented by (2), ready for using the available programmes.

A detailed description of interior point methods in SDP is not presented since the main objective is to show how the optimisation of contribution (1) can be reformulated as a SDP, but details may be found in [1,5,20]. Briefly they are iterative methods utilising a barrier function for the $\mathbf{Y}$ matrix to keep proposed solutions strictly within the space of feasible solutions. The methods involve identifying a central path consisting of analytic centres for the problem, where the sequence of centres is identified using Newton's method. In practice, the major complexity of implementing Newton's method is to solve a least squares problem of size k. Several methods implementing interior points to solve SDP problems have been proposed (see [1, 5, 20]) and these are categorised based on whether they solve the primal or the dual problem or both simultaneously. The latter category, called primal-dual methods, are more efficient in term of the steps they need to find the solution. For a comparison on the computer efficiency of primal-dual algorithms using Newton's method see [1].

2.3. Maximisation of genetic response with restricted inbreeding

Semidefinite programming is appropriate to the set of equations (1) because the problem is convex as all involved functions are convex. Consider the quadratic constraint $\mathbf{c}^T\mathbf{A}\mathbf{c}/2 \leq \mathrm{F}^*$: the space of feasible solutions for this constraint forms a convex set if it holds that $\lambda\left(\mathbf{c}_1^T\mathbf{A}\mathbf{c}_1/2\right)+(1-\lambda)\left(\mathbf{c}_2^T\mathbf{A}\mathbf{c}_2/2\right) \geq (\lambda\mathbf{c}_1 + (1-\lambda)\mathbf{c}_1)^T\,\mathbf{A}\,(\lambda\mathbf{c}_1 + (1-\lambda)\mathbf{c}_1)\,/2$, for any two solutions $\mathbf{c}_1$ and $\mathbf{c}_2$, and $0 < \lambda < 1$. Moving the expression at the right side of the inequality to the left, expanding it and rearranging the resulting terms, it can be shown that this inequality is equivalent to $\lambda(1-\lambda)\,(\mathbf{c}_1 - \mathbf{c}_2)^T\,\mathbf{A}\,(\mathbf{c}_1 - \mathbf{c}_2)\,/2 \geq 0$. Since λ is a positive constant <1 and $\mathbf{A}$ is a positive definite matrix representing the variance and covariance among the true breeding values, this inequality is always true. The linear functions in (1) are also convex and this can be shown in the same way as for the quadratic constraint.

Hence, the task is to reformulate the optimisation problem given in (1) into standard SDP form. The constraints mostly take the form as described previously however the quadratic constraint (1b) is given its linear form using *Schur complement* and the two equality constraints (1c) and (1d) are replaced with two inequality constraints. The problem (1) is, then, formulated as:

$$\text{Minimise:} \quad -\mathbf{c}^T\mathbf{g} \tag{5a}$$

$$\text{Subject to:} \quad \begin{bmatrix} \mathbf{A}^{-1} & \mathbf{c} \\ \mathbf{c}^T & 2\mathrm{F}^* \end{bmatrix} \geq \mathbf{0} \tag{5b}$$

$$\mathbf{c}^T\mathbf{s} - 0.5 \geq 0 \tag{5c}$$

$$-\mathbf{c}^T\mathbf{s} + 0.5 \geq 0 \tag{5d}$$

$$\mathbf{c}^T\mathbf{d} - 0.5 \geq 0 \tag{5e}$$

$$-\mathbf{c}^T\mathbf{d} + 0.5 \geq 0 \tag{5f}$$

$$\mathbf{c} \geq \mathbf{0} \tag{5g}$$

$$\mathbf{m} - \mathbf{c} \geq \mathbf{0} \tag{5h}$$

where (5b) is the linear equivalent of the quadratic constraint (1b), (5c) and (5d) are equivalent to constraint (1c), and the same for (5e) and (5f) for constraint (1d). This results in **Y** described by:

$$\mathbf{Y} = \begin{bmatrix} \begin{bmatrix} \mathbf{A}^{-1} & \mathbf{c} \\ \mathbf{c}^T & 2\mathrm{F}^* \end{bmatrix} & & & & & & \\ & \left[\mathbf{c}^T\mathbf{s} - 0.5\right] & & & & & \\ & & \left[-\mathbf{c}^T\mathbf{s} + 0.5\right] & & & & \\ & & & \left[\mathbf{c}^T\mathbf{d} - 0.5\right] & & & \\ & & & & \left[-\mathbf{c}^T\mathbf{d} + 0.5\right] & & \\ & & & & & [\mathbf{diag}(\mathbf{c})] & \\ & & & & & & [\mathbf{diag}\,(\mathbf{m} - \mathbf{c})] \end{bmatrix} \geq \mathbf{0} \tag{6a}$$

with the $(n + 1)$ affine matrices being:

$$\mathbf{Y_0} = \begin{bmatrix} \begin{bmatrix} \mathbf{A}^{-1} & \mathbf{0}_{(n\times1)} \\ \mathbf{0}_{(1\times n)} & 2\mathrm{F}^* \end{bmatrix} & & & & & & \\ & -0.5 & & & & & \\ & & +0.5 & & & & \\ & & & -0.5 & & & \\ & & & & +0.5 & & \\ & & & & & [\mathbf{0}_{(n\times n)}] & \\ & & & & & & [\mathbf{diag}(\mathbf{m})] \end{bmatrix} \tag{6b}$$

and

$$\mathbf{Y}_i = \begin{bmatrix} \begin{bmatrix} \mathbf{0}_{(n\times n)} & \mathbf{I}_i \\ \mathbf{I}_i^T & 0 \end{bmatrix} & & & & & & \\ & s_i & & & & & \\ & & -s_i & & & & \\ & & & d_i & & & \\ & & & & -d_i & & \\ & & & & & [\mathbf{diag}(\mathbf{I}_i)] & \\ & & & & & & [-\mathbf{diag}\,(\mathbf{I}_i)] \end{bmatrix}, \quad i = 1,\ 2, \ldots n \tag{6c}$$

where the size of the first block is $(n + 1) \times (n + 1)$, the next four are 1×1 and the last two blocks are of size $n \times n$. $\mathbf{0}_{(j\times k)}$ are matrices/vectors of zeros of size $j \times k$, $\mathbf{I}_i$ is the i column of the identity matrix of size $n \times n$ and $\mathbf{diag(I}_i\mathbf{)}$ is a diagonal matrix with diagonal equal to $\mathbf{I}_i$. All elements outside the block diagonal matrices are zero. Hence, with this reformulation the optimisation problem can now be solved using SDP.

Similarly to constrain (5h) imposing a restriction in the maximum contribution allowed from each candidate, restriction (5g) can also be modified to set a restriction on the minimum contribution (**h**) that is desired from each individual. The restriction (5g) becomes $\mathbf{c} - \mathbf{h} \geq \mathbf{0}$, where $h_i = 0$ for individuals with no restriction in their minimum contribution. Hence, for the SDP formulation, the sixth block diagonal in $\mathbf{Y}_0$ becomes –**diag(h)** rather than $\mathbf{0}_{(n\times n).}$ Using this approach, the contribution of individual i, may be fixed to a certain value very easily by setting $h_i = m_i$. Although setting $m_i = h_i$ may be the easiest and simplest way for fixing contribution of candidate i, a more computationally efficient approach would be to eliminate individuals with fixed contribution from the optimisation problem. The casting of the problem as SDP eliminating candidates with fixed contributions is given in Appendix A.

2.4. Minimisation of inbreeding without regard to genetic gain

Conservation programmes have as their sole objective the minimal accumulation of inbreeding without any regard to genetic gain, and this can also be achieved by optimising the genetic contributions of the parents. Semidefinite programming can be used to find the optimum contribution for each candidate to minimise the inbreeding in the next generation.

The objective here is to minimise the inbreeding function $\mathrm{F} = \mathbf{c}^T\mathbf{A}\mathbf{c}/2$, subject to the constraints that contributions are positive values between 0 and m_i and their sum within candidates of the same sex is 0.5. Since the objective function is quadratic, an auxiliary variable v is introduced that serves as the upper limit of the objective [20]. Then the optimisation problem is:

$$\text{Minimise:} \quad v \tag{7a}$$

$$\text{Subject to:} \quad \begin{bmatrix} \mathbf{A}^{-1} & \mathbf{c} \\ \mathbf{c}^T & 2v \end{bmatrix} \geq \mathbf{0} \tag{7b}$$

$$\mathbf{c}^T\mathbf{s} - 0.5 \geq 0 \tag{7c}$$

$$-\mathbf{c}^T\mathbf{s} + 0.5 \geq 0 \tag{7d}$$

$$\mathbf{c}^T\mathbf{d} - 0.5 \geq 0 \tag{7e}$$

$$-\mathbf{c}^T\mathbf{d} + 0.5 \geq 0 \tag{7f}$$

$$\mathbf{c} \geq \mathbf{0} \tag{7g}$$

$$\mathbf{m} - \mathbf{c} \geq \mathbf{0} \tag{7h}$$

where (7b) is the linear equivalent of $v - \mathbf{c}^T\mathbf{A}\mathbf{c}/2 \geq 0$, and the variables to optimise are v and **c**.

The LMI accounting for the seven constraints is:

$$\mathbf{Y} = \begin{bmatrix} \begin{bmatrix} \mathbf{A}^{-1} & \mathbf{c} \\ \mathbf{c}^T & 2\nu \end{bmatrix} & & & & & & \\ & \left[\mathbf{c}^T\mathbf{s} - 0.5\right] & & & & & \\ & & \left[-\mathbf{c}^T\mathbf{s} + 0.5\right] & & & & \\ & & & \left[\mathbf{c}^T\mathbf{d} - 0.5\right] & & & \\ & & & & \left[-\mathbf{c}^T\mathbf{d} + 0.5\right] & & \\ & & & & & [\mathbf{diag}(\mathbf{c})] & \\ & & & & & & [\mathbf{diag}\,(\mathbf{m} - \mathbf{c})] \end{bmatrix} \geq \mathbf{0} \tag{8a}$$

with the $(n + 2)$ affine matrices of $\mathbf{Y}$ equal to:

$$\mathbf{Y_0} = \begin{bmatrix} \begin{bmatrix} \mathbf{A}^{-1} & \mathbf{0}_{(n\times1)} \\ \mathbf{0}_{(1\times n)} & 0 \end{bmatrix} & & & & & & \\ & -0.5 & & & & & \\ & & +0.5 & & & & \\ & & & -0.5 & & & \\ & & & & +0.5 & & \\ & & & & & [\mathbf{0}_{(n\times n)}] & \\ & & & & & & [\mathbf{diag}\,(\mathbf{m})] \end{bmatrix}, \tag{8b}$$

$$\mathbf{Y}_i = \begin{bmatrix} \begin{bmatrix} \mathbf{0}_{(n\times n)} & \mathbf{I}_i \\ \mathbf{I}_i^T & 0 \end{bmatrix} & & & & & & \\ & s_i & & & & & \\ & & -s_i & & & & \\ & & & d_i & & & \\ & & & & -d_i & & \\ & & & & & [\mathbf{diag}(\mathbf{I}_i)] & \\ & & & & & & [-\mathbf{diag}\,(\mathbf{I}_i)] \end{bmatrix}, \quad i = 1,\ 2, \ldots\ n \tag{8c}$$

and

$$\mathbf{Y}_{n+1} = \begin{bmatrix} \begin{bmatrix} \mathbf{0}_{(n\times n)} & \mathbf{0}_{(n\times1)} \\ \mathbf{0}_{(1\times n)} & 2 \end{bmatrix} & & & & & & \\ & 0 & & & & & \\ & & 0 & & & & \\ & & & 0 & & & \\ & & & & 0 & & \\ & & & & & [\mathbf{0}_{(n\times n)}] & \\ & & & & & & [\mathbf{0}_{(n\times n)}] \end{bmatrix} \tag{8d}$$

where the size of the first block is $(n + 1) \times (n+1)$, the next four are 1×1 and the last two blocks of size $n \times n$. The missing elements outside the block diagonal matrices are zeros. The affine matrix $\mathbf{Y}_{n+1}$ corresponds to the auxiliary variable v.

When optimised, v is equal to the minimum expected inbreeding achievable in the next generation when candidates are assigned their optimum contribution.

2.5. The RSRO method

Here, we briefly describe an iterative method proposed and described in detail by Meuwissen [15] to optimise contributions for maximising ΔG and restricting ΔF. We have chosen to call this method Relaxed Successively Reduced Optimisation method (RSRO) as it obtains the solution through a series of relaxed optimisations using Lagrangian multipliers.

Briefly, the optimisation problem given in (1) is modified by converting the inequality restriction (1b) into equality so that $\mathbf{c}^T\mathbf{A}\mathbf{c}/2 = \mathrm{F}^*$, and eliminating restrictions (1e) and (1f), so the solution can be found using Lagrangian multipliers. Meuwissen [15] justified the modification to (1b) by pointing out that the objective function in (1) is linear, so the optimal solution to maximise gain, if it exists is at the boundary of the space limited by $\mathbf{c}^T\mathbf{A}\mathbf{c}/2 \leq \mathrm{F}^*$, where the equality holds.

However, the elimination of restriction (1e) can result in some individuals being assigned invalid negative contributions. This problem is solved by fixing the contribution for these candidates to be zero and repeating the optimisation in a smaller subset of candidates. This iterative process of repeating optimisations with fewer and fewer candidates is continued until a solution satisfying the restriction (1e) is found. A similar iterative process is also done to account for restriction on the maximum contribution (1f). In any iteration, candidates assigned contributions greater than their maximum have their values fixed to their maximum at all subsequent iterations. Since the restriction on the maximum contribution also imposes a requirement on the number of candidates to be selected, candidates assigned negative contribution should not be eliminated in the same iteration when another has its contribution fixed. Note that once a candidate is eliminated, it is no longer considered in further iterations and, therefore, is not selected in the final solution. Similarly when a candidate's contribution is fixed, it remains so in the further iterations and in the final solution.

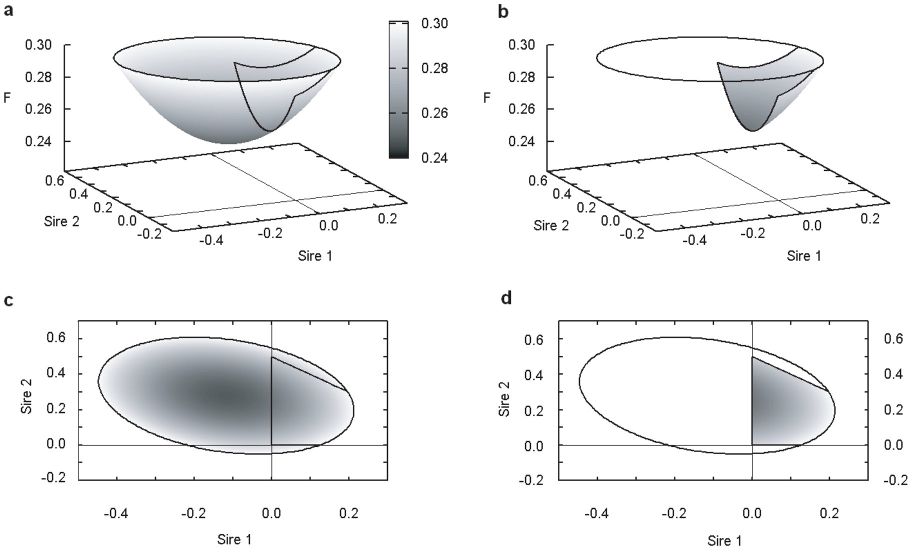

Figure 1. Surface representing the space of solutions for the constraints: $\mathbf{c}^T\mathbf{Ac}/2 \leq \mathrm{F}^*$ ($\mathrm{F}^* = 0.3$), $\mathbf{c}^T\mathbf{s} = 0.5$ and $\mathbf{c}^T\mathbf{d} = 0.5$ in a hypothetical example containing three male candidates. The z-axis is the expected F for the offspring, while the x- and y-axes are the contribution of sires 1 and 2, respectively. The contribution of sire 3 is 0.5-x-y. Figure 1a is the surface of solutions allowing for negative contributions, 1b is the surface of feasible solutions with $\mathbf{c} \geq 0$, hence the space of valid solutions, and Figures 1c and 1d are the corresponding contour plots.

A graphical representation of how RSRO searches for the solution is shown in Figure 1. Figure 1a shows the space of possible solutions for an hypothetical example containing three males candidates taking into account the restrictions $\mathbf{c}^T\mathbf{Ac}/2 \leq \mathrm{F}^*$, $\mathbf{c}^T\mathbf{s} = 0.5$ and $\mathbf{c}^T\mathbf{d} = 0.5$, which includes solutions with invalid negative contributions, and Figure 1b shows the space of solutions when the constraint $\mathbf{c} \geq 0$ is also taken into consideration thus the true space of feasible solutions. The modifications done to the constraints in RSRO means that, at any iteration, the solution is sought only at the ellipse curve delimiting the surface in Figure 1a, so any solution where F would be less than F^* is ignored. If the resulting solution is invalid, some candidates with negative contributions are eliminated or are fixed to their maximum contribution, and the optimisation redone with the remaining candidates until a valid solution is found. Any valid solution found with RSRO would lay on the section of the ellipse which is shown in Figure 1b.

Meuwissen [15] proposed that all candidates assigned negative contributions in an iteration be eliminated together. An alternative is to eliminate only the candidate with the most negative contribution or to fix the candidate exceeding their maximum by the greatest amount. This alternative way of eliminating or fixing candidates will increase the computational need as more iterations are required to obtain the final solution, but might give greater opportunity for finding the optimum solution.

3. RESULTS

In this section two numerical examples are given to demonstrate that, in certain situations, RSRO does not guarantee that the final solution is the global maximum, but rather a suboptimal solution. Each example contains six candidates: three male and three female candidates, where only the contributions of male candidates were optimised (female contributions were fixed to be 1/6). Optimising only the male contributions allowed the examples to be traceable and easy to illustrate. All feasible solutions could be evaluated so the true optimum was known with absolute certainty. Similar behaviour may also be observed when optimising a larger set of candidates, as they may contain clusters of individuals with similar structures as the small examples presented here. Genetic contributions of candidates were optimised in both examples using RSRO and SDP. The RSRO was carried out implementing the algorithm proposed by Meuwissen [15]. The SDP was done by reformulating the problem to the standard SDP form as described before, and using the software SDPA [10]. SDPA uses a primal-dual interior point algorithm for searching the solution during the optimization.

Example A

This example mimics a closed population where external sires are introduced. The relationship for the candidates was obtained from the pedigree shown in Figure 2. A pedigree with full sib mating was used to keep its graphical representation small and simple.

Figure 3 shows the contour plot for the surface of feasible solutions when imposing a restriction on the inbreeding in the offspring to be less or equal to 0.29. The solution from the optimisation using SDP assigned the contributions to be [0.0495, 0.0000, 0.4505] for sires 1, 2 and 3, respectively, with an expected genetic gain of 0.935. Testing all possible combinations confirmed that this was the optimum solution to maximise genetic gain.

On the other hand, the solution using RSRO depended on the approach for eliminating individuals assigned negative contributions. Under the restriction of $F^* \leq 0.29$, the solution from the first iteration of optimisation

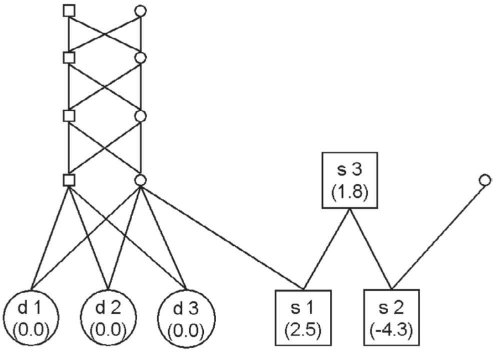

Figure 2. Pedigree structure for Example A. Values in bracket are their estimated breeding value and their relationship was calculated using the shown pedigree structure.

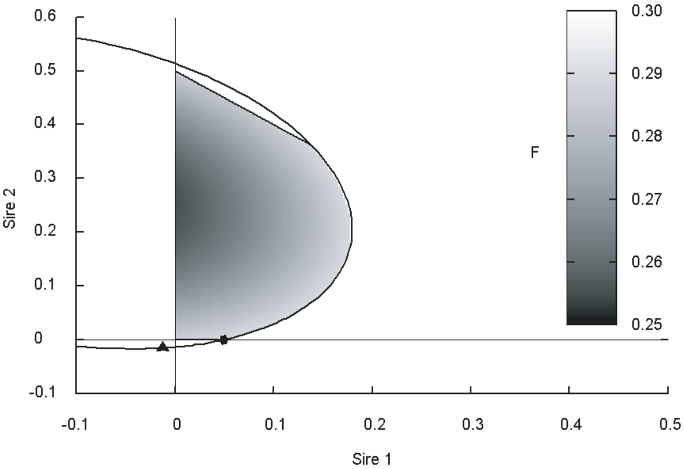

Figure 3. Contour plot showing the space of feasible solutions for Example A. The x- and y-axes are the contribution of sires 1 and 2, respectively. The contribution of sire 3 is 0.5-x-y. Shadow area is the space of feasible solutions. The ellipse curve is contour line for F = 0.29, and where RSRO searches for the solution. The solid triangle (▲) is the solution for first iteration of RSRO. The solid circle (●) is the optimum solution.

was [−0.012, −0.015, 0.527] (see Fig. 3). As sires 1 and 2 were assigned negative contributions they were eliminated, so only sire 3 remained to be selected for the final solution (*i.e.* **c** = [0.000, 0.000, 0.500]). This solution was shown to be suboptimal with an expected genetic gain 3.7% lower than the expected gain for the SDP solution (0.93 *vs.* 0.90). Similar results were found for other degrees of restriction on the maximum inbreeding, *e.g.* when the restriction on

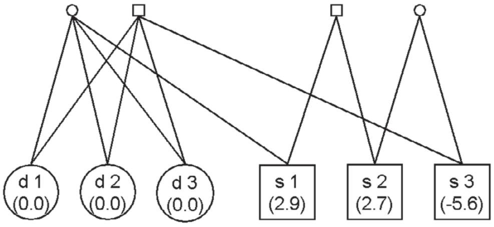

Figure 4. Pedigree structure for Example B. Values in bracket are their estimated breeding value and their relationship was calculated using the shown pedigree structure.

maximum inbreeding was set to 0.3 (**m** = 0.5), the expected gain for the RSRO solution was 8.8% lower than the maximum achievable (0.987 *vs.* 0.900).

In this example suboptimum solutions using RSRO were avoided if only one candidate was eliminated per iteration. Eliminating only the candidate with the most negative contribution increased the number of iterations but, for this example assuming **m** = 0.5, the final solution with RSRO was the true optimum. However, in different scenarios where the set of candidates is very large, the elimination of candidates one by one could have some practical implications when implementing RSRO. Additionally, the certainty with which the optimum would be found is still unclear.

Example B

The pedigree relating the candidates for Example B is shown in Figure 4. The simple half-sib relationship between the candidates is commonly found in practical situations. When no restriction in the maximum contribution was imposed (*i.e.* constraint (1f) was eliminated or **m** = 0.5), both RSRO and SDP yielded the same results. Testing the whole space of feasible solutions by evaluating all possible combinations of contributions of the candidate sires showed that the solutions were the true optimum given the constraint in the maximum inbreeding. For example, when $F^* = 0.2$ and **m** = 0.5, the optimum contributions are [0.296, 0.204, 0.000] for sires 1, 2 and 3, respectively, to give a gain of 1.41. This was found with both RSRO and SDP.

However, when a restriction in the maximum contribution was included, RSRO failed to find the optimum solution for many cases. For example, consider the case with $F^* = 0.2$, with a constraint in **m** added so sires cannot have a contribution higher than 0.3. The solution must be the same as when there is no restriction on **m** as it already fulfills this restriction, since $c_i \leq 0.3$ for all three sires. SDP managed to find the correct optimum when adding the restriction on **m**, but a suboptimal solution was obtained with RSRO, even when removing a candidate at a time. Examining the results from the first iteration of RSRO, the initial assigned contributions were [0.191, 0.370, −0.061]. Then, based in this invalid solution, RSRO fixed the contribution of sire 2, so the final solution obtained after further iterations was [0.200, 0.300, 0.000] compared with the true optimum [0.296, 0.204, 0.00]. For this particular example, with the extra restriction on **c** ≤ 0.3, RSRO failed to find the true optimum solution whenever F^* was imposed to be 0.188 or greater.

The expected gain for the solution from RSRO when a restriction on **m** was imposed was approximately 1.4% less than the optimum found with SDP (1.39 *vs.* 1.41), because both sires which were selected have similar breeding values. But when examining the differences in the contribution assigned to each candidate, RSRO assigned around 50% more contribution to sire 2 than the optimum (0.30 *vs.* 0.20).

Paradoxically, if the restriction on maximum contribution is less stringent, the problem of RSRO can become worse. For instance, the optimum solution found by SDP when $F^* = 0.22$ and **m** = 0.4 was [0.375, 0.125, 0.000]. Whereas the final solution obtained with RSRO was [0.100, 0.400, 0.000], with sire 2 being assigned over 3-fold its optimum contribution. With **m** = 0.4, RSRO failed to find the optimum value in any case whenever F^* was set to be 0.215 or higher.

An interesting observation is that if candidates are ordered by their assigned contribution, their ranking when considering the optimum solution is not the same as in previous rounds of optimisation of RSRO (*e.g.* in the optimum solution, the candidate assigned the greatest contribution was sire 1, but this was not the case in the first iteration). Hence, one would extrapolate that the candidate assigned the most negative contribution may not necessarily be the one with the least chance of being selected. So the elimination of candidates one by one still has the risk of wrongly removing a candidate which should be included in the optimum solution.

Intuitively the restriction on the maximum contribution also imposes a requirement on the number of candidates to be selected and the option of fixing candidates in RSRO (as above) appears preferable. However, in Example B, if sire 3 would have been eliminated before sire 2 was fixed, the final solution would have been the same as the one found with SDP. But this is not a general panacea: for example, consider Example B where $F^* = 0.181$ and **m** = 0.2. Although both SDP and RSRO find the optimum solution of [0.200, 0.200, 0.100] with an expected gain of 0.56, this is not the case if another unrelated sire 4 with breeding value equal to −6.0 is included in the optimisation. The optimum solution for the extended set of candidates is the same since the extra candidate has lower breeding value than the original three candidates. However, in the first iteration of RSRO, the assigned contribution was [0.161, 0.356, −0.109, 0.092], where the extra sire 4 was preferred over sire 3 despite having a lower breeding value. Hence, if the elimination of candidates is done before others are fixed, sire 3 would be eliminated and the final solution would be [0.200, 0.200, 0.000, 0.100] with an expected gain of 0.52 (*i.e.* 7.1% less than the optimum). Hence, the correct decision on whether to fix or to eliminate candidates at a given iteration of RSRO is very specific to the situation being optimised and no general rules can be applied. The small size of Example B allows all options to be tested to determine which decision was the correct one, but this would not be practical in a much larger data set.

4. DISCUSSION

In this study a new method based upon semidefinite programming was implemented to optimise genetic contributions of candidates to selection for maximising genetic gain while restricting inbreeding. The method was also implemented to minimise inbreeding without regard to genetic gain (results not shown). Using numerical examples it was shown that the proposed methodology would find the optimum solution even in situations where a previous method based in Lagrangian multipliers failed to do so.

The main advantage of the proposed method is that the supporting theory guarantees that the optimum solution is found in an efficient manner. The theoretical tractability from convexity and the development of primal-dual interior point algorithms ensure the solution found using the SDP framework would be the true optimal [5, 20], unlike the approach used in RSRO. In RSRO, the replacement of the inequality in (1b) with equality and the iterative process of eliminating candidates or fixing their contributions, in essence, re-direct and restrict the search for the optimum to a smaller space. Whilst focusing the search only at the boundary (by using $\mathbf{c}^T\mathbf{A}\mathbf{c}/2 = \mathrm{F}^*$) may be justified, the examples show that the use of invalid solutions as criteria to narrow the search space in this way does not have theoretical validity, with the possibility that the optimum solution be eliminated during this process. Eliminating one candidate per iteration in Example A improved the performance of RSRO, but did not prevent the contribution of a candidate in Example B being fixed when it should not be. Moreover, whether or not RSRO yielded the true optimum solution when imposing a restriction on $\mathbf{m}$, depended on the decision on the order of fixing or on the elimination of candidates which appeared case specific. Additionally, the results from Example B showed that the ranking of candidates according to their optimum contribution differ with their ranking in previous RSRO iterations. Hence, the candidate assigned the most negative contribution is not necessarily the least likely to be selected in the optimum solution, so eliminating only this candidate would not prevent that the wrong candidate be eliminated or fixed.

Another desirable characteristic of the SDP framework is that new complex constraints can easily be added to the optimisation. Provided the new restrictions are convex functions (which include linear and quadratic functions) they are accounted for by adding them as an extra block diagonal in the LMI. In principle, RSRO can also be extended to take into account other restrictions. However, it requires the reformulation and solving a new set of equations for the extra Lagrangian multipliers introduced with the extra constraints. The properties of an extended RSRO with the new set of constraints are unclear when it involves additional quadratic restrictions and or inequalities. For instance, if the intersection of two quadratic inequality constraints does not include the boundaries for one of them, converting the inequalities to equalities will result in RSRO finding no feasible solution either optimal or suboptimal. Interior point algorithms commonly used for searching the solution in SDP would always search the solution within the whole feasible space, hence the optimum solution can be found.

The set of constraints included in the optimisation problem given in (1) are the basic restrictions required to control inbreeding while maximising genetic gain, but other restrictions may be included into the optimisation to suit the breeders' objectives, provided they are convex. However, such constraints include a number which may be of value to the breeders. For instance, the variance of the response can be accounted for by adding a quadratic constraint involving the prediction error variance (PEV) matrix for the estimated breeding values [24]. The level of inbreeding at specific locations of the genome may also be restricted using the genetic relationships among individuals at the location in question estimated with linked marker information. Changes in allele frequency at specific major genes or QTL may also be controlled by including additional constraints using candidate genotypes, for example, to optimise the fixation rate of the QTL [8].

The output from both the SDP algorithm used here and the RSRO are expressed as proportions on a continuous scale, although subsequently these need to be interpreted as an integer number of matings or offspring. An *ad hoc* solution is to round the values scaled by twice the total number of matings (offspring) to the nearest integer, but the optimality of the final result may not be guaranteed longer, especially when population size is small. A refinement of the optimisation of contributions would be to implement an integer programming to express the output of the optimisation as the number of offspring, so the problem arising from "rounding" the solution is avoided and the optimality of the solution retained. Integer programming under a SDP framework is possible and for further information about its implementation see [14]. The implementation of integer programming using the RSRO framework may also be possible but the optimality problems would remain. The use of integer programming may also be useful for other practical constraints of breeding programmes. For instance, where differential contributions may not be practical and the solution to equation (1) is required in the form of a simple sequence of 1's and 0's where 1 indicates select for breeding, and 0 indicates cull. However, the inability to utilise differential contributions will have consequences for genetic gain.

From a more practical point of view, another advantage of SDP is the availability of general purpose softwares for solving SDP optimisation problems. Currently, several programmes such as SDPA [10], CSDP [4], DSDP [3] and SDPSOL [28] are already available. Hence from a practical point of view, the key process to optimise contributions using SDP is to re-cast the problem in the standard form as shown here. However, a characteristic of interior point algorithm is that the data structure influences its computational efficiency. Hence, the development of a specialised algorithm considering specific characteristics and properties of the **A** matrix may be of significant benefit for the practical implementation of the optimisation.

The relative computational efficiency of the RSRO and the SDP methods were beyond the objectives of this study and would require implementing methods using the same programming languages and operating systems. However, in general, the performance of both methods appeared similar within our study, with RSRO slightly faster especially with small data sets and with all candidates assigned negative contributions eliminated simultaneously. One of the characteristics of interior point algorithms is that they are very computer efficient for solving SDP problems with 1000s of variables and constraints [20]. Currently, a version of SDPA able to efficiently solve problems with a matrix size of 20 000 is already available [10]. Our experience when optimising the contributions of large data sets (hundreds of candidates) showed that the computational efficiency of SDP was comparable with those obtained with RSRO (results not shown). The main task when implementing the Newton's method for interior points is mainly to solve a least-squares problem with size equal to the number of candidates and, using a common animal breeding task as analogy, the computational need for each interior-point step would be similar to that required to perform a BLUP evaluation with an equal number of individuals in the pedigree. Vandengerghe and Boyd [20] pointed out that the rule-of-thumb is that interior point may solve an SDP problem with 5–50 iterations, then the computational need would be comparable to performing 5–50 BLUP iterations. Methods to solve SDP problems can also account for data structure of the problem and exploit the sparseness of the LMI to improve its computational efficiency since computational demand depends on the number of non-zero elements in the LMI matrix. For the case of optimisation of contributions, including an extra candidate increases the LMI by 3 extra rows but many fewer non-zero elements are added to the **Y** matrix (see equations (6) and (8)).

Another method used to maximise gain with a constraint in inbreeding is evolutionary algorithms [13]. The comparison between the performance of SDP and evolutionary algorithms for optimising contribution has not been done yet, but because of the stochastic nature of genetic algorithms, there is never absolute certainty that the solution is optimal. In principle, genetic algorithms can cope with a large number of constraints of variable form, but their convergence rate decreases according to the complexity of the constraints, so the accuracy of their performance may further decrease if new restrictions are added to the problem.

In conclusion, the problem of obtaining optimum solution for managing genetic variation either in maximising ΔG with restricted ΔF or simply minimising ΔF irrespective of gain can be solved using SDP. The strength of SDP is that it guarantees the finding of the optimum, unlike other computational approaches. One of the benefits of the method is that it can encompass extended problems including constraining PEV and or separately managing genetic variation at different locations within the genome.

ACKNOWLEDGEMENTS

The authors acknowledge financial support from the Biotechnology and Biological Sciences Research Council.

REFERENCES

[1] Alizadeh F., Haeberly J.P.A., Overton M.L., Primal-dual interior point methods for semidefinite programming: convergence rates, stability and numerical results, SIAM J. Optim. 8 (1998) 746–768.

[2] Belonsky G.M., Kennedy B.W., Selection on individual phenotype and best linear unbiased predictor of breeding value in a closed swine herd, J. Anim. Sci. 66 (1988) 1124–1131.

[3] Benson S.J., Yinyu Y., DSDP5: Software for Semidefinite programming, ACM Trans. Math. Software (2005) http://www.stanford.edu/~yyye/DSDP5-Paper.pdf.

[4] Borchers B., CSDP, A C library for semidefinite programming, Optim. Methods Softw. 11 (1999) 613–623.

[5] Boyd S., Vandenberghe L., Convex programming, Cambridge University Press (2005).

[6] Brisbane J.R., Gibson J.P., Balancing selection response and rate of inbreeding by including genetic-relationships in selection decisions, Theor. Appl. Genet. 91 (1995) 421–431.

[7] Colleau J.J., Moureaux S., Briend J.B., A method for the dynamic management of variability in dairy cattle, Genet. Sel. Evol. 36 (2004) 373–394.

[8] Dekkers J.C.M., van Arendonk J.A.M., Optimizing selection for quantitative traits with information on an identified locus in outbred populations, Genet. Res. Camb. 71 (1998) 257–275.

[9] Fernando R.L., Grossman M., Marker assisted selection using best linear unbiased prediction, Genet. Sel. Evol. 21 (1989) 467–477.

[10] Fujisawa K., Kojima M., Nakata K., Yamashita M., SDPA (SemiDefinite Programming Algorithm) user's manual – version 6.00. Research Reports on Mathematical and Computer Sciences (2002) Series B: Operations Research.

[11] Grundy B., Villanueva B., Woolliams J.A., Dynamic selection procedures for constrained inbreeding and their consequences for pedigree development, Genet. Res., Camb. 72 (1998) 159–168.

[12] Karmarkar N., A new polynomial-time algorithm for linear programming, Combinatorica 4 (1984) 373–395.

[13] Kinghorn B.P., Meszaros S.A., Vagg R.D., Dynamic tactical decision systems for animal breeding, in: Proc. 7th World Congr. Genet. Appl. Livest. Prod. (2002) CD-ROM Communications 23-07.

[14] Laurent M., Rendl F., Semidefinite programming and integer programming, in: Ardal K., Nemhauser G.L., Weismantel R. (Eds.), Handbooks in Operations Research and Management Science: Discrete Optimization, Elsevier, Amsterdam (2005) pp. 393–514.

[15] Meuwissen T.H.E., Maximizing the response of selection with a predefined rate of inbreeding, J. Anim. Sci. 75 (1997) 934–940.

[16] Meuwissen T.H.E., Hayes B.J., Goddard M.E., Prediction of genetic value using genome-wide dense marker maps, Genetics 157 (2001) 1819–1829.

[17] Nesterov Y., Nemirovsky A., A general approach to polynomial-time algorithms design for convex programming, Technical Report (1988) Centr. Econ. & Math. Inst., USSR Acad. Sci., Moscow, USSR.

[18] Nesterov Y., Nemirovsky A., Interior-point polynomial methods in convex programming, Studies in Applied Mathematics, Vol. 13. SIAM, Philadelphia, PA (1994).

[19] Pong-Wong R., Woolliams J.A., Response to mass selection when an identified major gene is segregating, Genet. Sel. Evol. 30 (1998) 313–337.

[20] Vandenberghe L., Boyd S., Semidefinite programming, SIAM Review 38 (1996) 49–95.

[21] Villanueva B., Pong-Wong R., Grundy B.,Woolliams J.A., Potential benefit from using an identified major gene and BLUP estimated breeding values in selection programmes, Genet. Sel. Evol. 31 (1999) 115–133.

[22] Villanueva B., Pong-Wong R., Woolliams J.A., Marker assisted selection with optimized contributions of the candidates to selection, Genet. Sel. Evol. 34 (2002) 679–703.

[23] Woolliams J.A., Modifications to MOET nucleus schemes to improve rates of genetic progress and decrease rates of inbreeding in diary-cattle, Anim. Prod. 49 (1989) 1–14.

[24] Woolliams J.A., Meuwissen T.H.E., Decision rules and variance of response in breeding schemes, Anim. Prod. 56 (1993) 179–186.

[25] Woolliams J.A., Thompson R., A theory of genetic contributions, Proc. 5th World Cong. Genet. Appl. Livest. Prod. 25 (1994) 127–134.

[26] Woolliams J.A., Pong-Wong R., Villanueva B., Strategic optimisation of short- and long- term gain and inbreeding in MAS and non-MAS schemes, in: Proc. 7th World Congr. Genet. Appl. Livest. Prod. (2002) CD-ROM Communications 23-02.

[27] Wray N.R., Goddard M.E., Increasing long-term response to selection, Genet. Sel. Evol. 26 (1994) 431–451.

[28] Wu S.P., Boyd S., DPSOL: A parser/solver for semidefinite programs with matrix structure, in: El Ghaoui L., Niculescu S.I. (Eds.), Advances in linear matrix inequality methods in control, SIAM, 2000, pp. 79–91.

APPENDIX A: OPTIMISATION WITH CANDIDATES WITH FIXED CONTRIBUTIONS

In this appendix we will show a more computationally efficient way to account for candidates with fixed contributions. Let assume n_o candidates with contributions that may vary and need to be optimised and n_f candidates with fixed contributions ($n = n_f + n_o$). Ordering candidates with the ones to be optimised being first:

$$\mathbf{c} = \begin{bmatrix} \mathbf{c}_o \\ \mathbf{c}_f \end{bmatrix}, \quad \mathbf{g} = \begin{bmatrix} \mathbf{g}_o \\ \mathbf{g}_f \end{bmatrix}, \quad \mathbf{s} = \begin{bmatrix} \mathbf{s}_o \\ \mathbf{s}_f \end{bmatrix}, \quad \mathbf{d} = \begin{bmatrix} \mathbf{d}_o \\ \mathbf{d}_f \end{bmatrix} \quad \text{and} \quad \mathbf{A} = \begin{bmatrix} \mathbf{A}_{oo} & \mathbf{A}_{fo} \\ \mathbf{A}_{fo} & \mathbf{A}_{ff} \end{bmatrix}$$

where $\mathbf{c}_f$ is the vector containing the contribution assigned to candidates with fixed contribution.

Then the problem is to find the optimum $\mathbf{c}_o$ such as:

$$\begin{aligned}
&\text{Minimise:} && -\mathbf{c}_o^T\mathbf{g}_o && \text{(Aa)}\\
&\text{Subject to:} && 2\mathrm{F}^* - \mathbf{c}_f^T\mathbf{A}_{ff}\mathbf{c}_f - 2\mathbf{c}_o^T\mathbf{A}_{of}\mathbf{c}_f - \mathbf{c}_o^T\mathbf{A}_{oo}\mathbf{c}_o \geq 0 && \text{(Ab)}\\
& && \mathbf{c}_f^T\mathbf{s}_f + \mathbf{c}_o^T\mathbf{s}_o - 0.5 \geq 0 && \text{(Ac)}\\
& && -\mathbf{c}_f^T\mathbf{s}_f - \mathbf{c}_o^T\mathbf{s}_o + 0.5 \geq 0 && \text{(Ad)}\\
& && \mathbf{c}_f^T\mathbf{d}_f + \mathbf{c}_o^T\mathbf{d}_o - 0.5 \geq 0 && \text{(Ae)}\\
& && -\mathbf{c}_f^T\mathbf{d}_f - \mathbf{c}_o^T\mathbf{d}_o + 0.5 \geq 0 && \text{(Af)}\\
& && \mathbf{c}_o \geq \mathbf{0} && \text{(Ag)}\\
& && \mathbf{m}_o - \mathbf{c}_o \leq \mathbf{0} && \text{(Ah)}
\end{aligned}$$

Note that sometimes all candidates from a given sex can have fixed contributions. If this is the case, the pair of constraints to ensure that their contributions sum to 0.5 become redundant and they can be dropped out from the optimisation. An example where this may be appropriate is when female candidates have a single offspring per litter, so optimisation is done only in males with contributions of females fixed. In this situation constraints (Ae) and (Af) are eliminated from the problem.

Hence, the **Y** matrix forming the LMI is:

$$\mathbf{Y} = \mathbf{diag}\begin{bmatrix}
\begin{bmatrix} \mathbf{A}_{oo}^{-1} & \mathbf{c}_o \\ \mathbf{c}_o^T & 2\mathrm{F}^* - 2\mathbf{c}_f^T\mathbf{A}_{fo}\mathbf{c}_o - \mathbf{c}_f^T\mathbf{A}_{ff}\mathbf{c}_f \end{bmatrix}\\
\left[\mathbf{c}_o^T\mathbf{s}_o + \mathbf{c}_f^T\mathbf{s}_f - 0.5\right]\\
\left[-\mathbf{c}_o^T\mathbf{s}_o - \mathbf{c}_f^T\mathbf{s}_f + 0.5\right]\\
\left[\mathbf{c}_o^T\mathbf{d}_o + \mathbf{c}_f^T\mathbf{d}_f - 0.5\right]\\
\left[-\mathbf{c}_o^T\mathbf{d}_o - \mathbf{c}_f^T\mathbf{d}_f + 0.5\right]\\
[\mathbf{c}_o]\\
[(\mathbf{m}_o - \mathbf{c}_o)]
\end{bmatrix} \geq \mathbf{0}$$

with the $n_o + 1$ affine matrices:

$$\mathbf{Y}_0 = \mathbf{diag}\left[\begin{array}{l} \begin{bmatrix} \mathbf{A}_{oo}^{-1} & \mathbf{0}_{(n_o\times 1)} \\ \mathbf{0}_{(1\times n_o)} & 2\mathrm{F}^* - \mathbf{c}_f^T\mathbf{A}_{ff}\mathbf{c}_f \end{bmatrix} \\ \left[\mathbf{c}_f^T\mathbf{s}_f - 0.5\right] \\ \left[-\mathbf{c}_f^T\mathbf{s}_f + 0.5\right] \\ \left[\mathbf{c}_f^T\mathbf{d}_f - 0.5\right] \\ \left[-\mathbf{c}_f^T\mathbf{d}_f + 0.5\right] \\ \left[\mathbf{0}_{(1\times n_o)}\right] \\ \left[\mathbf{m}_o\right] \end{array}\right],$$

$$\mathbf{Y}_i = \mathbf{diag}\left[\begin{array}{l} \begin{bmatrix} \mathbf{0}_{(n_o\times n_o)} & \mathbf{I}_i \\ \mathbf{I}_i^T & -2\mathbf{c}_f^T\mathbf{A}_{fo}\mathbf{I}_i \end{bmatrix} \\ s_i \\ -s_i \\ d_i \\ -d_i \\ \left[\mathbf{I}_i\right] \\ \left[\mathbf{I}_i\right] \end{array}\right], \quad i = 1 \ldots n_o.$$

Now the problem can be solved as a standard SDP. This is more computationally efficient because the number of variables is n_o while with the formulation in the main text is $n_o + n_f$.