Optimizing the reference population in a genomic selection design

Jean-Michel Elsen[1], Simon de Givry[2], George Katsirelos[2], and Felicien Shumbusho[1]

[1] SAGA, UR 631, INRA, F-31320 Castanet Tolosan, France
[2] MIA-T, UR 875, INRA, F-31320 Castanet Tolosan, France
{Jean-Michel.Elsen,degivry,george.katsirelos,Felicien.Shumbusho}@toulouse.inra.fr

Abstract. In genomic selection, when candidate animals for reproduction are selected on an estimate of their breeding value from genomic information (using Single Nucleotide Polymorphims (SNP) chips), it is needed to build a reference population whose members are both genotyped on the SNPs and phenotyped for the economical trait(s) to be improved. We studied, with numerical simulations of such genomic selection plan, how to optimize the design of this reference population. The problem is summarized as minimizing a quadratic function on Boolean variables with a cardinality constraint. Integer linear/quadratic/constraint programming and weighted Max-SAT and CSP solvers are compared on a few examples.

1 Introduction

Thanks to the discovery of very abundant Single Nucleotide Polymorphisms (SNP) and availability of high throughput genotyping technologies, *genomic selection*, as described by [8] more than ten years ago, became realistic and rapidly turned to be the new standard in Dairy cattle breeding schemes [16]. Its application to other species is still a matter of discussion, as described for instance by [18] in pig or [17] in sheep. Genomic selection schemes comprise two steps. The estimation step, performed from phenotypes and genotypes recorded in a *reference population*, provides estimations of SNPs effects on the quantitative trait of interest. Different models were proposed for these estimations, the simplest, *Genomic Best Linear Unbiased Prediction* (GBLUP), modeling the performance as the sum of fixed nuisance effects and all SNPs random effects with a prior in a Gaussian distribution of known variance [8]. The selection step comprises an estimation of *Genomic Breeding Values* (GBV) merging the genotypic information about each candidate and the SNP effects previously estimated.

Amongst other factors, the efficiency of genomic selection largely depends on the design of the reference population [1, 11]. There are increasing evidence that closer the reference population to the selected population is, more precise the genomic evaluation will be. As an example, between breeds designs with SNPs estimated in a breed and selection candidates belonging to another breed (e.g. Jersey and Holstein breeds in dairy cattle) are efficient only with very dense SNP chips [14].

The present work aims at providing a tool for optimizing the reference population design. Populations displaying realistic linkage disequilibrium structures were simulated. Efficiency of different reference population designs were evaluated from the mean correlation between true and GBLUP estimated breeding values. As in [13], this criterion was used as an objective function to be maximized given a constraint of the reference population size. This paper describes a new approach to perform this optimization using a Taylor approximation in the framework of integer linear/quadratic/constraint programming and weighted Max-SAT/CSP.

2 The genomic selection design problem

The phenotyped population has n_p individuals. Among them, we want to select n_r individuals, forming the reference population, to be genotyped on m markers. The candidate population has n_c individuals, different of those in the phenotyped population. These candidate individuals are assumed to be already genotyped.

We assume a GBLUP linear mixed model [19] for the observed phenotypes of the reference population with the genetic effects modeled as random effects (and no fixed effects for the purpose of this study). In matrix notation, we have:

$$y = Xq + e$$

where $y = (y_1, \ldots, y_{n_r})$ is the column vector of observed (single value) phenotypes for the reference population, $X = (\forall l \in [1, n_r], \forall i \in [1, m] \quad x_{li})$ the matrix of recentered genotypes for the reference population with n_r rows and m columns, $q = (q_1, \ldots, q_m)$ is the column vector of m random genetic effects, and e is a vector of independent and identically distributed random error terms representing an environmental deviation. For each genotype, we have $x_{li} = a_{li} - 2f_i$, where $a_{li} \in \{0, 1, 2\}$ is the number of alleles A_i possessed by individual l at marker i (having two possible alleles A_i, B_i), and f_i is the frequency of A_i in the population.

q and e follow a normal distribution with zero mean and different variances: $\forall i \in [1, m], q_i \sim \mathcal{N}(0, \sigma_q^2)$ and $e \sim \mathcal{N}(0, \sigma_e^2)$. We denote $\lambda = \frac{\sigma_e^2}{\sigma_q^2}$, a known parameter value in our simulation. It can be shown that λ is related to heritability h^2 of the observed phenotypes: $\lambda = \frac{(1-h^2)2\sum_i^m f_i(1-f_i)}{h^2}$.

The estimation of the random genetic effects $\hat{q} = (\hat{q}_1, \ldots, \hat{q}_m)$ is obtained by the following formula [19]:

$$\hat{q} = (X^T X + \lambda I)^{-1} X^T y$$

We define the quality of this estimation on the candidate population by the mean square Pearson correlation $r^2_{g,\hat{g}} = \frac{1}{n_c} \sum_k^{n_c} r^2_{g_k,\hat{g}_k}$, by marginalizing the phenotypes, where $g_k = w_k q$ is the genotypic value of individual k and $\hat{g}_k = w_k \hat{q}$ its estimate, with $w_k = (w_{k1}, \ldots, w_{km})$ is the row vector of recentered genotypes of individual k in the candidate population.

Using standard calculus we get:

$$r^2_{g_k,\hat{g}_k} = \frac{cov^2(g_k,\hat{g}_k)}{var(g_k)var(\hat{g}_k)} = \frac{var(\hat{g}_k)}{var(g_k)} = 1 - \lambda \frac{w_k(X^TX+\lambda I)^{-1}w_k^T}{w_k{w_k}^T}$$

Our goal is to maximize the quality of the estimation, that is to minimize:

$$\begin{aligned} D(X) &= \lambda \sum_k^{n_c} \frac{w_k(X^TX+\lambda I)^{-1}w_k^T}{w_k{w_k}^T} \\ &= \lambda \sum_k^{n_c} \tilde{w}_k(X^TX+\lambda I)^{-1}\tilde{w}_k^T \end{aligned}$$

with $\forall k \in [1,n_c], \forall i \in [1,m] \quad \tilde{w}_{ki} = \frac{w_{ki}}{\sqrt{\sum_j^m w_{kj}^2}}$, the normalized genotypes in the candidate population.

For $m = 2$, we have:

$$\begin{aligned} D(X) &= \lambda \sum_k^{n_c} (\tilde{w}_{k1},\tilde{w}_{k2})(X^TX+\lambda I)^{-1}(\tilde{w}_{k1},\tilde{w}_{k2})^T \\ &= \lambda \sum_k^{n_c} \frac{\tilde{w}_{k1}^2(v_2+\lambda)+\tilde{w}_{k2}^2(v_1+\lambda)-2\tilde{w}_{k1}\tilde{w}_{k2}c}{(\tilde{w}_{k1}^2+\tilde{w}_{k2}^2)((v_1+\lambda)(v_2+\lambda)-c^2)} \end{aligned}$$

with $v_1 = \sum_l^{n_r} x_{l1}^2$, $v_2 = \sum_l^{n_r} x_{l2}^2$, and $c = \sum_l^{n_r} x_{l1}x_{l2}$.

For the general case, we will approximate the matrix inversion by using a Taylor approximation. In the case of a Taylor approximation of order 1, we have:

$$\begin{aligned} D(X) &= \lambda \sum_k^{n_c} \tilde{w}_k(X^TX+\lambda I)^{-1}\tilde{w}_k^T \\ &= \sum_k^{n_c} \tilde{w}_k(\frac{X^TX}{\lambda}+I)^{-1}\tilde{w}_k^T \\ &\approx \sum_k^{n_c} \tilde{w}_k(I-\frac{X^TX}{\lambda})\tilde{w}_k^T \\ &\approx \sum_k^{n_c}\sum_i^m \tilde{w}_{ki}^2 - \frac{1}{\lambda}\sum_k^{n_c}\sum_i^m \tilde{w}_{ki}\sum_j^m \tilde{w}_{kj}\sum_l^{n_r} x_{li}x_{lj} \end{aligned}$$

We can rewrite this objective function by introducing Boolean variables $\delta_l \in \{0,1\}$ for all individuals in the phenotyped population ($l \in [1,n_p]$). We denote z_{li} the recentered genotype of individual l at marker i in this population (whereas x_{li} is on the reference population).

We have:

$$D(X) \approx \sum_{k}^{n_c}\sum_{i}^{m} \tilde{w}_{ki}^2 - \frac{1}{\lambda}\sum_{k}^{n_c}\sum_{i}^{m} \tilde{w}_{ki}\sum_{j}^{m} \tilde{w}_{kj}\sum_{l}^{n_p} \delta_l z_{li} z_{lj}$$

$$D(X) \approx D_1(X) = \underbrace{\sum_{k}^{n_c}\sum_{i}^{m} \tilde{w}_{ki}^2}_{a} - \frac{1}{\lambda}\sum_{l}^{n_p}\underbrace{\sum_{k}^{n_c}\left(\sum_{i}^{m} \tilde{w}_{ki} z_{li}\right)^2}_{b_{ll}} \delta_l$$

In the case of a Taylor approximation of order 2, we have:

$$\begin{aligned}
D(X) \approx D_2(X) &= \sum_{k}^{n_c} \tilde{w}_k \left(I - \frac{X^T X}{\lambda} + \frac{(X^T X)^2}{\lambda^2}\right) \tilde{w}_k^T \\
&= \sum_{k}^{n_c}\sum_{i}^{m} \tilde{w}_{ki}^2 - \frac{1}{\lambda}\sum_{k}^{n_c}\sum_{i}^{m} \tilde{w}_{ki}\sum_{j}^{m} \tilde{w}_{kj}\sum_{l}^{n_r} x_{li} x_{lj} \\
&\quad + \frac{1}{\lambda^2}\sum_{k}^{n_c}\sum_{i}^{m} \tilde{w}_{ki}\sum_{j}^{m} \tilde{w}_{kj}\sum_{h}^{m}\Big(\sum_{l}^{n_r} x_{li} x_{lh}\Big)\Big(\sum_{l}^{n_r} x_{lh} x_{lj}\Big) \\
&= \sum_{k}^{n_c}\sum_{i}^{m} \tilde{w}_{ki}^2 - \frac{1}{\lambda}\sum_{k}^{n_c}\sum_{i}^{m} \tilde{w}_{ki}\sum_{j}^{m} \tilde{w}_{kj}\sum_{l}^{n_p} \delta_l z_{li} z_{lj} \\
&\quad + \frac{1}{\lambda^2}\sum_{k}^{n_c}\sum_{i}^{m} \tilde{w}_{ki}\sum_{j}^{m} \tilde{w}_{kj}\sum_{h}^{m}\Big(\sum_{l}^{n_p} \delta_l z_{li} z_{lh}\Big)\Big(\sum_{o}^{n_p} \delta_o z_{oh} z_{oj}\Big)
\end{aligned}$$

Finally we reorganize the terms depending on the different combinations of δ_l variables.

$$\begin{aligned}
D_2(X) &= a - \frac{1}{\lambda}\sum_{l}^{n_p} b_{ll}\delta_l + \frac{1}{\lambda^2}\sum_{l}^{n_p}\sum_{o}^{n_p}\left(\sum_{h}^{m} z_{lh} z_{oh}\right)\underbrace{\left[\sum_{k}^{n_c}\left(\sum_{i}^{m} \tilde{w}_{ki} z_{li}\right)\left(\sum_{j}^{m} \tilde{w}_{kj} z_{oj}\right)\right]}_{b_{lo}} \delta_l \delta_o \\
&= a - \frac{1}{\lambda}\sum_{l}^{n_p} b_{ll}\delta_l + \frac{1}{\lambda^2}\sum_{l}^{n_p}\sum_{o}^{n_p}\underbrace{\left(\sum_{h}^{m} z_{lh} z_{oh}\right)}_{c_{lo}} b_{lo}\, \delta_l \delta_o
\end{aligned}$$

To conclude we are going to minimize a quadratic objective function with $n_p(1+n_p)$ terms, n_p Boolean variables ($\delta_l \, \forall l \in \{1, \ldots, n_p\}$), and an additional linear *cardinality constraint* $\sum_l^{n_p} \delta_l = n_r$. Note that the time for computing the objective function coefficients is already $O(n_p^2 n_c m)$. Depending on the size of this minimization problem, it can be solved by complete search methods (*e.g.,* best-first or depth-first Branch and Bound) or by local search methods (*e.g.,* simulated annealing or Tabu search) in the framework of integer linear/quadratic/constraint programming and weighted Max-SAT/CSP.

3 Integer linear/quadratic/constraint programming models

We add n_p^2 extra variables γ_{lo} in order to linearize the quadratic objective function. For every pair of Boolean variables (δ_l, δ_o), there is a Boolean variable γ_{lo} that is equal to 1 iff $\delta_l = \delta_o = 1$. We have the following 0/1 linear programming (01LP) formulation:

$$\min \sum_{l}^{n_p} \sum_{o}^{n_p} c_{lo}\gamma_{lo} - \lambda \sum_{l}^{n_p} b_{ll}\delta_l$$

$$\begin{aligned} \text{s.t. } & \sum_{l}^{n_p} \delta_l = n_r \\ & \delta_l + \delta_o \leq 1 + \gamma_{lo} \quad (\forall l \in \{1,\ldots,n_p\}, o \in \{1,\ldots,n_p\}) \\ & \gamma_{lo} \leq \delta_l \quad (\forall l \in \{1,\ldots,n_p\}, o \in \{1,\ldots,n_p\}) \\ & \gamma_{lo} \leq \delta_o \quad (\forall l \in \{1,\ldots,n_p\}, o \in \{1,\ldots,n_p\}) \end{aligned}$$

By removing the last three inequations and replacing γ_{lo} by $\delta_l * \delta_o$, we get a 0/1 quadratic programming (01QP) formulation. The same 01QP formulation can be used by constraint programming (CP) languages such as MiniZinc [6]. By removing the cardinality constraint, we get a pure boolean quadratic optimization (BQO) formulation.

4 Weighted CSP and weighted Max-SAT models

A Weighted Constraint Satisfaction Problem (WCSP) [7] P is a triplet $P = (X, F, k)$ where X is a set of variables and F a set of cost functions. Each variable $x \in X$ has a finite domain of values that can be assigned to it. A cost function $f(S) \in F$, with scope S a sequence of distinct variables of X, is a function which associates to every assignment t of its variables a positive integer in $[0, k]$ where k is a maximum integer cost used for representing forbidden assignments.

The Weighted Constraint Satisfaction Problem is to find a complete assignment t minimizing the total cost $\mathcal{W} = \sum_{f(S)\in F} f(t[S])$ where $t[S]$ denotes the projection of t over variables S. This optimization problem has an associated NP-complete decision problem.

The genomic selection cost minimization problem has $X = \{\delta_1, \ldots, \delta_{n_p}, x_1, \ldots, x_{n_p+1}\}$, all δ_l (resp. x_l) domains are equal to $\{0{,}1\}$ (resp. $[0, n_r]$), $F = \{f(\delta_l)\, \forall l \in \{1, \ldots, n_p\}\} \cup \{f(\delta_l, \delta_o)\, \forall l \times o \in \{1, \ldots, n_p\}^2, l \neq o\} \cup \{f(x_1), f(x_{n_p+1})\} \cup \{f(x_l, \delta_l, x_{l+1})\, \forall l \in \{1, \ldots, n_p\}\}$, and $k = +\infty$.

We define:

$$\forall l \in \{1, \ldots, n_p\} \quad f(\delta_l) = \lfloor 0.5 + M(\lambda b_{ll}(1 - \delta_l) + c_{ll}\delta_l) \rfloor \text{ if } c_{ll} \geq 0$$

$$\begin{array}{rcl}
& = & \lfloor 0.5 + M(\lambda b_{ll} - c_{ll})(1-\delta_l) \rfloor \quad \text{if } c_{ll} < 0 \\
\forall l \times o \in \{1,\ldots,n_p\}^2, l \neq o \quad f(\delta_l, \delta_o) & = & \lfloor 0.5 + M c_{lo} \delta_l \delta_o \rfloor \quad \text{if } c_{lo} \geq 0 \\
& = & \lfloor 0.5 + M c_{lo}(\delta_l \delta_o - 1) \rfloor \quad \text{if } c_{lo} < 0 \\
f(x_1) & = & 0 \quad \text{if } x_1 = 0 \\
f(x_1) & = & k \quad \text{if } x_1 \neq 0 \\
f(x_{n_p+1}) & = & 0 \quad \text{if } x_{n_p+1} = n_r \\
f(x_{n_p+1}) & = & k \quad \text{if } x_{n_p+1} \neq n_r \\
\forall l \in \{1,\ldots,n_p\} \quad f(x_l, \delta_l, x_{l+1}) & = & 0 \quad \text{if } x_l + \delta_l = x_{l+1} \\
& = & k \quad \text{if } x_l + \delta_l \neq x_{l+1}
\end{array}$$

with M a large value used to convert real numbers into integers (rounding to the nearest integer). We have $\mathcal{W} \simeq D_2(X) + C$, where C is a positive constant shift value used in order to keep all cost functions positive. Cost functions $f(x_l, \delta_l, x_{l+1})$ are used to decompose the cardinality constraint $\sum_l^{n_p} \delta_l = n_r$ into an equivalent set of low arity cost functions, by introducing extra counting variables $\{x_1, \ldots, x_{n_p+1}\}$.

By removing the part for encoding the cardinality constraint, we get a formulation ready for Max-SAT solvers.

5 Preliminary results

5.1 Simulation of genomic data

A population with a linkage disequilibrium (LD) extent comparable to one found in a real sheep population (*Manech Tête Rousse* breed) was simulated with the `QMSim` software [15]. For that, a historical population of $20,000$ individuals was simulated for $1,050$ generations by considering an equal number of individuals from both sexes, discrete generations, random matings, no selection and no migration to create an initial LD, and establish a mutation-drift equilibrium state. For the first $1,000$ generations, the population size was decreased to $2,000$ individuals and then increased to $16,000$ individuals within the last 50 generations to create a bottleneck and eventual decrease in effective population size as known in domestic animals. Furthermore, $15,000$ females and 350 males from the last historical generation were used as founders of the selected population. From the founder population, 10 overlapping generations of selection (with 20% and 30% replacement rate for females and males, respectively) and random mating were simulated as contemporary born animals. For the purpose of this study, females from generations 8 and 9 served as the phenotyped population, *i.e.,* $n_p \leq 20,928$, where to select the reference population, and males from generation 10 were used as the candidate population, *i.e.,* $n_c \leq 10,453$. The simulated genome consisted of $m = 10,000$ SNP markers, equally spaced across 5 chromosomes of 100 cM each and $2.5 * 10^{-5}$ mutation rate per marker.

5.2 Comparison of 01LP, 01QP, 01BQO, CP, Max-SAT, WCSP solvers

We compare the models described in Section 3 and 4, in terms of CPU-time, for solving the Taylor approximation of order 2. We vary the problem size n_p from 20 to 200, and experiment with different ratios $\frac{n_r}{n_p}$ from 0.25 to 0.5. We also compare with an unconstrained model where the cardinality constraint $\sum_l^{n_p} \delta_l = n_r$ has been discarded.

We compare the 01LP solver `SCIP` (version 1.2.0), the 01LP and 01QP solver IBM ILOG `cplex` (version 12.4.0.0), the semidefinite programming based BQO tool `BiqMac` [12], the pseudo-Boolean solvers `clasp` (version 2.0.4) and `SAT4J` (version 2.3.4), the CP solver `mistral` (version 1.3.40), the Max-SAT solvers `minimaxsat` [5] and `maxhs` [3] (both using the *tuple* encoding as described in [2]), all solvers using default options, and the WCSP solver `toulbar2` (version 0.9.6[3]) using default options except an initial limited discrepancy search phase [4] with a maximum discrepancy of 2 (option `-l=2` and no initial upper bound). `SCIP`, `toulbar2`, and `mistral` are accessed via the Python multi-solver modeling interface offered by `NumberJack`[4]. All real value coefficients in the models are multiplied by $M = 0.01$ and rounded to the nearest integer, ensuring completeness of the solvers. We measured the search effort for finding the optimum and proving optimality as reported in Table 1.

For the smallest instances ($n_p \in [20, 100]$), the quadratic programming solver QP/`cplex` and the semidefinite programming based boolean quadratic optimization tool `BiqMac`, used in the unconstrained case only, clearly dominate the other solvers. For the largest instances ($n_p \in \{200\}$), all the approaches failed to solve the problem in less than 10 hours.

In order to solve large problems (up to $n_p = 200$), we use a two-step procedure. First, we apply a local search method, called `ID Walk` for *Intensification / Diversification Walk* [10], available as a library [9][5] integrated in `toulbar2`. Due to its neighborhood structure (changing only one variable assignment per move), `ID Walk` can only be applied to the unconstrained problem. We perform 1 run of `ID Walk` with 10,000 iterations, selecting at random among 200 candidate neighbors. The best solution found by the local search method is then used as a pre-selection of the individuals[6] such that the second step is done by a complete search method (using `SCIP`) to satisfy the cardinality constraint. The resulting two-step procedure is called `ID Walk&SCIP`.

For the smallest instances solved optimally by complete search methods ($n_p \in [20, 100]$), `ID Walk&SCIP` always found the optimum for the unconstrained

[3] `http://mulcyber.toulouse.inra.fr/projects/toulbar2`

[4] `http://numberjack.ucc.ie/` and `http://github.com/eomahony/Numberjack/tree/fzn`.

[5] INCOP version 1.1 `http://www-sop.inria.fr/coprin/neveu/incop/presentation-incop.html`

[6] Either by discarding the remaining unselected individuals if too many individuals have been selected by the local search method, or by fixing the selected individuals if they are less than the required number n_r.

problems. The distance to the optimum increases slightly when the required number n_r is (very) different than the one found for the unconstrained case, *e.g.,* being up to 34% for $n_p = 100, n_r = 50$ as reported in Table 2. The overall time of the two-step procedure is clearly dominated by its second step, *e.g.,* unfinished after 10 hours for $n_p = 200, n_r = 100$, which means that the proposed approach should scale to larger problems only if n_r is close to the optimal unconstrained number of selected individuals.

Table 1. Time in seconds of complete search methods (−: unsolved after 10 hours, N/A: non applicable for `BiqMac`,`minimaxsat`, and `maxhs`, which were applied only in the unconstrained case). For unconstrained instances, the number of selected individuals (n_r) in the optimal solution is given in parentheses.

	`SCIP`	`cplex`	QP/`cplex`	`BiqMac`	`clasp`	`SAT4J`	`mistral`	`minimaxsat`	`maxhs`	`toulbar2`
n_p	$n_r = 25\%$									
20	0.7	0.3	0.02	N/A	0.01	0.7	1.3	N/A	N/A	0.16
40	15.8	6.7	0.51	N/A	$8,680$	–	–	N/A	N/A	48.7
60	942.8	$1,089$	12.3	N/A	–	–	–	N/A	N/A	–
100	–	–	223.2	N/A	–	–	–	N/A	N/A	–
200	–	–	–	N/A	–	–	–	N/A	N/A	–
n_p	$n_r = 50\%$									
20	2.5	1.0	0.02	N/A	0.8	3.4	19.3	N/A	N/A	0.14
40	101.1	22.7	0.65	N/A	–	–	–	N/A	N/A	77.2
60	–	$26,853$	9.3	N/A	–	–	–	N/A	N/A	–
100	–	–	$1,031$	N/A	–	–	–	N/A	N/A	–
200	–	–	–	N/A	–	–	–	N/A	N/A	–
n_p	n_r unconstrained (found $n_r = (9, 15, 21, 25)$ resp. for $n_p = (20, 40, 60, 100)$)									
20	1.9	1.1	0.02	0.86	1.1	5.3	19.8	2.1	14.7	0.04
40	94.5	68.3	1.1	13.7	–	–	–	$4,781$	–	5.0
60	–	$20,249$	22.4	29.8	–	–	–	–	–	$11,062$
100	–	–	348.1	87.8	–	–	–	–	–	–
200	–	–	–	–	–	–	–	–	–	–

6 Conclusion

We have presented an optimization problem occuring in the context of genomic selection design. Finding the optimal reference population can be approximated by a quadratic minimization problem on Boolean variables with a cardinality constraint. Preliminary results showed that only quadratic programming solvers such as `cplex` and the semidefinite programming based boolean quadratic optimization tool `BiqMac`, in the unconstrained case, are able to solve optimally

Table 2. Relative distances between the best solutions found by the local search method ID Walk followed by SCIP post-processing and by a complete search method (QP/cplex). CPU-times in seconds for ID Walk and SCIP are given in parentheses when appropriate.

	ID Walk&SCIP		
	n_r/n_p		
n_p	25%	50%	Unconstr.
20	$0.17\%(0.3 + 0.1)$	$0\%(0.3 + 0.03)$	$0\%(n_r = 9)$
40	$0.32\%(0.6 + 0.39)$	$4.17\%(0.6 + 1.17)$	$0\%(n_r = 15)$
60	$0.59\%(0.9 + 0.64)$	$4.56\%(0.9 + 9.17)$	$0\%(n_r = 21)$
100	$0\%(1.4 + 2.47)$	$34.2\%(1.4 + 18,684)$	$0\%(n_r = 25)$
200	$14.32\%(2.8 + 22,746)$	$55.16\%(2.8 + 36,000)$	$0\%(n_r = 35)$

the Taylor approximation of order 2 for a phenotyped population up to 100 individuals. Also, performances of all the solvers vary based on the tightness of the cardinality constraint. These results are useful to assess the quality of local search methods, which are able to tackle much larger problems. Moreover, we have shown how to combine a local search and a complete method in a simple two-step procedure, while degrading the solution quality when the desired number of selected individuals differs significantly from the local search solution. More experiments remain to be done to better distinguish the quality of the two Taylor approximations, and to analyze the performance of local search methods on realistic datasets ($n_p \approx 10,000$) and the properties of the resulting reference population structures.

References

1. Albrecht, T., Wimmer, V., Auinger, H.J., Erbe, M., Knaak, C.: Genome-based prediction of testcross values in maize. Theor. Appl. Genet. 123, 339–350 (2011)
2. Bacchus, F.: Gac via unit propagation. In: Principles and Practice of Constraint Programming–CP 2007. pp. 133–147. Springer (2007)
3. Davies, J., Bacchus, F.: Solving maxsat by solving a sequence of simpler sat instances. In: Principles and Practice of Constraint Programming–CP 2011, pp. 225–239. Springer (2011)
4. Harvey, W.D., Ginsberg, M.L.: Limited discrepency search. In: Proc. of the 14^{th} IJCAI. Montréal, Canada (1995)
5. Heras, F., Larrosa, J., Oliveras, A.: Minimaxsat: An efficient weighted max-sat solver. J. Artif. Intell. Res.(JAIR) 31, 1–32 (2008)
6. Marriott, K., Nethercote, N., Rafeh, R., Stuckey, P., de la Banda, M.G., Wallace, M.: The design of the zinc modelling language. Constraints 13(3), 229–267 (2008)
7. Meseguer, P., Rossi, F., Schiex, T.: Soft constraints processing. In: Rossi, F., van Beek, P., Walsh, T. (eds.) Handbook of Constraint Programming, chap. 9. Elsevier (2006)
8. Meuwissen, T.H., Hayes, B.J., Goddard, M.E.: Prediction of total genetic value using genome-wide dense marker maps. Genetics 157, 1819–1829 (2001)
9. Neveu, B., Trombettoni, G.: INCOP: An Open Library for INcomplete Combinatorial OPtimization. In: Proc. of CP-03. pp. 909–913. Cork, Ireland (2003)

10. Neveu, B., Trombettoni, G., Glover, F.: Id walk: A candidate list strategy with a simple diversification device. In: CP. pp. 423–437 (2004)
11. Pszczola, M., Strabel, T., Mulder, H., Calus, M.: Reliability of direct genomic values for animals with different relationships within and to the reference population. J. Dairy Sci. 95, 389–400 (2012)
12. Rendl, F., Rinaldi, G., Wiegele, A.: Solving Max-Cut to optimality by intersecting semidefinite and polyhedral relaxations. Math. Programming 121(2), 307 (2010)
13. Rincent, R., Laloë, D., Nicolas, S., Altmann, T., Brunel, D., Revilla, P., Rodríguez, V., Moreno-Gonzalez, J., Melchinger, A., Bauer, E., Schoen, C.C., Meyer, N., Giauffret, C., Bauland, C., Jamin, P., Laborde, J., Monod, H., Flament, P., Charcosset, A., Moreau, L.: Maximizing the reliability of genomic selection by optimizing the calibration set of reference individuals: Comparison of methods in two diverse groups of maize inbreds (zea mays l.). Genetics 192, 715–728 (2012)
14. de Roos, A., Hayes, B., Spelman, R., Goddard, M.: Linkage disequilibrium and persistence of phase in holstein-friesian, jersey and angus cattle. Genetics 179(3), 1503–1512 (2008)
15. Sargolzaei, M., Schenkel, F.S.: Qmsim: a large-scale genome simulator for livestock. Bioinformatics 25, 680–681 (2009)
16. Schaeffer, L.: Strategy for applying genome-wide selection in dairy cattle. J Anim Breed Genet 123(4), 218–223 (2006)
17. Shumbusho, F., Raoul, J., Astruc, J., Palhiere, I., Elsen, J.: Potential benefits of genomic selection on genetic gain of small ruminant breeding programs. J Anim Sci in press (2013)
18. Tribout, T., Larzul, C., Phocas, F.: Efficiency of genomic selection in a purebred pig male line. J Anim. Sci 12, 4164–4176 (2012)
19. West, B.T., Welch, K.B., Galecki, A.T.: Linear mixed models: A practical guide to using statistical software. Chapman & Hall/CRC (2007)