A Mathematical Programming Approach to Marker-Assisted Gene Pyramiding

Stefan Canzar[1,⋆] and Mohammed El-Kebir[1,2,⋆]

[1] Centrum Wiskunde & Informatica, Life Sciences Group,
Science Park 123, 1098 XG Amsterdam, The Netherlands
{s.canzar,m.el-kebir}@cwi.nl
[2] Centre for Integrative Bioinformatics VU (IBIVU), VU University Amsterdam,
De Boelelaan 1081A, 1081 HV Amsterdam, The Netherlands

Abstract. In the *crossing schedule* optimization problem we are given an initial set of parental genotypes and a desired genotype, the ideotype. The task is to schedule crossings of individuals such that the number of generations, the number of crossings, and the required populations size are minimized. We present for the first time a mathematical model for the general problem variant and show that the problem is $\mathcal{NP}$-hard and even hard to approximate. On the positive side, we present a mixed integer programming formulation that exploits the intrinsic combinatorial structure of the problem. We are able to solve a real-world instance to provable optimality in less than 2 seconds, which was not possible with earlier methods.

1 Introduction

Plant breeding is the practice of creating improved varieties of cultivated crops with for instance a higher yield, better appearance or enhanced disease resistance [2]. Up to recently, selection of favorable traits has been solely on the basis of observable *phenotype* [4]. With the availability of *genetic maps*, containing the exact locations on the genome of genetic markers associated with desirable traits, selection at the *genotypic* level has become possible [8]. This knowledge allows to design a schedule of crossings of individuals resulting ultimately in an individual with all alleles corresponding to desired favorable traits present. In the plant breeding literature this process is called *marker-assisted gene-pyramiding* and the resulting plan a *gene-pyramiding scheme* or a *crossing schedule* [3, 10, 14]. In this work we consider a mathematical programming approach to the problem that asks to identify given (1) a genetic map, (2) an initial set of parental genotypes and (3) the desired genotype—the so called *ideotype*—a crossing schedule that results most cost-efficiently in the ideotype with respect to the following three criteria. Firstly, it takes time for the progeny to mature such that a next crossing can be performed. So the *number of generations* is a measure on the time it takes to execute the crossing schedule. Secondly, every crossing between

⋆ Joint first authorship.

T.M. Przytycka and M.-F. Sagot (Eds.): WABI 2011, LNBI 6833, pp. 26–38, 2011.

two individual plants requires an effort from the breeder, e.g. plants have to be treated such that they flower at the same time. So typically the *number of crossings* is also to be minimized. Thirdly, in order to obtain the genotypes required by the schedule, for every crossing a specific number of offspring need to be generated among which the desired genotype is expected to be present. Simply speaking, the more difficult it is to obtain the desired genotype out of its parental genotypes, the larger the required number of offspring will be. Since every individual in the offspring has to be screened for having the desired genotype, the *total population size* is also to be minimized.

Related work. Most work on gene pyramiding lacks a formal framework; instead only an overview of guidelines and rules of thumb is given [6, 14]. A notable exception, however, is the work by Servin et al. [10] who were the first to introduce a special case of the problem considered in this paper in a formal way. The authors show how to make use of the genetic map in determining the population sizes needed for all crossings. Contrary to our formulation, they allow a genotype to only participate in one crossing. In addition, very restrictive assumptions about the genotypes of the initial parents were made. These restrictions allowed the authors to exhaustively enumerate all crossing schedules and compare them in terms of population size needed. By introducing a heuristic, which partially alleviates the restriction on re-use of genotypes, the authors could compute smaller population sizes for the instances considered. Later papers by Ishii and Yonezawa [6] assume that target genes are always unlinked, which imposes a lower bound on the genetic distance of pairs of target genes. Similar to our work, in [6] the number of generations, number of crossings and the total population size are identified as important attributes. An experimental evaluation is performed on manually obtained crossing schedules having different topologies for a fixed number of parents.

Our contribution. In this work we lift the restrictions imposed by Servin et al. and consider a more general variant of the problem where genotypes are allowed to be re-used and no assumption about the initial parental genotypes is made. For the first time we formulate a mathematical model of the general problem. We show NP-hardness using an approximation-factor preserving reduction from an inapproximability result follows. We introduce a mixed integer linear program (MIP) formulation which exploits various aspects of the inherent combinatorial structure of the problem and which approximates the non-linear objective by a piecewise linear curve. Finally, we show that our approach is capable of solving real-world instances to provable optimality within a precise mathematical model, which was not possible with earlier methods. The rest of the paper is organized as follows. We start by formally defining the problem and showing hardness of the problem. In Section 3 we introduce our method and state a MIP formulation. An experimental evaluation on a real-word instance and on randomly generated instances is presented in Section 4. We conclude with a discussion on our results in Section 5. Due to the lack of space, we omit the proofs of the given lemmas.

2 Problem Definition and Complexity

A *genotype* C is a $2 \times m$ matrix whose elements are called *alleles*. The two rows, $C_{1,\cdot}$ and $C_{2,\cdot}$, are called the lower and upper *chromosome*, respectively. Each column in C corresponds to a *locus*. So at a locus p two alleles are present, which we denote by $c_{1,p}$ and $c_{2,p}$. A locus is said to be *homozygous* if its two alleles are identical, otherwise it is *heterozygous*. Likewise, a genotype is homozygous if all its loci are homozygous, otherwise the genotype is said to be heterozygous. The desired genotype is called the *ideotype*, which we denote by C^*. In plant breeding often pure lines are desired, as they allow for instance for the production of F1 hybrids [2]. Therefore for the remainder of the paper we assume the ideotype to be homozygous. In this case, actual alleles can be classified as being present in the ideotype or not. Hence, the alleles in any genotype C are binary.

We represent a *crossing schedule* as a *connected directed acyclic graph* (DAG) whose nodes are labeled by genotypes. Specifically, the source nodes correspond to the initial parental genotypes. A non-source node, which we refer to as an *inner node*, corresponds to a crossing. The single target node is labeled by the ideotype. The arcs are directed towards the ideotype and relate a parent with its child. Since a genotype is obtained from two parents, the in-degree of an inner node is exactly 2. The two parents of a node need not be distinct. We say that a genotype is obtained via *selfing* if its two parents are identical. From the topology of a crossing schedule the number of generations and the number of crossings can be inferred. The number of generations is the length of the longest path from a source node to the target node. On the other hand, the number of crossings corresponds to the number of inner nodes. In Figure 1 an example crossing schedule is given.

The third attribute of a crossing schedule, the *total population size*, is the sum of the population sizes implied by the crossings represented by inner nodes. Let C be the genotype of an inner node and let D and E be the genotypes of the two parents of C. Later, we will show what the probability $\Pr[\, D, E \to C \,]$ of obtaining C out of D and E is. For now we denote this probability with ρ. The population size $N(\rho, \gamma)$ corresponding to ρ is the number of offspring one needs to generate in order to find with a given *probability of success* γ an individual with genotype C among the offspring. Since ρ is the probability of success in a Bernoulli trial, the probability that none of the $N(\rho, \gamma)$ offspring have genotype C is $(1-\rho)^{N(\rho,\gamma)} = 1 - \gamma$. Therefore we have that

$$N(\rho, \gamma) = \frac{\log(1-\gamma)}{\log(1-\rho)}. \tag{1}$$

As also remarked in [10], it is sensible to have an upper bound on every population size in the schedule, as depending on the plant species only a limited number of offspring can be generated. For that purpose we define $N_{\max}$ to be the *maximal population size* to which every crossing in a crossing schedule has to adhere.

In diploid organisms, the genotype of a zygote is obtained by the fusion of two haploid gametes originating from one parent each. So one of the chromosomes of

the resulting genotype C, say $C_{1,\cdot}$, corresponds to a gamete given rise to by D and the other chromosome corresponds to a gamete produced by E. A gamete is the result of a biological process called *meiosis* where in pairs of homologous chromosomes crossover events may occur. In our setting, this means that an allele $c_{1,p}$ corresponds to either $d_{1,p}$ or $d_{2,p}$ (where $1 \leq p \leq m$). In case a pair of alleles at loci p and q of $C_{1,\cdot}$ do not correspond to the same chromosome of D, we say that a *crossover* has occurred between loci p and q (see Figure 1). From the genetic map, the probability of having a crossover between any pair of loci can be inferred using for instance Haldane's mapping function [5]. Let R be a $m \times m$ matrix containing all crossover probabilities. Due to the nature of meiosis, we have that $r_{p,q} \leq 0.5$ for $1 \leq p < q \leq m$. Let $s = (\nu(1), \ldots, \nu(k))$ be an ordered sequence of heterozygous loci in D. The probability of obtaining $C_{1,\cdot}$ out of D, i.e. $\Pr[\, D \to C_{1,\cdot}\,]$, is then as follows [10]. If there is an allele in $C_{1,\cdot}$ that does not occur in D at the same locus then $\Pr[\, D \to C_{1,\cdot}\,] = 0$. Otherwise, if s is empty then $\Pr[\, D \to C_{1,\cdot}\,] = 1$. Otherwise

$$\Pr[\, D \to C_{1,\cdot}\,] = \frac{1}{2} \prod_{i=1}^{k-1} \begin{cases} r_{\nu(i),\nu(i+1)} & \text{if } c_{1,\nu(i)} = d_{1,\nu(i)} \wedge c_{1,\nu(i+1)} = d_{2,\nu(i+1)} \\ & \text{or } c_{1,\nu(i)} = d_{2,\nu(i)} \wedge c_{1,\nu(i+1)} = d_{1,\nu(i+1)} \\ 1 - r_{\nu(i),\nu(i+1)} & \text{otherwise.} \end{cases} \tag{2}$$

We can now compute $\Pr[\, D, E \to C\,]$ using the following lemma.

Lemma 1. *The probability of obtaining C out of genotypes D and E is*

$$\Pr[\, D, E \to C\,] = \begin{cases} \Pr[\, D \to C_{1,\cdot}\,] \cdot \Pr[\, E \to C_{2,\cdot}\,] & \text{if } C_{1,\cdot} = C_{2,\cdot} \\ \Pr[\, D \to C_{1,\cdot}\,] \cdot \Pr[\, E \to C_{2,\cdot}\,] & \\ \quad + \Pr[\, E \to C_{1,\cdot}\,] \cdot \Pr[\, D \to C_{2,\cdot}\,] & \text{if } C_{1,\cdot} \neq C_{2,\cdot} \end{cases} \tag{3}$$

A common way to deal with multiple objectives is to consider a convex combination of the objective criteria involved [12]. Given a crossing schedule, let crs, gen and pop denote the number of crossings, number of generations and the total population size, respectively. For $\lambda_{\text{crs}}, \lambda_{\text{gen}}, \lambda_{\text{pop}} \geq 0$ and $\lambda_{\text{crs}} + \lambda_{\text{gen}} + \lambda_{\text{pop}} = 1$, the *cost* of that crossing schedule is given by the convex combination $\lambda_{\text{crs}} \cdot \text{crs} + \lambda_{\text{gen}} \cdot \text{gen} + \lambda_{\text{pop}} \cdot \text{pop}$.

*Problem 1 (*CrossingSchedule*).* Given $\mathcal{P} = \{C^1, \ldots, C^n\}$, the set of parental genotypes we start with, the homozygous ideotype $C^* \notin \mathcal{P}$, the recombination matrix R, the desired probability of success $\gamma \in (0, 1)$, the maximal population size $N_{\max} \in \mathbb{N}$ allowed per crossing, and a vector λ of the cost efficients, problem CrossingSchedule asks for a crossing schedule of minimum cost.

We propose a polynomial-time reduction from the decision problem SetCover [7]: the loci correspond to the elements in the universe and the initial set of parents to the family of subsets. The first chromosome of a parent C^i has a 1 at locus p if p is contained in the corresponding subset. The second chromosomes of all parental genotypes consists of only zeros. The ideotype has 1 alleles at every locus. In the cost function we only consider the number of crossings, i.e. $\lambda_{\text{crs}} = 1$ and $\lambda_{\text{gen}} = \lambda_{\text{pop}} = 0$.

Theorem 1. CrossingSchedule *is NP-hard.*

Due to the approximation-factor preserving reduction, the inapproximability result for SetCover [9] carries over:

Theorem 2. *Approximating* CrossingSchedule *within* $\mathcal{O}(\log n)$ *is NP-hard.*

3 Method

After exploring the combinatorial structure of the problem, we present an algorithm in which iteratively an MIP is solved. Details on the MIP formulation are given in Section 3.1.

Since we are considering homozygous ideotypes, we can assume without loss of generality that C^* has only 1-alleles and derive a lower bound based on the minimum set cover as follows. The universe corresponds to the loci, i.e. $U = \{1, \ldots, m\}$, and the subsets $\mathcal{S} = \{S_1, \ldots, S_n\}$ correspond to $\mathcal{P} = \{C^1, \ldots, C^n\}$. We define $p \in S_i$ if either $c^i_{1,p} = 1$ or $c^i_{2,p} = 1$ where $1 \leq i \leq n$ and $1 \leq p \leq m$. The following lemma now follows.

Lemma 2. *The cardinality of a minimum set cover is a lower bound on the number of crossings of any feasible crossing schedule*

Computing the minimum set cover is NP-hard. However, since in our experiments the number of loci and parents are relatively small, we are able to obtain the lower bound by solving a corresponding ILP [13] in a fraction of a second.

A lower bound on the population size can be obtained when considering the set $\mathcal{L}$ of all pairs of consecutive loci for which there are no genotypes in $\mathcal{P}$ containing 1-alleles at the respective loci on the same chromosome:

Lemma 3. *The following is a lower bound on the total population size.*

$$LB_{\text{pop}} = \sum_{(p,p+1) \in \mathcal{L}} N(r_{p,p+1}, \gamma) \tag{4}$$

Using (3) one can show that there is an optimal crossing schedules where homozygous genotypes are obtained via selfings.

Lemma 4. *There is an optimal schedule in which the (inner) homozygous genotypes are obtained via selfings.*

Finally, parental genotypes that contain a 1-allele at a locus at which all other parental genotypes contain all 0 have to be used by any feasible schedule. To reduce the search space explored by the MIP solver we fix these *compulsory* parental genotypes to be contained in any solution.

We present a MIP formulation for the problem variant where the number of crossings and the number of generations is fixed to F, respectively G. The reason for this is to be able to introduce cuts that ensure monotonically better solutions. In order to solve a problem instance, we iteratively consider combinations of (F, G) starting from $F = LB_{\text{crs}}$ and $G = 1 + \lceil \log_2 F \rceil$. In addition we enforce that

the objective value of any feasible solution must be better than the currently best one. We do this by computing an upper bound UB_{pop} on the total population size, based on the best objective value found so far and the current values of (F, G) (see Algorithm 1, line 4). If at some point, say (F', G'), $LB_{\text{pop}} \geq UB_{\text{pop}}$ then we know that none of the combinations of $F'' \geq F'$, $G'' \geq G'$ will lead to a better solution. Therefore if $G = 1 + \lceil \log_2 F \rceil$ and $LB_{\text{pop}} \geq UB_{\text{pop}}$, we have found the optimal solution (see Algorithm 1, line 7). To guarantee termination for the case where $\lambda_{\text{crs}} = \lambda_{\text{gen}} = 0$, we stop incrementing F as soon as it reaches a pre-specified parameter UB_{crs}. Similarly, UB_{gen} is a pre-specified parameter bounding G. In Algorithm 1 the pseudo code is given.

Algorithm 1. OPTCROSSINGSCHEDULE($UB_{\text{crs}}, UB_{\text{gen}}$)

Input: UB_{crs} and UB_{gen} are the maximum number of crossings and generations considered.

1 OPT $\leftarrow \infty$
2 **for** $F \leftarrow LB_{\text{crs}}$ **to** UB_{crs} **do**
3 **for** $G \leftarrow 1 + \lceil \log_2 F \rceil$ **to** $\min(F, UB_{\text{gen}})$ **do**
4 $UB_{\text{pop}} \leftarrow \frac{1}{\lambda_{\text{pop}}}(\text{OPT} - F \cdot \lambda_{crs} - G \cdot \lambda_{gen})$
5 **if** $LB_{\text{pop}} < UB_{\text{pop}}$ **then** OPT $\leftarrow \min(\text{OPT}, \text{MIP}(F, G, UB_{\text{pop}}))$
6 **else** $UB_{\text{gen}} \leftarrow G - 1$
7 **if** $UB_{\text{gen}} \leq 1 + \lceil \log_2 F \rceil$ **then** **return** OPT
8 **return** OPT

3.1 MIP Formulation

Given an instance to CROSSINGSCHEDULE with initial parental genotypes $\mathcal{P} = \{C^1, \ldots, C^n\}$, a feasible solution with G generations and F crossings can be characterized by the following five conditions: (i) The topology of the schedule is represented by a DAG with n source nodes $s_1, \ldots, s_n$, one target node t, and $F - 1$ additional nodes, where every non-source node has in-degree two. Parallel arcs are allowed and represent selfings. (ii) The longest path from a source node to the target node has length G. (iii) The alleles of each non-source node are derived from either the upper or lower chromosome of the node's respective predecessors. (iv) The genotype of a source node s_i is C^i, the genotype of t is C^*. (v) The probability of obtaining the genotype of an inner node v is at least $1-(1-\gamma)^{\frac{1}{N_{\max}}}$ such that its corresponding population size is at most $N_{\max}$. In the following we show how these conditions can be formulated as linear constraints. Throughout our formulation, we let $L := F + n$ be the total number of nodes. Dummies $1 \leq i, j \leq L$ correspond to genotypes, loci are indexed by $1 \leq p, q \leq m$ and chromosomes are referred to by $1 \leq k, l \leq 2L$. In the remainder of the paper we will omit the linearization of products of binary variables. Unless otherwise stated, we applied a standard transformation [1]. Similarly, we omit the details of the implementation of absolute differences of binary variables.

Feasibility constraints. The first set of constraints encodes the structure of the underlying DAG $D = (V, A)$. We assume a numbering of the vertices according to their topological order. In particular, arcs always go from vertices $j < i$ to a vertex i, $i, j \in V$. Based on the node numbering, the lower and upper chromosomes of a node $i \in V$ are respectively $2i - 1$ and $2i$. For convenience we introduce a mapping function $\delta(k)$ that returns the node a chromosome k corresponds to. Then binary variables $x_{k,i} \in \{0, 1\}$, $2n < k \leq 2L$, $i < \delta(k)$, denote whether chromosome k originates from genotype i, that is, they indicate an arc $(i, \delta(k))$. Since a chromosome originates from exactly one genotype, we have

$$\sum_{j=1}^{\delta(k)-1} x_{k,j} = 1 \qquad 2n < k \leq 2L \tag{5}$$

We capture the second condition by fixing a path of length G using the x variables and by restricting the depth of all remaining nodes, represented by additional integer variables, to be at most $G-1$. To model the third condition, we introduce binary variables $a_{k,p}$, $1 \leq k \leq 2L$, $1 \leq p \leq m$, which indicate the allele at locus p of chromosome k. Note that for chromosomes k corresponding to initial parental genotypes, $a_{k,p}$, $1 \leq p \leq m$, is a constant rather than a variable. In addition to knowing from which genotype a chromosome originates, we also need to know from which of the two chromosomes of that parental genotype an allele comes. Therefore we define binary variable $y_{k,p}$, $2n < k \leq 2L$, $1 \leq p \leq m$, to be 1 if the allele at locus p of chromosome k comes from the lower chromosome of its originating genotype; conversely $y_{k,p}$ is 0 if the allele originates from the upper chromosome. Now we can relate alleles to originating chromosomes. We do this by introducing binary variables $g_{k,p,l}$, for $2n < k \leq 2L$, $1 \leq p \leq m$, and $1 \leq l < 2\delta(k) - 1$. We define $g_{k,p,l} = 1$ if and only if the allele at locus p of chromosome k originates from chromosome l and has value 1. This is established through constraints

$$g_{k,p,2i} - a_{2i,p} \cdot x_{k,i} \cdot (1 - y_{k,p}) = 0 \qquad 2n < k \leq 2L, 1 \leq p \leq m, i < \delta(k) \tag{6}$$

$$g_{k,p,2i-1} - a_{2i-1,p} \cdot x_{k,i} \cdot y_{k,p} = 0 \qquad 2n < k \leq 2L, 1 \leq p \leq m, i < \delta(k) \tag{7}$$

Finally, an allele is 1 if and only if it originates from exactly one 1-allele:

$$a_{k,p} - \sum_{i=1}^{\delta(k)-1} (g_{k,p,2i-1} + g_{k,p,2i}) = 0 \qquad 2n < k \leq 2L, 1 \leq p \leq m \tag{8}$$

The fourth property can be ensured by simply forcing the variables representing the alleles of the parental genotypes and the alleles of the desired ideotype to the actual value of the respective allele. Thus for the parental genotypes we have $a_{2i-1,p} = c^i_{1,p}$ and $a_{2i,p} = c^i_{2,p}$ for $1 \leq i \leq n, 1 \leq p \leq m$ and for the ideotype $a_{2L-1,p} = a_{2L,p} = c^*_{1,p}$ for $1 \leq p \leq m$. The fifth property is enforced implicitly by the objective function.

Objective function. The probability of a given genotype i giving rise to a specific chromosome k determines the required population size (see (1)). This probability in turn depends on the exact set of crossovers necessary to generate chromosome k and on the sequence s of heterozygous loci (see (2)). Binary variable $\tilde{a}_{i,p} = 1$ if and only if locus p of genotype i is heterozygous: $\tilde{a}_{i,p} = |a_{2i-1,p} - a_{2i,p}|$ for $1 \leq i \leq L, 1 \leq p \leq m$. Now a genotype i is heterozygous, indicated by $h_i = 1$, if at least one of its loci is heterozygous: $h_i \geq \tilde{a}_{i,p}$ for $1 \leq i \leq L, 1 \leq p \leq m$. It is ensured that $h_i = 0$ whenever $\tilde{a}_{i,p} = 0$, $\forall 1 \leq p \leq m$, as $h_i = 1$ would increase the required population size. The distinction between the two different cases in (2) is based on crossover events between two successive heterozygous loci, i.e. $\nu(i)$ and $\nu(i+1)$. We capture the sequence s of heterozygous loci used in (2) by binary variables $b_{i,p,q}$, which indicate a maximal block of homozygous loci between heterozygous loci p and q in genotype i:

$$b_{i,p,q} = \tilde{a}_{i,p} \cdot \tilde{a}_{i,q} \cdot \prod_{r=p+1}^{q-1} (1 - \tilde{a}_{i,r}) \qquad 1 \leq i \leq L, 1 \leq p < q \leq m \tag{9}$$

To formulate the probability given in (2), let ξ_k^j denote the event of obtaining a chromosome k from a genotype j. Using variables h, b, and z, we can express $\Pr[\,\xi_k^j\,]$ such that in the heterozygous case every maximal homozygous block contributes $r_{p,q}$ if it contains at least one crossover, and $(1 - r_{p,q})$ otherwise. Finally, if j_1 and j_2 are the two parental genotypes of chromosomes k_1 and k_2 forming genotype i, we compute in variable $\bar{z}_i$ the log probability of event $\xi_{k_1}^{j_1} \cap \xi_{k_2}^{j_2}$ as $\ln(\Pr[\,\xi_{k_1}^{j_1}\,]) + \ln(\Pr[\,\xi_{k_2}^{j_2}\,])$. For that we have to sum over all possible $j < i$ to identify j_1 and j_2:

$$\begin{aligned} \bar{z}_i = \sum_{j<i} \sum_{l \in \{1,2\}} x_{k_l,j} \Bigg(& h_j \ln(\frac{1}{2}) + \sum_{p=1}^{m-1} \sum_{q=p+1}^{m} b_{j,p,q} \ln(1 - r_{p,q}) \\ & + \sum_{p=1}^{m-1} \sum_{q=p+1}^{m} \sum_{r=p+1}^{q} b_{j,p,q} \cdot \ln(\frac{r_{p,q}}{1 - r_{p,q}}) \cdot |y_{k,r} - y_{k,r-1}| \Bigg) \end{aligned} \tag{10}$$

Notice that we neglect the possibility that the two chromosomes k_1 and k_2 may swap their originating genotypes as accounted for in the second case of equation (3) and we therefore might overestimate the population size. We will discuss this simplification in Section 5. Finally, we develop an approximation of the nonlinear function $N(\rho, \gamma)$ defining the required population size so that LP techniques can be utilized. More precisely, we reduce $N(\rho, \gamma)$ to a separable form [12] that depends only on a single decision variable and approximate it according to the λ-*method* [12] by a piecewise-linear curve specified by the points $(a_j, N(\mathrm{e}^{a_j}, \gamma))$ for $j = 1, \ldots, \ell + 1$. We replace the populations size $N(\mathrm{e}^{\bar{z}_i}, \gamma)$ for each crossing i in the objective function by a convex combination of the respective breakpoint scores to derive $\lambda_{\text{pop}} \cdot \left(\sum_{i=n+1}^{L} \sum_{j=1}^{\ell+1} \lambda_j^i \cdot N(\mathrm{e}^{a_j}, \gamma) \right) + \lambda_{\text{gen}} \cdot G + \lambda_{\text{crs}} \cdot F$.

Additional cuts. We consider three additional cuts. The first one is due to Lemma 4. The following constraints enforce that a homozygous genotype re-

sults via selfing: $|x_{2i-1,j} - x_{2i,j}| \leq h_j$ for $n < i \leq L, 1 \leq j < i$. In addition, the lower and upper bound on the population size correspond to $LB_{\text{pop}} \leq \sum_{i=n+1}^{L} \sum_{j=1}^{\ell+1} \lambda_j^i \cdot N(\mathrm{e}^{a_j}, \gamma) \leq UB_{\text{pop}}$ for $n < i \leq L$. For the sake of simplicity we omit the additional constraints required to enforce compulsory parental genotypes to be contained in the solution. To come back to condition five of our characterization of feasible solutions in the beginning of this section, we simply set $a_1 = \log(1 - (1-\gamma)^{\frac{1}{N_{\max}}})$. Then any $\bar{z}_i < a_1$ implying a population size larger than $N_{\max}$ cannot be expressed as a convex combination of break points a_j, $j = 1, \ldots, \ell + 1$, and hence any feasible solution must satisfy the bound on the population size. In total, our MIP formulation comprises $\mathcal{O}(L(Lm^2 + \ell))$ many variables and $\mathcal{O}(L^2 m)$ constraints.

4 Experimental Results

We have implemented OptCrossingSchedule in C++ using CPLEX 12.2[1] (default settings) with Concert Technology. We ran the experiments on a compute cluster with 2.26 GHz processors with 24 GB of RAM, running 64 bit Linux. We applied a time limit of 10 hours. Computations exceeding this limit were aborted. As mentioned earlier, there exist no previous methods for the general problem formulation we are considering. However, our problem formulation subsumes the one given by Servin et al., therefore we consider the same instances as well. In addition, we study a real-world instance. We conclude by evaluating our method on automatically generated instances. Throughout this section, the term 'provably optimal solution' indicates that the objective value of any feasible solution with respect to the piecewise-linear approximation and the simplification of (3) is at most the objective value of the obtained solution.

Instances by Servin et al. As opposed to our setting, in [10] a crossing schedule is required to be a tree. In addition, the number of initial parental genotypes $\mathcal{P} = \{C^0, C^1, \ldots, C^m\}$ is one more than the number of loci m. Parental genotypes are assumed to be homozygous. More specifically, C^0 consists of only 0-alleles, whereas for a genotype C^i, $1 \leq i \leq m$, the only 1-alleles are present at locus i. The ideotype is comprised entirely of 1-alleles and only the population size is considered, i.e. $\lambda_{\text{pop}} = 1, \lambda_{\text{gen}} = \lambda_{\text{crs}} = 0$. The desired probability of success is $\gamma = 0.999$ and the genetic distance between pairs of consecutive loci is 20 centimorgans (cM). By including constraints forcing a crossing schedule to be a tree (i.e. the out-degree of a node is forced to be 1), we were obtained the same optimal results (see Table 1). Servin et al. realize that better crossing schedules can be obtained when dropping the tree restriction. Rather than considering general DAGs, the authors consider a heuristic (PWC2) that transforms every enumerated tree into a DAG with smaller total population size. As opposed to the tree case, our method does not guarantee the solutions found in the DAG case to be optimal. This is because the objective function does neither include the number of crossings nor the number of generations. In addition, we put a

[1] http://www.cplex.com

Table 1. Results for instances by Servin et al. First column are the results on the tree cases (as obtained by Servin et al's method and our MIP), second column corresponds to PWC2 heuristic and the last column to our MIP for DAGs.

#loci	tree			PWC2			MIP		
	pop	crs	gen	pop	crs	gen	pop	crs	gen
4	374	5	5	359	7	5	350	5	5
5	551	6	6	516	8	6	482	9	8
6	770	7	7	691	9	6	624	9	7
7	1046	8	8	890	13	7	901	10	9
8	1394	9	9	1147	15	7	1329	10	10

time limit of 10 hours in place. In Table 1 we can see that we obtain better solutions w.r.t. the population size for the instances up to six loci. Due to the time limit, the best *feasible* solutions found for the instances with 7 and 8 loci are worse than the ones computed by Servin et al. Since PWC2 solutions are also feasible to our general model, a higher time limit would result in solutions that are at least as good as Servin's. We expect our approach to be less competitive with PWC2 on larger instances of this specific class. This comes at no surprise since PWC2 is specifically tailored toward these restricted instances.

Real-world instance. We consider a real-world case that deals with a disease in pepper called powdery mildew. This disease is caused by the fungus *Leveillula Taurica*. In severe cases of the disease the infected pepper plant may lose a significant amount of its leaves, which in turn results in crop loss. The fungus is resistant to fungicides, so host-plant resistance is desired. There is a wild-type pepper line that is resistant to the fungus. For this wild-type, three dominant quantitative trait loci (QTLs), numbered 1,2 and 3, that explain the resistance have been identified [11]. In addition to resistance, we also look at pungency, which is a dominant monogenic trait whose locus we assign number 4. The pungency gene is closely linked with one of the resistance QTLs, say the one of locus 3, with a genetic distance of 0.01 cM, i.e. $r_{3,4} = 0.01$ [5]. The resistant line is pungent. On the other hand, the elite line used for production is sweet but susceptible to the disease. Both lines are pure lines, i.e. they are homozygous at all loci. The goal now is to come up with a crossing schedule that results in a homozygous individual that is both resistant and sweet. We do this by using 1-alleles to indicate desired alleles. Therefore the parent set is $\mathcal{P} = \left\{ \left(\begin{smallmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \end{smallmatrix}\right), \left(\begin{smallmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{smallmatrix}\right) \right\}$, and the ideotype is $C^* = \left(\begin{smallmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{smallmatrix}\right)$. Unlinked loci by definition have a crossover probability of 1/2. So except for $r_{3,4}$, $r_{p,q} = 1/2$ for all $1 \leq p < q \leq 4$. We set $N_{\max} = 5000$ and $\gamma = 0.95$. Setting $\lambda_{\text{pop}} = 1/201, \lambda_{\text{gen}} = \lambda_{\text{crs}} = 100/201$ is a good trade off between the three criteria. In a practical setting, the λ-s are to be chosen such that they reflect the actual costs. Since there is a cost associated with the number of crossings and the number of generations, we are able to obtain a provably optimal solution in 1.5 seconds which is depicted (right) in Figure 1. It is important to note that this problem instance cannot be expressed in the restricted framework of Servin et al.[10]: treating the resistance loci as a

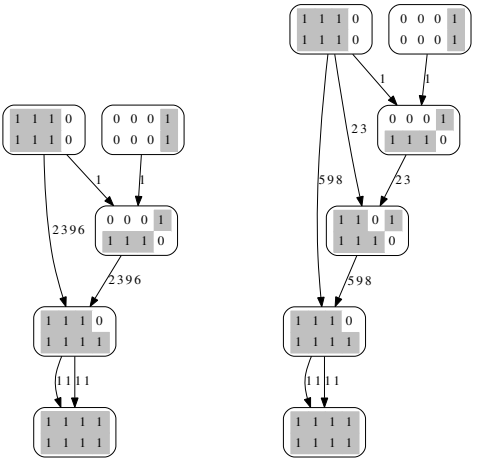

Fig. 1. Crossing schedules for the pepper instance. Inner nodes are obtained via crossings requiring a population size shown on the arcs, in both schedules the final crossing is a selfing. Chromosomes of an inner node are obtained via crossovers in their parents. *Left:* $F = 3$, $G = 3$, pop $= 2408$ and obj $= 14.69$. *Right:* provably optimal, $F = 4$, $G = 4$, pop $= 633$ and obj $= 7.13$.

single locus does not result in the best crossing schedule (see left of Figure 1), as the second genotype is obtained via a crossover between the second and third locus. To the best of our knowledge, such a real-world instance is solved for the first time to provable optimality within a precise mathematical model.

Generated instances. We generate random instances on which we evaluate the performance of our method. The instances either have 5 or 10 parents and concern 4-8 loci. The number of correct alleles per parental genotype affects the difficulty of the instances, we vary this number depending on the number of loci.

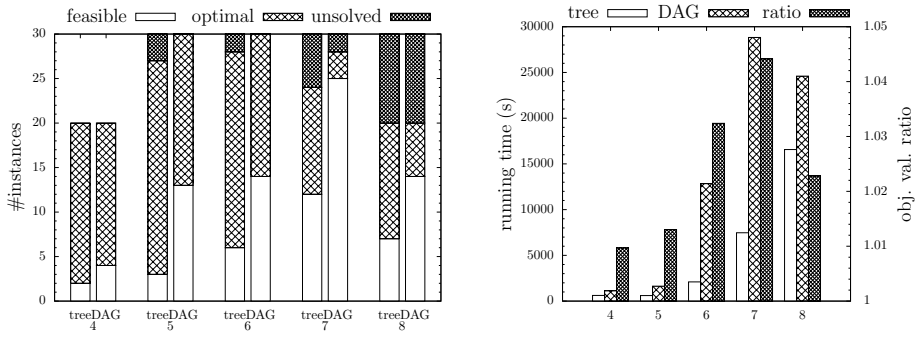

Fig. 2. Results for generated instances. *Left:* optimality of solutions. *Right:* running times; instances exceeding the time limit were not considered, objective value ratio (right y-axis).

In total 140 instances are generated, among which 20 concern instances of 4 loci; the classes of 5-8 loci are comprised by 30 instances each. We run both the DAG and the tree version of the MIP on all instances. For the DAG case, we were able to obtain solutions to 128 instances compared to 119 instances (see Figure 2) for the tree version. Among the unsolved instances for the tree case, there are also instances that are infeasible due to the value of $N_{\max}$ which requires re-use of genotypes. The number of instances that were solved to provable optimality in the DAG case is 58; for the tree case this number is 89. DAGs provide a gain in solution quality of up to 5% on average compared to the tree. Note that none of the instances is of the nature that is captured by Servin's model. Not surprisingly, trees are easier to solve.

5 Conclusion

For the first time we have described a mathematical model capturing the problem of marker-assisted gene pyramiding to its full extent. We show that our approach is capable of solving a real-world instance and generated instances, often to provable optimality. As mentioned earlier, our method is not exact due to (i) the piecewise-linear approximation of the population size function and (ii) a simplification in (10) of neglecting the possibility that the two chromosomes may swap their originating genotypes. However, in our experiments we have not observed any crossing where this could have happened. The NP-hardness proof involves only the number of crossings; as for the number of generations, the same reduction can be applied. The hardness with respect to the population size remains open. Possible extensions to our problem definition include considering heterozygous ideotypes. This requires an extension to tertiary alleles. Another extension would be to consider so called 'don't care' alleles, which are alleles that are not preserved due to crossover events, and as such do not need to be considered in the probability function.

Acknowledgments. We would like to thank Bertrand Servin for kindly providing us the source code of his method. In addition we are very grateful for the constructive comments of the anonymous referees.

References

1. Bradley, S.P., Hax, A.C., Magnanti, T.L.: Applied Mathematical Programming. Addison-Wesley, Reading (1977)
2. Brown, J., Caligari, P.: Introduction to Plant Breeding. Wiley-Blackwell (2008)
3. Collard, B.C.Y., Mackill, D.J.: Marker-assisted selection: an approach for precision plant breeding in the twenty-first century. Phil. Trans. R. Soc. B 363(1491), 557–572 (2008)
4. Dekkers, J.C.M., Hospital, F.: The use of molecular genetics in the improvement of agricultural populations. Nature Reviews Genetics 3, 22–32 (2002)
5. Haldane, J.B.S.: The combination of linkage values and the calculation of distances between the loci of linked factors. Journal of Genetics 8, 299–309 (1919)

6. Ishii, T., Yonezawa, K.: Optimization of the marker-based procedures for pyramiding genes from multiple donor lines: I. Schedule of crossing between the donor lines. Crop Science 47, 537–546 (2007)
7. Karp, R.M.: Reducibility among combinatorial problems. In: Miller, R.E., Thatcher, J.W. (eds.) Complexity of Computer Computations, pp. 85–103. Plenum Press, New York (1972)
8. Moose, S.P., Mumm, R.H.: Molecular plant breeding as the foundation for 21st century crop improvement. Plant Physiology 147, 969–977 (2008)
9. Raz, R., Safra, S.: A sub-constant error-probability low-degree test, and a sub-constant error-probability PCP characterization of NP. In: Proc. 29th ACM Symp. on Theory of Computing, pp. 475–484 (1997)
10. Servin, B., Martin, O.C., Mézard, M., Hospital, F.: Toward a theory of marker-assisted gene pyramiding. Genetics 168(1), 513–523 (2004)
11. Shifriss, C., Pilowsky, M., Zacks, J.M.: Resistance to Leveillula Taurica mildew (=Oidiopsis taurica) in Capsicum annuum. Phytoparasitica 20(4), 279–283 (1992)
12. Steuer, R.E.: Multiple Criteria Optimization: Theory, Computation and Application. Krieger Pub. Co. (1986)
13. Wolsey, L.A.: Integer programming. Wiley-Interscience series in discrete mathematics and optimization. Wiley, Chichester (1998)
14. Ye, G., Smith, K.F.: Marker-assisted gene pyramiding for inbred line development: Basic principles and practical guidelines. International Journal of Plant Breeding 2(1), 1–10 (2008)