11 Algorithms for Multiplex PCR Primer Set Selection with Amplification Length Constraints

K.M. KONWAR, I.I. MĂNDOIU, A.C. RUSSELL, and A.A. SHVARTSMAN

Department of Computer Science & Engineering, University of Connecticut, Storrs, CT, USA

11.1 INTRODUCTION

Numerous high-throughput genomics assays require rapid and cost-effective amplification of a large number of genomic loci. Most significantly, Single Nucleotide Polymorphism (SNP) genotyping protocols often require the amplification of thousands of SNP loci of interest [13]. Effective amplification can be achieved using the polymerase chain reaction [17] (PCR), which cleverly exploits the DNA replication machinery in a cyclic reaction that creates an exponential number of copies of specific DNA fragments.

In its basic form, PCR requires a pair of short single-stranded DNA sequences called *primers* for each amplification target. More precisely, the two primers must be (perfect or near perfect) reversed Watson-Crick complements of the $3'$ ends of the forward and reverse strands of the double-stranded amplification target (see Figure 11.1). Typically there is significant freedom in selecting the exact ends of an amplification target, i.e., in selecting PCR primers. Consequently, primer selection can be optimized with respect to various criteria affecting reaction efficiency, such as primer length, melting temperature, secondary structure, etc. Since the efficiency of PCR amplification falls off exponentially as the length of the amplification product increases, an important practical requirement is that the distance between the binding sites of the two primers should not exceed a certain threshold (typically around 1000 base pairs).

Multiplex PCR (MP-PCR) is a variant of PCR in which multiple DNA fragments are amplified simultaneously. While MP-PCR is still making use of two oligonucleotide primers to define the boundaries of each amplification fragment, a primer may now

participate in the amplification of multiple targets. A primer set is feasible as long as it contains a pair of primers that amplify each target. Note that MP-PCR amplification products are available only as a mixture and may include unintended products. Nevertheless, this is not limiting the use of MP-PCR in applications such as SNP genotyping, since allelic discrimination methods (typically hybridization based) can be applied directly to complex mixtures of and are not significantly affected by the presence of a small number of undesired amplification products [13].

Much of the previous work on PCR primer selection has focused on single primer pair optimization with respect to the above biochemical criteria. This line of work has resulted in the release of several robust software tools for primer pair selection, the best known of which is the Primer3 package [21]. In the context of multiplex PCR, an important optimization objective is to minimize the total number of primers [4, 18], since reducing the number of primers reduces assay cost, increases amplification efficiency by enabling higher effective concentration of the primers, and minimizes primer cross-hybridization and unintended amplification. Pearson et al. [19] were the first to consider minimizing the number of primers in their *optimal primer cover problem*: given a set of n DNA sequences and an integer ℓ, select a minimum number of ℓ-mers such that each sequence contains at least one selected ℓ-mer. Pearson et al. proved that the primer cover problem is as hard to approximate as set cover (i.e., not approximable within a factor better than $(1 - o(1))O(\log n)$ unless NP $\subseteq$ TIME$(n^{O(\log \log n)})$ [5]), and that the classical greedy set cover algorithm achieves an approximation factor of $O(\log n)$.

The problem formulation in Pearson et al. [19] decouples the selection of forward and reverse primers, and, in particular, cannot explicitly enforce bounds on PCR amplification length. A similar remark applies to problem formulations in recent works on *degenerate* PCR primer selection [15, 23]. Such bounds can be enforced only by conservatively defining the allowable primer binding regions. For example, in order to guarantee a distance of L between the binding sites of the forward and reverse primers amplifying a SNP, one could confine the search to primers binding within $L/2$ nucleotides on each side of the SNP locus. However, since this approach reduces the number of feasible candidate primer pairs by a factor of almost 2,[1] it may lead to significant sub-optimality in the total number of primers needed to amplify all given SNP loci.

Motivated by the requirement of unique PCR amplification in synthesis of spotted microarrays, Fernandes and Skiena [6] introduced an elegant *minimum multi-colored subgraph* formulation for the primer selection problem, in which each candidate primer is represented as a graph node and each two primers that feasibly amplify a desired locus define an edge "colored" by the locus number. Minimizing the number of PCR primers reduces to finding a minimum subset of the nodes inducing edges of all possible colors. Unfortunately, approximating the minimum multi-colored

[1] E.g., assuming that all DNA ℓ-mers can be used as primers, out of the $(L-\ell+1)(L-\ell+2)/2$ pairs of forward and reverse ℓ-mers that can feasibly amplify a SNP locus, only $(L-\ell+1)^2/4$ have both ℓ-mers within $L/2$ bases of this locus.

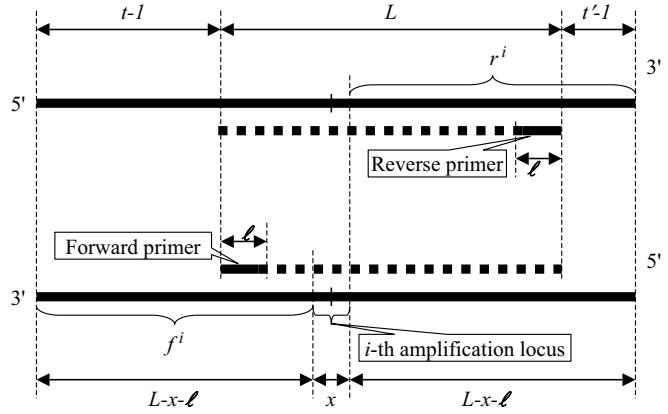


Fig. 11.1 Strings f^i and r^i consist of the $L - x - \ell$ DNA bases immediately preceding in $3' - 5'$ order the i-th amplification locus along the forward (respectively reverse) DNA genomic sequence, where L is the given threshold on PCR amplification length, ℓ is the primer length, and x is the length of an amplification locus ($x = 1$ for SNP genotyping). If forward and reverse PCR primers cover f^i and r^i at positions t and t' respectively, then PCR amplification product length is equal to $[2(L - x - \ell) + x] - [(t - 1) + (t' - 1)]$. This is no larger than L if an only $t + t' \geq L' + 1$, where $L' = (L - x - \ell) - (\ell - 1)$.

subgraph appears to be difficult – the best approximation factor derived via this reduction is currently $O(L \log n)$, where n is the number of amplification loci and L is the upperbound on the PCR amplification length [7].

In this chapter we make the following contributions:

- First, we introduce a new string covering formulation for the MP-PCR primer set selection problem with amplification length constraints that translates into integer programs that are much more compact than those resulting from the minimum multicolored subgraph formulation of Fernandes and Skiena [6]. Our compact integer programs enable computing exact solutions for moderate problem instances using general purpose integer programming solvers such as CPLEX [3].

- Second, we show that a modification of the classical greedy algorithm for the set cover problem achieves an approximation factor of $1 + \ln(\Delta)$, where Δ is the maximum "coverage gain" of a primer. The value of Δ is never more than nL, and in practice it is up to orders of magnitude smaller. The approximation factor is established using a novel framework for analyzing greedy algorithms based on monotonic potential functions. Our potential function technique generalizes several results for the classical set cover problem and its variants [1, 2, 10, 16, 22], and is of interest in its own right.

- Finally, we give the results of a comprehensive experimental study comparing our integer programming and greedy algorithms with other heuristics proposed in the literature. Experiments on both synthetic and human genome test cases show that the new potential function greedy algorithm obtains significant reductions in the number of primers with highly scalable running time.

The rest of the chapter is organized as follows. In next section we introduce notations and give a formal problem definition of MP-PCR primer selection with amplification length constraints. In Section 11.3 we introduce the string covering formulation of the problem and give a compact integer program formulation. In Section 11.4 we describe the greedy algorithm, give its performance analysis, and discuss practical implementation issues. Finally, we present experimental results in Section 11.5 and conclude in Section 11.6.

11.2 NOTATIONS AND PROBLEM FORMULATION

Let $\Sigma = \{A, C, G, T\}$ be the four nucleotide DNA alphabet. We denote by Σ^* the set of strings over Σ, and by $|s|$ the length of string $s \in \Sigma^*$. For a string s and an integer $1 \leq t \leq |s|$, we denote by $s[1..t]$ the prefix of length t of s. We use ℓ to denote the required primer length, L to denote the given threshold on PCR amplification length, and n to denote the number of amplification loci. We say that primer $p = p_1 p_2 \dots p_\ell$ *hybridizes* (or *covers*) string $s = s_1 s_2 \dots s_m$ at position $t \leq m - \ell + 1$ if $s_t s_{t+1} \dots s_{t+\ell-1}$ is the reversed Watson-Crick complement of p, i.e., if s_{t+j} is the Watson-Crick complement of $p_{\ell-j}$ for every $0 \leq j \leq \ell - 1$.

For each $i \in \{1, \dots, n\}$, we denote by f^i (respectively r^i) the string preceding the amplification locus in $3'-5'$ order in the forward (respectively reverse) DNA genomic sequence where potentially useful primer binding may occur. More precisely, if the length of the amplification locus is denoted by x ($x = 1$ for SNP genotyping), then f^i and r^i consist of the $L - x - \ell$ DNA bases immediately preceding in $3' - 5'$ order the i-th amplification locus along the forward (respectively reverse) DNA genomic sequence. Note that a primer can hybridize f^i (respectively r^i) only at positions t between 1 and L', where $L' = (L - x - \ell) - (\ell - 1)$. Simple arithmetic shows that two primers that hybridize to f^i and r^i at positions t and t' lead to an amplification product of length at most L if and only if $t + t' \geq L' + 1$ (see Figure 11.1, and note that f^i and r^i, and hence hybridization positions, are indexed in the respective $3' - 5'$ orders, i.e., they increase when moving towards the amplification locus).

Primers p and p' (not necessarily distinct) are said to feasibly amplify SNP locus i if there exist integers $t, t' \in \{1, \dots, L - \ell + 1\}$ such that the following conditions are simultaneously satisfied:

1. p hybridizes at position t of f^i,
2. p' hybridizes at position t' of r^i, and
3. $t + t' \geq L' + 1$.

A set of primers P is said to be an *L-restricted primer cover* for n SNPs defining sequences (f^i, r^i), if, for every $i = 1, \ldots, n$, there exist primers $p, p' \in P$ feasibly amplifying SNP locus i. The *minimum primer set selection problem with amplification length constraints* (MPSS-L) is defined as follows: Given primer length ℓ, amplification length upperbound L, and n pairs of sequences (f^i, r^i), $i = 1, \ldots, n$, find a minimum size L-restricted primer cover consisting of primers of length ℓ.

11.3 INTEGER PROGRAM FORMULATIONS FOR MPSS-L

Fernandes and Skiena [6] proposed an elegant *minimum multicolored subgraph* formulation for primer set selection. In this formulation, each candidate primer is viewed as a graph node, and each two primers that feasibly amplify a desired locus define an edge "colored" by the locus number. The objective is to find a minimum number of nodes inducing edges of all possible colors. The minimum multicolored subgraph formulation can be cast as an integer linear program by introducing a 0/1 variable x_p for every candidate primer p, and a 0/1 variable $y_{p,p'}$ for every two (not necessarily distinct) primers p and p' feasibly amplifying at least one of the SNP loci, as follows [7]:

$$\text{minimize} \quad \sum_{p \in \mathcal{P}} x_p \tag{11.1}$$

subject to

$$\sum y_{p,p'} \geq 1, \quad i = 1, \ldots, n \tag{11.2}$$

$$\sum_{p'} y_{p,p'} \leq x_p, \quad p \in \mathcal{P} \tag{11.3}$$

$$x_p,\ y_{p,p'} \in \{0, 1\} \tag{11.4}$$

where $\mathcal{P}$ is the set of $O(nL)$ candidate primers. The sum in (11.2) is over all pairs (p, p') feasibly amplifying SNP locus i; this set of constraints ensures that each SNP locus is feasibly amplified by two of the selected primers. Constraints (11.3) ensure that only selected primers can be used to amplify SNP loci.

Unfortunately, the integer program (11.1)-(11.4) cannot be used to solve practical MPSS-L problem instances due to its large size. In particular, the number of variables $y_{p,p'}$ can be as large as $\Theta(nL^2)$, which reaches into the millions for typical values of L.

Below we give a much more compact integer program formulation based on a novel string covering formulation of MPSS-L. The key idea is to view MPSS-L as a generalization of the partial set cover problem [22], in which the objective is to cover a certain fraction of the total number of elements of a ground set using the minimum number of given subsets. In the case of MPSS-L the elements to be covered are the non-empty prefixes in $\{f^i[1..j], r^i[1..j] \mid 1 \leq i \leq n,\ 1 \leq j \leq L'\}$, where, as in Section 11.2, $L' = (L - x - \ell) - (\ell - 1)$. Each primer p covers the set of prefixes

$f^i[1..j]$ and $r^i[1..j]$ for which p hybridizes to f^i, respectively r^i, at a position $t \geq j$. The objective is to choose the minimum number of primers that cover at least $L'+1$ of the $2L'$ elements of each set $\{f^i[1..j], r^i[1..j] \mid 1 \leq j \leq L'\}$ for $i \in \{1, \ldots, n\}$. To formulate this as an integer program, we again introduce a 0/1 variable x_p for every candidate primer p, which is set to 1 if and only if primer p is selected. We also introduce 0/1 variables $z(f^i, j)$ (respectively $z(r^i, j)$) for every $i = 1, \ldots, n$, $1 \leq j \leq L'$; such a variable is set to 1 if and only if the prefix $f^i[1..j]$ (respectively $r^i[1..j]$) is covered by at least one of the selected primers. Using these variables, MPSS-L can be formulated as follows:

$$\text{minimize} \quad \sum_{p \in \mathcal{P}} x_p \tag{11.5}$$

subject to

$$\sum_{j=1}^{L'} z(f^i, j) + \sum_{j=1}^{L'} z(r^i, j) \geq L'+1 \quad i = 1, \ldots, n \tag{11.6}$$

$$z(f^i, j) \leq \sum_{\substack{p \text{ hybridizes} \\ \text{to } f^i \text{ at } t \geq j}} x_p, \quad i = 1, \ldots, n, 1 \leq j \leq L' \tag{11.7}$$

$$z(r^i, j) \leq \sum_{\substack{p \text{ hybridizes} \\ \text{to } r^i \text{ at } t \geq j}} x_p, \quad i = 1, \ldots, n, 1 \leq j \leq L' \tag{11.8}$$

$$x_p,\ z(f^i, j),\ z(r^i, j) \in \{0, 1\} \tag{11.9}$$

Integer program (11.5)-(11.9) has $O(nL)$ variables and $O(nL)$ constraints. However, its solution via general purpose solvers such as CPLEX still requires prohibitively long runtime, mostly due to the fact that each constraint has $O(L)$ variables, and therefore the underlying integer program matrix is relatively dense. An equivalent formulation leading to a much sparser matrix, and, in practice, to greatly reduced runtime, is obtained as follows. Let $p(f^i, j)$ (respectively $p(r^i, j)$) be the unique primer hybridizing at position j of f^i (respectively r^i). Constraints (11.7) ensure that $z(f^i, j)$ is set to 1 only when at least one of the primers hybridizing to f^i at a position $t \geq j$ is selected. This happens if either $p(f^i, j)$ or a primer hybridizing to f^i at a position $t > j$ is selected, and in the latter case $z(f^i, j+1)$ will be set to 1 as well. Thus, constraints (11.7) can be replaced by

$$z(f^i, L') \leq x_{p(f^i, L')}, \quad i = 1, \ldots, n \tag{11.10}$$

$$z(f^i, j) \leq x_{p(f^i, j)} + z(f^i, j+1), \quad i = 1, \ldots, n, 1 \leq j < L' \tag{11.11}$$

and (11.8) can be similarly replaced by the nL' constraints obtained from (11.10) and (11.11) after substituting r^i for f^i.

1. $P \leftarrow \emptyset$
2. While $\Phi(P) < n(L'+1)$ do
   (a) Find a primer $p \notin P$ maximizing $\delta(p, P) := \Phi(P \cup \{p\}) - \Phi(P)$
   (b) $P \leftarrow P \cup \{p\}$
3. Return P

Fig. 11.2 The generic greedy algorithm.

11.4 A GREEDY ALGORITHM

In this section we describe an efficient greedy algorithm for MPSS-L and then establish its approximation guarantee. The algorithm, which can be seen as a generalization of the greedy algorithm for the set cover problem, critically exploits the string covering formulation introduced in Section 11.3 . To enable future application of our techniques to other covering problems, we describe the algorithm and its analysis using an axiomatic framework based on monotonic potential functions.

For a set of primers P, let $\Phi_i(P)$ denote the minimum between $L'+1$ and the number of prefixes of $\{f^i[1..j], r^i[1..j] \mid 1 \leq j \leq L'\}$ covered by at least one primer in P. Also, let $\Phi(P) = \sum_{i=1}^{n} \Phi_i(P)$. The following properties of the integer valued set function Φ are immediate:

(A1) $\Phi(\emptyset) = 0$.

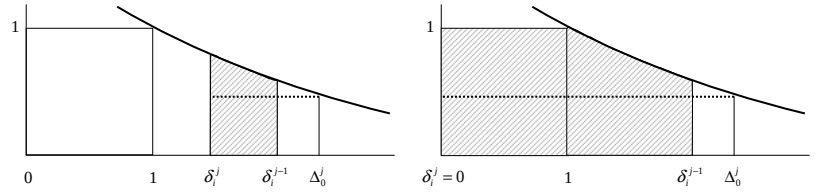
(A2) There exists a constant $\Phi_{\max}$ such that $\Phi(P) = \Phi_{\max}$ if and only if P is a feasible solution ($\Phi_{\max} = n(L'+1)$ for MPSS-L).

(A3) Φ is a non-decreasing set function, i.e., $\Phi(P) \geq \Phi(P')$ whenever $P \supseteq P'$, and, furthermore, for every P such that $\Phi(P) < n(L'+1)$, there exists $p \notin P$ such that $\Phi(P \cup \{p\}) > \Phi(P)$.

Properties (A1)–(A3) suggest using $\Phi(\cdot)$ as a measure of progress towards feasibility, and employing the generic greedy algorithm in Figure 11.2 to solve MPSS-L. The greedy algorithm starts with an empty set of primers and then iteratively adds the primer that gives the largest increase in Φ, until reaching feasibility. By (A1)-(A3) this algorithm will end in a finite number of steps and will return a feasible MPSS-L solution.

Let us denote by $\Delta(p, P)$ the increase in Φ (also referred to as the "gain") obtained by adding primer p to set P, i.e., $\Delta(p, P) = \Phi(P \cup \{p\}) - \Phi(P)$. By (A3), it follows that the gain function Δ is non-negative. It is easy to verify that Δ is also monotonically non-increasing in the second argument, i.e.,

(A4) $\Delta(p, P) \geq \Delta(p, P')$ for every p and $P \subseteq P'$.



[!p] (a) (b)

Fig. 11.3 A graphical illustration of the cost lower-bound used in the proof of Theorem 11.1 for $\delta_i^{j-1} \geq \delta_i^j > 0$ (a), and for $\delta_i^{j-1} > \delta_i^j = 0$ (b). In each case, c_i^j is equal to the area shaded under the curve $\min\{1, 1/x\}$. Since $\Delta_0^j \geq \delta_i^{j-1}$, the shaded area is larger than the area of a rectangle with width $\delta_i^{j-1} - \delta_i^j$ and height $1/\Delta_0^j$.

Theorem 11.1. *For every set function Φ satisfying (A1)-(A4), the greedy algorithm in Figure 11.2 returns a feasible solution of size at most $1 + \ln \Delta$ times larger than the optimum, where $\Delta = \max_{p,P} \Delta(p, P)$.*

Proof. We begin with some additional notations. Let $P^* = \{p_1^*, p_2^*, \ldots, p_k^*\}$ be an optimum solution, i.e., a feasible set of primers of minimum size. Let also $P = \{p_1, p_2, \ldots, p_g\}$ denote the solution returned by the greedy algorithm, with primers indexed in the order in which they are selected by the algorithm. Let $\Phi_i^j = \Phi(\{p_1^*, \ldots, p_i^*\} \cup \{p_1, \ldots, p_j\})$, $\Delta_i^j = \Phi_i^j - \Phi_i^{j-1}$, and $\delta_i^j = \Phi_i^j - \Phi_{i-1}^j$. Note that, by (A4) and (A2), $\Delta_0^j \geq \Delta_1^j \geq \ldots \geq \Delta_k^j = 0$ for every $0 \leq j \leq g$, and $\delta_i^0 \geq \delta_i^1 \geq \ldots \geq \delta_i^g = 0$ for every $0 \leq i \leq k$. Furthermore, note that $\Delta_0^j \geq \delta_i^{j-1}$ for every $1 \leq i \leq k$ and $1 \leq j \leq g$. Indeed, Δ_0^j is the gain achieved by the greedy algorithm when selecting primer p_j. This gain must be at least $\Delta(p_i^*, \{p_1, ..., p_{j-1}\})$ since the greedy algorithm selects the primer with maximum gain in each iteration. Finally, by (A4), $\Delta(p_i^*, \{p_1, ..., p_{j-1}\}) \geq \Delta(p_i^*, \{p_1, ..., p_{j-1}\} \cup \{p_1^*, \ldots, p_{i-1}^*\}) = \Phi_i^{j-1} - \Phi_{i-1}^{j-1} = \delta_i^{j-1}$.

To analyze the size of the solution produced by the greedy algorithm, we use a charging scheme in which a certain cost is assigned to each primer in the optimal solution for every greedy primer. More precisely, the cost charged to p_i^* by the greedy primer p_j is

$$c_i^j = \begin{cases} \ln(\delta_i^{j-1}) - \ln(\delta_i^j), & \text{if } \delta_i^{j-1} \geq \delta_i^j > 0 \\ \ln(\delta_i^{j-1}) + 1, & \text{if } \delta_i^{j-1} > \delta_i^j = 0 \\ 0, & \text{if } \delta_i^{j-1} = \delta_i^j = 0 \end{cases}$$

Notice that the total cost charged to primer p_i^*, $\sum_{j=1}^{g} c_i^j$, is a telescopic sum equal to $1 + \ln(\delta_i^0) \leq 1 + \ln \Delta$. Hence, the overall cost is at most $k(1 + \ln \Delta)$. To prove the approximation factor of $1 + \ln \Delta$ it suffices to prove that we charge at least one unit of cost for each greedy primer. Indeed, consider a fixed $j \in \{1, \ldots, g\}$. Since

$\Delta_0^j \geq \delta_i^{j-1}$, it follows that

$$c_i^j \geq \frac{\delta_i^{j-1} - \delta_i^j}{\Delta_0^j}$$

for every $1 \leq i \leq k$ (see Figure 11.3). Using that $\delta_i^{j-1} - \delta_i^j = \Delta_j^{i-1} - \Delta_j^i$ and $\Delta_j^k = 0$ gives

$$\sum_{i=1}^{k} c_i^j \geq \sum_{i=1}^{k} \frac{\Delta_j^{i-1} - \Delta_j^i}{\Delta_0^j} = 1$$

which completes the proof. □

Note that the maximum gain Δ is at most nL, and therefore Theorem 11.1 implies a worst case approximation factor of $1 + \ln(nL)$ for MPSS-L. For practical MPSS-L instances, Δ is much smaller than nL, implying a significantly better approximation factor on these instances.

11.4.1 Implementation details

In this section we discuss the details of an efficient implementation of the generic greedy algorithm in Figure 11.2. First, we note that although there are 4^ℓ DNA sequences of length ℓ, no more than $2nL$ of these sequences (substrings of length ℓ of the input genomic sequences $\mathcal{S} = \{f^i, r^i \mid 1 \leq i \leq n\}$) can be used as primers. Our implementation starts by creating a list with all feasible primers by removing substrings that do not meet user-specified constraints on GC content and melting temperature T_m (computed as in the Primer3 package [21]). Masking of repetitive elements and more stringent candidate filtering based, e.g., on sophisticated statistical scoring models [24] can also be easily incorporated in this pre-processing step. For each surviving candidate primer we precompute all hybridization positions within the strings of $\mathcal{S}$, which allows computing the coverage gain of a primer candidate p in time $O(n_p)$, where n_p is the number of hybridization positions for p. The primer with maximum gain is then found in step 2(a) of the algorithm by sequentially computing the gain of each remaining primer.

In order to speed up the implementation, we use two further optimizations. A feasible primer is called *unique* if it hybridizes only one of the sequences in $\mathcal{S}$. The first optimization is to retain only the unique feasible primer closest to the amplification locus for each f^i and r^i. The exact number of eliminated unique candidate primers depends on primer length ℓ and number of amplification loci, but is often a significant fraction of the number of feasible candidate primers. Clearly, removing these primers does not worsen the quality of the returned solution.

The second optimization is to adopt a lazy strategy for recomputing primer gains in step 2(a). In first execution of step 2(a) we compute and store the gain for all feasible primers. In subsequent iterations, the gain of a primer is only recomputed if the saved gain is higher than the best gain seen in current iteration. Since gains are

monotonically non-increasing, this optimization is not affecting the set of primers returned by the algorithm.

11.5 EXPERIMENTAL RESULTS

We performed experiments on test cases extracted from the human genome databases as well as simulated test cases. The human genome test cases are regions surrounding known SNPs collected from National Center for Biotechnology Information's genomic databases. Random test cases were generated from the uniform distribution induced by assigning equal probabilities to each nucleotide. All experiments were run on a PowerEdge 2600 Linux server with 4 Gb of RAM and dual 2.8 GHz Intel Xeon CPUs – only one of which is used by our sequential implementations – using the same compiler optimization options. Integer programs were solved using the CPLEX solver version 9.1 with default parameters.

For all experiments we used a bound $L = 1000$ on the PCR amplification length, and a bound ℓ between 8 and 12 on primer length. Although it has been suggested that such short primers may not be specific enough [9], we note that hybridization outside the target region will not result in significant amplification unless *two* primers hybridize sufficiently closely to each other, a much less likely event [6]. Indeed, the feasibility of using primers with only 8-12 target specific nucleotides for simultaneous amplification of thousands of loci has been experimentally validated by Jordan et al. [11].[2] The potential function greedy algorithm in Figure 11.2, referred to as G-POT, was implemented as described in Section 11.4.1, except that, in order to facilitate comparison with other algorithms we did not use any constraints on the GC content or melting temperature of candidate probes.

We ran experiments modeling two different scenarios. In the first scenario the amplification target is a set of SNP loci where no two loci are within a distance of L of each other; under this scenario, the number of primers can only be reduced by primer reuse between different amplification reactions. In the second scenario the amplification target is the set of all confirmed SNP loci within a gene, which results in much closer SNP loci. In this case primer minimization is achieved by both primer reuse and inclusion of multiple SNP loci in a single amplification product.

11.5.1 Amplification of Sparse Sets of SNP Loci

In the first set of experiments we compared G-POT with the following algorithms:

- The iterative beam-search heuristic of Souvenir et al. [23]. We used the primer-threshold version of this heuristic, MIPS-PT, with degeneracy bound set to 1

[2] In addition to 8-12 specific nucleotides at the $3'$ end, primers used in Jordan et al. contain a $5'$ end sequence (CTCGAGNNNNNN) consisting of a fixed G/C rich $5'$ anchor and 6 fully degenerate nucleotides.

and the default values for the remaining parameters (in particular, beam size was set to 100).

- The greedy primer cover algorithm of Pearson et al. [19] (G-FIX). In this algorithm the candidate primers are collected from the reverse and forward sequences within a distance of $L/2$ around the SNP. This ensures that the resulting set of primers meets the product length constraints. The algorithm repeatedly selects the candidate primer that covers the maximum number of not yet covered forward and reverse sequences.

- The optimum primer cover of the reverse and forward sequences within $L/2$ bases of each SNP (OPT-FIX), computed by running CPLEX on a natural integer program formulation of the problem.

- A naïve modification of G-FIX, referred to as G-VAR, in which the candidate primers are initially collected from the reverse and forward sequences within a distance of L around the SNP. The algorithm proceeds by greedily selecting primers like G-FIX, except that when a primer p covers for the first time one of the forward or reverse sequences corresponding to a SNP, say at position t, we appropriately truncate the opposite sequence to a length of $L - t$ to ensure that the final primer cover is L-restricted.

- The optimum MPSS-L solution (OPT) computed by running CPLEX on the compact integer linear program formulation described in Section 11.3.

Table 11.1 gives the number of primers selected and the running time (in CPU seconds) for the compared methods on on instances consisting of up to 100 SNP loci extracted from the NCBI repository. The optimum primer cover of the reverse and forward sequences within $L/2$ bases of each SNP can be found by CPLEX for all instances, often in time comparable to that required by G-FIX. In contrast, the integer linear program in Section 11.3 can be solved to optimality only for small instance sizes. For instances with 100 SNPs, even finding good feasible solutions to this ILP seems difficult for general purpose solvers like CPLEX. Among greedy algorithms, G-POT has the best performance on all test cases, reducing the number of primers by up to 24% compared to G-FIX and up to 30% compared to G-VAR. In most cases, G-POT gives fewer primers than OPT-FIX, and always comes very close to the optimum MPSS-L solutions computed using CPLEX whenever the latter are available. The MIPS-PT heuristic has the poorest performance in both runtime and solution quality, possibly because it is fine-tuned to perform well with high degeneracy primers.

To further characterize the performance of the three greedy algorithms, in Figure 11.4 we plot their average solution quality versus the number of target SNPs (on a logarithmic scale) for randomly generated test cases. MIPS and the integer programming methods are not included in this comparison due to their non-scalable running time. In order to facilitate comparisons across instance sizes, the size of the primer cover is normalized by the double of the number of SNPs, which is the size of the trivial cover obtained by using two distinct primers to amplify each SNP. Although

Table 11.1 Number of primers (#P) and runtime in seconds (CPU) on NCBI test cases for primer length $\ell = 8, 10, 12$ and amplification length constraint $L = 1000$. Entries marked with a dagger represent the best feasible solutions found by CPLEX in 24 hours.

#	ℓ	MIPS-PT		G-FIX		OPT-FIX		G-VAR		G-POT		OPT	
SNPs		#P	CPU	#P	CPU	#P	CPU	#P	CPU	#P	CPU	#P	CPU
10	8	5	3	4	0.01	4	0.01	4	0.02	4	0.02	3	372
	10	6	4	5	0.00	5	0.01	7	0.03	6	0.03	5	979
	12	10	6	8	0.00	8	0.01	9	0.03	7	0.03	6	518
20	8	8	10	7	0.04	6	0.04	7	0.08	6	0.10	5	112,407
	10	13	15	9	0.03	8	0.01	10	0.08	9	0.08	7	13,494
	12	18	26	14	0.04	14	0.01	13	0.08	13	0.11	$11^{\dagger}$	24h
30	8	12	24	9	0.11	8	0.07	9	0.18	7	0.12	$8^{\dagger}$	24h
	10	18	37	14	0.07	12	0.03	13	0.14	12	0.17	$11^{\dagger}$	24h
	12	26	84	20	0.12	19	0.03	19	0.19	21	0.15	$15^{\dagger}$	24h
40	8	17	35	10	0.09	9	0.84	15	0.27	10	0.25	$10^{\dagger}$	24h
	10	24	49	19	0.16	15	0.05	21	0.22	14	0.20	$15^{\dagger}$	24h
	12	32	183	24	0.10	24	0.03	25	0.23	22	0.28	$21^{\dagger}$	24h
50	8	21	48	13	0.13	11	5.87	15	0.30	10	0.32	$12^{\dagger}$	24h
	10	30	150	23	0.22	19	0.06	24	0.36	18	0.33	$19^{\dagger}$	24h
	12	41	246	31	0.14	29	0.03	32	0.30	29	0.28	$25^{\dagger}$	24h
100	8	32	226	17	0.49	16	180.42	20	0.89	14	0.58	$121^{\dagger}$	24h
	10	50	844	37	0.37	30	0.23	37	0.72	31	0.75	$35^{\dagger}$	24h
	12	75	2601	53	0.59	45	0.09	48	0.84	42	0.61	$46^{\dagger}$	24h

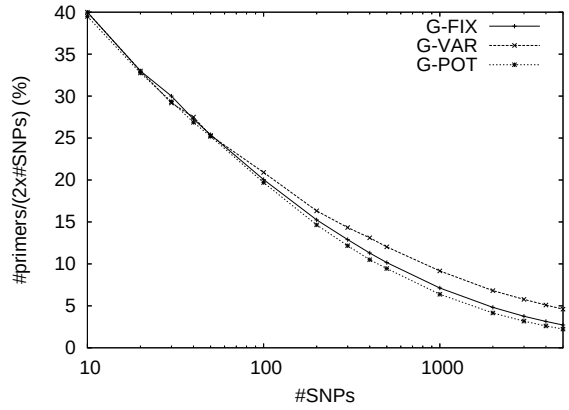
the improvement is highly dependent on primer length and number of SNPs, G-POT is still consistently outperforming the G-FIX algorithm and, with few exceptions, its G-VAR modification.

Figure 11.5 gives a log-log plot of the average CPU running time (in seconds) versus the number of pairs of sequences for primers of size 10 and randomly generated pairs of sequences. The runtime of all three greedy algorithms grows linearly with the number of SNPs, with G-VAR and G-POT incurring only a small factor penalty in runtime compared to G-FIX. This suggests that a robust practical meta-heuristic is to run all three algorithms and return the best of the three solutions found.

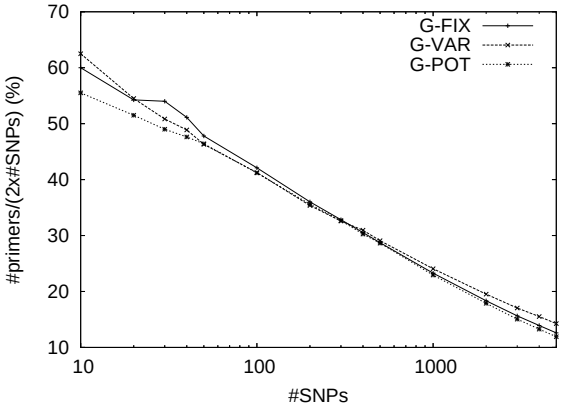
11.5.2 Amplification of Dense Sets of SNP Loci

In a second set of experiments we used as amplification targets the SNP loci identified and verified within 14 genes at the Program for Genomic Applications of the University of Washington [20]. For each gene, we consider SNP loci within all exons and introns, within the first 2,000 bp upstream of first exon, and within the first 1,500 bp downstream of the poly-A signal.

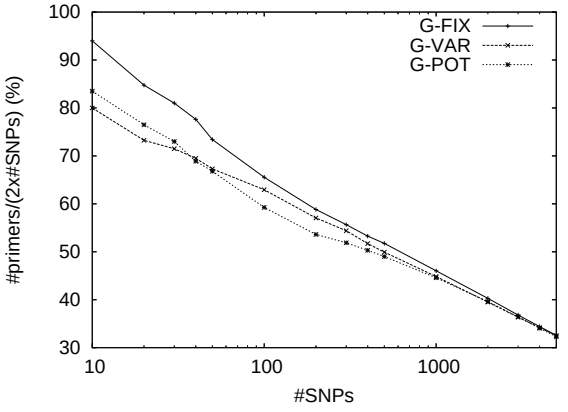
In addition to G-FIX, G-VAR, and G-POT, on these testcases we also ran a natural greedy primer selection algorithm, referred to as *greedy intervals* (G-INT), which works as follows. First, G-INT selects a forward primer immediately upstream of



$\ell = 8$



$\ell = 10$



$\ell = 12$

Fig. 11.4 Performance of the compared algorithms, measured as relative improvement over the trivial solution of using two primers per SNP, for $\ell = 8, 10, 12$, $L = 1000$, and up to 5000 SNPs. Each number represents the average over 10 test cases of the respective size.

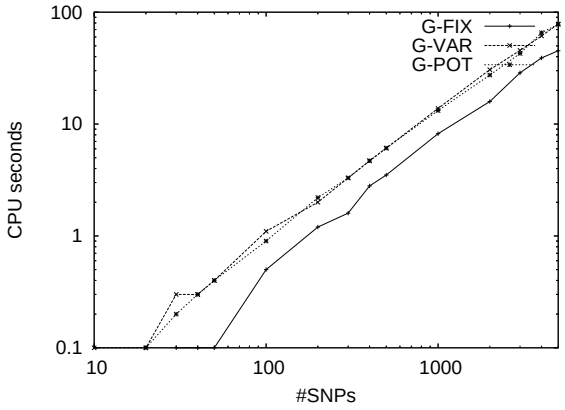


Fig. 11.5 Average runtime of the compared algorithms for $\ell = 10$, $L = 1000$, and up to 5000 SNPs.

the leftmost SNP locus, and pairs it up with a reverse primer placed as far as possible downstream subject to the product length constraint. All SNP loci covered by the selected pair of primers are removed, and the above step is repeated until all loci are covered. It is easy to see that this algorithm minimizes the *number of amplification products* required to cover the given SNP loci. As shown in Table 11.2, G-POT continues to consistently outperform the other algorithms, with G-INT and G-VAR producing fewer primers for a few testcases.

11.6 CONCLUSIONS

In this chapter we have proposed exact algorithms based on integer programming and a more scalable potential function greedy approximation algorithm for MP-PCR primer set selection with amplification length constraints, and have presented experimental results showing that our methods lead to significant reductions in the number of primers compared to previous algorithms. Open source C implementations of both algorithms are available at http://dna.engr.uconn.edu/˜software/G-POT/.

A promising approach to further increasing MP-PCR efficiency is the use of *degenerate PCR primers* [14, 15, 23], see also Section 5.3.3. A degenerate primer is essentially a mixture consisting of multiple non-degenerate primers sharing a common pattern. Remarkably, degenerate primer cost is nearly identical to that of a non-degenerate primer, since the synthesis requires the same number of steps (the only difference is that one must add multiple nucleotides in some of the synthesis steps). Since degenerate primers may lead to excessive unintended amplification, a bound on the degeneracy of a primer (i.e., the number of distinct non-degenerate primers in the mixture) is typically imposed [15, 23].

Our greedy algorithm extends directly to the problem of selecting, for a given set of genomic loci, a minimum size L-restricted primer cover consisting of degenerate primers with bounded degeneracy. However, even for moderate degeneracy constraints, it becomes impractical to explicitly evaluate the gain function for all candidate primers. Indeed, as remarked by Linhart and Shamir [15], the number of candidate degenerate primers may be as large as $2nL\binom{k}{\delta}15^{\delta}$, where n is the number of loci, L is the PCR amplification length upperbound, and δ is the number of "degenerate nucleotides" allowed in a primer. To maintain a practical runtime, one may sacrifice optimality of the greedy choice in step 2(a) of the greedy algorithm, using instead approximation algorithms similar to those of Linhart and Shamir [15] for finding degenerate primers guaranteed to have near optimal gain. The analysis in Section 11.4 extends to this modification of the greedy algorithm as follows:

Theorem 11.2. *Assume that the greedy algorithm in Figure 11.2 is modified to select in step 2(a) a primer whose gain is within a factor of α of the maximum possible gain, for some fixed $0 < \alpha \leq 1$. Then, the modified algorithm returns an L-restricted primer cover of size at most $(1 + \ln \Delta)\alpha$ times larger than the optimum, where $\Delta = \max_{p,P} \Delta(p, P)$.*

Another interesting direction for future research is extending primer selection algorithms to ensure that there is no cross-hybridization between selected primers, which is one of the main causes of amplification failure in MP-PCR [8]. Cross-hybridization constraints can be directly enforced in the integer program in Section 11.3 by the addition of inequalities of the form $x_p + x_{p'} \leq 1$ for every two primers p and p' predicted to cross-hybridize. The potential function greedy algorithm can also ensure lack of primer cross-hybridization via a simple modification: after selecting a primer p, discard all candidates predicted to cross-hybridize with p. Although this modification does no longer guarantee that the resulting set of primers is near-minimal, preliminary experiments show that in practice it leads to only minor increases in the number of primers.

Acknowledgments

The work of KMK and AAS was supported in part by NSF ITR grant 0121277. The work of IIM was supported in part by NSF CAREER award IIS-0546457, NSF DBI grant 0543365, and a Large Grant from the University of Connecticut's Research Foundation. A preliminary version of this work has appeared in [12].

References

1. P. Berman, B. DasGupta, and M.-Y. Kao. Tight approximability results for test set problems in bioinformatics. In *Proc. 9th Scandinavian Workshop on Algorithm Theory (SWAT)*, pages 39–50, 2004.

2. V. Chvátal. A greedy heuristic for the set covering problem. *Mathematics of Operations Research*, 4:233–235, 1979.

3. ILOG Corp. Cplex optimization suite, http://www.ilog.com/products/cplex.

4. K. Doi and H. Imai. A greedy algorithm for minimizing the number of primers in multiple PCR experiments. *Genome Informatics*, 10:73–82, 1999.

5. U. Feige. A threshold of $\ln n$ for approximating set cover. *Journal of the ACM*, 45:634–652, 1998.

6. R.J. Fernandes and S.S. Skiena. Microarray synthesis through multiple-use PCR primer design. *Bioinformatics*, 18:S128–S135, 2002.

7. M.T. Hajiaghayi, K. Jain, L.C. Lau, I.I. Măndoiu, A.C. Russell, and V.V. Vazirani. The minimum multicolored subgraph problem in haplotyping and PCR primer set selection. In V.N. Alexandrov et al., editor, *Proc. 6th International Conference on Computational Science (ICCS 2006), Part II*, volume 3992 of *Lecture Notes in Computer Science*, pages 758–766, Berlin, 2006. Springer-Verlag.

8. O. Henegariu, N.A. Heerema, S.R. Dlouhy, G.H. Vance, and P.H. Vogt. Multiplex PCR: critical parameters and step-by-step protocol. *Biotechniques*, 23:504–511, 1997.

9. M.-H. Hsieh, W.-C. Hsu, S.-Kay, and C.M. Tzeng. An efficient algorithm for minimal primer set selection. *Bioinformatics*, 19:285–286, 2003.

10. D.S. Johnson. Approximation algorithms for combinatorial problems. *Journal of Computer and System Sciences*, 9:256–278, 1974.

11. B. Jordan, A. Charest, J.F. Dowd, J.P. Blumenstiel, R. f. Yeh, A. Osman, D.E. Housman, and J.E. Landers. Genome complexity reduction for SNP genotyping analysis. *Proc. Natl. Acad. Sci. USA*, 99:2942–2947, 2002.

12. K.M. Konwar, I.I. Măndoiu, A.C. Russell, and A.A. Shvartsman. Improved algorithms for multiplex PCR primer set selection with amplification length constraints. In Y.-P. Phoebe Chen and L. Wong, editors, *Proc. 3rd Asia-Pacific Bioinformatics Conference (APBC)*, pages 41–50, London, 2005. Imperial College Press.

13. P.Y. Kwok. Methods for genotyping single nucleotide polymorphisms. *Annual Review of Genomics and Human Genetics*, 2:235–258, 2001.

14. S. Kwok, S.Y. Chang, J.J. Sninsky, and A. Wong. A guide to the design and use of mismatched and degenerate primers. *PCR Methods and Appl.*, 3:S539–S547, 1994.

15. C. Linhart and R. Shamir. The degenerate primer design problem. *Bioinformatics*, 18:S172–S181, 2002.

16. L. Lovász. On the ratio of optimal integral and fractional covers. *Discrete Mathematics*, 13:383–390, 1975.

17. K. Mullis. Process for amplifying nucleic acid sequences. United States Patent 4,683,202, 1987.

18. P. Nicodème and J.-M. Steyaert. Selecting optimal oligonucleotide primers for multiplex PCR. In *Proc. 5th Intl. Conference on Intelligent Systems for Molecular Biology*, pages 210–213, 1997.

19. W.R. Pearson, G. Robins, D.E. Wrege, and T. Zhang. On the primer selection problem for polymerase chain reaction experiments. *Discrete and Applied Mathematics*, 71:231–246, 1996.

20. Program for Genomic Applications, University of Washington. Genes sequenced for snps, http://pga.gs.washington.edu/finished_genes.html.

21. S. Rozen and H.J. Skaletsky. Primer3 on the WWW for general users and for biologist programmers. In S. Krawetz and S. Misener, editors, *Bioinformatics Methods and Protocols: Methods in Molecular Biology*, pages 365–386. Humana Press, Totowa, NJ, 2000. Code available at http://www-genome.wi.mit.edu/genome_software/other/primer3.html.

22. P. Slavik. Improved performance of the greedy algorithm for partial cover. *Information Processing Letters*, 64:251–254, 1997.

23. R. Souvenir, J. Buhler, G. Stormo, and W. Zhang. Selecting degenerate multiplex PCR primers. In *Proc. 3rd Intl. Workshop on Algorithms in Bioinformatics (WABI)*, pages 512–526, 2003.

24. A. Yuryev, J. Huang, K.E. Scott, J. Kuebler, M. Donaldson, M.S. Phillipes, M. Pohl, and M.T Boyce-Jacino. Primer design and marker clustering for multiplex SNP-IT primer extension genotyping assay using statistical modeling. *Bioinformatics*, 20(18):3526–3532, 2004.