Constructing 1-rotational NRDFs through an Optimization Approach: New (46,9,8), (51,10,9) and (55,9,8)-NRBDs

Luis B. Morales

Departamento de Métodos Matemáticos y Numéricos, Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas, Universidad Nacional Autónoma de México, México, DF, 04510, México

Abstract

In this paper we formulate the problem of constructing 1-rotational near resolvable difference families as a combinatorial optimization problem where a global optimum corresponds to a desired difference family. Then, we develop an algorithm based on scatter search in conjunction with a tabu search to construct many of these difference families. In particular, we construct three new near resolvable difference families which lead to an equal number of new 1-rotational near resolvable block designs with parameters: (46,9,8), (51,10,9) and (55,9,8). Our results indicate that this conjunction outperforms both scatter search and tabu search.

Key words: Difference families; Near resolvable block designs; Combinatorial optimization; Scatter and Tabu search algorithms

1 Introduction

Metaheuristics and Combinatorial Optimization methods perform well in many construction problems. In particular, Morales (2000, 2001, 2005) formulated as an optimization problem the construction of difference families, first for balanced incomplete block designs (BIBDs), then for resolvable BIBDs, and finally for partially balanced incomplete block designs with two associate classes. The author of these papers constructed many such families using a tabu search algorithm. Here we formulate the problem of constructing 1-rotational difference families for near resolvable designs as a discrete optimization where we require optimal solutions rather than close approximations to them. The goal of this work is to develop a heuristic optimization technique...
based on the scatter search (SS) (Glover et al., 2001) and tabu search (TS) (Glover, 1989) methodologies to tackle the optimization problem.

Our algorithm was able to construct many near resolvable 1-rotational difference families. In particular, we constructed three new near resolvable 1-rotational difference families which lead to an equal number of new near resolvable balanced incomplete block designs with parameters: (46,9,8), (51,10,9) and (55,9,8), see (Abel and Furino, 1996; Abel et al., 2001, 2007; Furino, 1995; Furino et al., 1996).

In Section 2 we define the difference families for NRBDs. Then we formulate the construction of these difference families as a discrete optimization problem in Section 3. In Sections 4 and 5 a brief review of the basic principles of SS and TS are given respectively, and we present an implementation of scatter search and tabu search for the optimization problem. Computational results are reported in Section 6. The latter section contains the conclusions.

To conclude this introduction, we give some types of design definitions. A BIBD is a pair \((V, B)\) where \(V\) is a \(v\)-set and \(B\) is a collection of \(k\)-subsets of \(V\) called blocks where \(k < v\) such that each pair of elements of \(V\) occurs together in exactly \(\lambda\) blocks. We denote such design as a \(\langle v, k, \lambda\rangle\)-BIBD.

A near parallel class on a BIBD \((V, B)\), with respect to a point \(s\), is a set of blocks that partitions the set \(V - \{s\}\) into \((v - 1)/k\) blocks of that design. We call \(s\) the missing point of this class.

A near resolvable block design (NRBD) is a BIBD \((V, B)\) whose blocks can be partitioned into \(v\) near parallel classes, and every point of \(V\) is missing from exactly one class (see, for example, (Abel and Furino, 1996)). For two different near parallel classes \(\{B_{s,1}, \ldots, B_{s,q}\}\) and \(\{B_{t,1}, \ldots, B_{t,q}\}\), define the near parallel class intersection matrix (NPCIM) (see (Morales et al., 2007)), as the \(q \times q\) matrix \(((n(s, t))_{i,j})\), where \(n(s, t)_{i,j} = |B_{s,i} \cap B_{t,j}|\).

The definition of near resolvable designs implies that if there exists a \(\langle v, k, \lambda\rangle\)-NRBD, then \(v \equiv 1 \pmod{k}\) and \(\lambda \equiv 0 \pmod{k - 1}\).

A \(\langle v, k, \lambda\rangle\)-BIBD \((V, B)\) is 1-rotational if \(V = \mathbb{Z}_{v - 1} \cup \{\infty\}\), and the mapping \(\phi\) from \(i\) to \(i + 1 \pmod{v}\), fixing \(\infty\), is an automorphism of the design.

### 2 1-Rotational Difference Families

**Definition 1** Let \(G\) be an additive group of order \(v\) and let \(N\) be a subgroup of \(G\) of order \(k\). A 1-rotational \(\langle G, N, k, k - 1\rangle\) near resolvable difference family (NRDF) is a partition \(P = \{D_1, \ldots, D_q\}\) of the set \(G \setminus \{0\} \cup \{\infty\}\) with the following two properties

1. Each \(D_2, \ldots, D_q\) contains only elements from \(G\), \(D_1\) contains \(\infty\), and \(|D_i| = k\) for \(1 \leq i \leq q\).
2. Let \(D_i = \{d_{i1}, \ldots, d_{ik}\}\) (\(2 \leq i \leq q\)) and \(D_1 = \{d_{11}, \ldots, d_{1k-1}, \infty\}\). Among the differences \(d_{ix} - d_{iy}\) \((i = 2, \ldots, q; x, y \in \{1, \ldots, k\}, x \neq y\) and \(d_{iz} - d_{iw}\) \((z, w \in \{1, \ldots, k - 1\}, z \neq w\)), each element of \(G \setminus N\) occurs \(k - 1\) times whereas each element of \(N \setminus \{0\}\) occurs \(k - 2\) times.
If $G$ is the cyclic group of order $v$ and $N$ is the subgroup of $G$ of order $k$ then the difference family is denoted by $(v, k, k - 1)$-NRDF.

If the sets $D_1, \ldots, D_q$ satisfy only condition (1) of Definition 1 we say that these sets form a 1-rotational family.

If $D = \{d_1, \ldots, d_k\}$ is a $k$-subset of $\mathbb{Z}_v \cup \{\infty\}$ then the set $D + g = \{d_1 + g, \ldots, d_k + g\} \ (g \in \mathbb{Z}_v) \ (\text{mod } v)$ is called a translate of $D$, with the convention that $\infty + g = \infty$ for all $g \in \mathbb{Z}_v$.

The following theorem explains how to get 1-rotational NRBDs in a general way.

**Theorem 2** Let $D_1, \ldots, D_q$ be the blocks of a 1-rotational $(v - 1, k, k - 1)$-NRDF. Then, the blocks $\{D_1 + g, \ldots, D_q + g \mid g \in \mathbb{Z}_{v-1}\}$ and $\{N + h \mid 0 \leq h < q\}$ form a 1-rotational $(v, k, k - 1)$-NRBD.

**PROOF.** See [Abel et al., 2001, Theorem 3.1].

It follows from Theorem 2 that if $D_1, \ldots, D_q$ are the blocks of a $(v - 1, k, k - 1)$-NRDF, then the translates

$$\{D_1 + g, \ldots, D_q + g\} \quad (1)$$

form a near parallel class of the design with the missing point $g$, for each $g$ in $\mathbb{Z}_{v-1}$. In particular, for $g = 0$, the blocks of (1) form a near parallel class of the design with missing point 0. Moreover,

$$\{N + h \mid 0 \leq h < q\}$$

form a near parallel class of the design with the missing point $\infty$.

Let $D_1, \ldots, D_q$ be a 1-rotational $(v - 1, k, k - 1)$-NRDF. Here $N = \{0, q, \ldots, (k - 1)q\}$. Let $N_g = N + g$ for $1 \leq g < q$. Define

$$S_{i,g} = D_i \cap N_{g} \quad (1 \leq i \leq q; \ 0 \leq g \leq q - 1).$$

Clearly, $(n_{i,g}) = (|S_{i,g}|)$ is the NPCIM of the near parallel classes $\{D_1, \ldots, D_q\}$ and $\{N + h \mid 0 \leq h < q\}$. It is not hard to see that

$$D_1 \setminus \{\infty\} = \bigcup_{g=0}^{q-1} S_{1,g} \quad (2)$$

$$D_i = \bigcup_{g=0}^{q-1} S_{i,g} \quad \text{for } 2 \leq i \leq q \quad (3)$$

$$N_0 \setminus \{0\} = \bigcup_{i=1}^{q} S_{i,0} \quad (4)$$

$$N_g = \bigcup_{i=1}^{q} S_{i,g} \quad \text{for } 1 \leq g < q. \quad (5)$$
Proposition 3 Let \( D_1, \ldots, D_q \) be a 1-rotational \((v-1, k, k-1)\)-NRDF. Then the NPCIM \((n_{i,g})\) satisfies

\[
\begin{align*}
\sum_{g=0}^{q-1} n_{1,g} &= k - 1, & \sum_{g=0}^{q-1} n_{i,g} &= k & \text{for } i = 2, \ldots, q \\
\sum_{i=1}^{q} n_{i,0} &= k - 1, & \sum_{i=1}^{q} n_{i,g} &= k & \text{for } g = 1, \ldots, q \\
\sum_{i=1}^{q} \sum_{g=0}^{q-1} n_{i,g}(n_{i,g} - 1) &= (k - 2)(k - 1). 
\end{align*}
\]

(6) \( \quad \) (7) \( \quad \) (8)

For \( 1 \leq p \leq q - 1 \), we have

\[
\sum_{i=1}^{q} \sum_{0 \leq g \neq h \leq q \atop g-h \equiv p \pmod{q}} n_{i,g}n_{i,h} = (k - 1)k 
\]

(9)

PROOF. (2) and (3) establish (6), and (4) and (5) establish (7), see also (Morales et al., 2007).

Now we prove (8) and (9). Suppose that \( x \) and \( y \) are two different elements of \( S_{i,g} \). Hence \( x = a + g, y = b + g \) \((a, b \in N \setminus \{0\}; 0 \leq g \leq q - 1)\), and \( x - y = a - b \). Since \( N \) is a group, \( x - y \) belongs to \( S_{i,g} \). Moreover, by Definition 1(2), every element of \( N \setminus \{0\} \) occurs \( k - 2 \) times among the differences arising from the \( q \) sets. From this and since \( |N \setminus \{0\}| = k - 1 \), it follows immediately that (8) holds. On the other hand, it is easy to see that if \( x \) in \( S_{i,g} \) and \( y \) in \( S_{i,h} \) with \( g \neq h \), then \( x - y \) belongs to \( S_{i,p} \), where \( p = g - h \pmod{q} \) with \( 1 \leq p \leq q - 1 \). By Definition 1(2), every element of \( G \setminus N \) occurs \( k - 1 \) times among the differences arising from the \( q \) sets. From this and since \( |N_g| = k \) for \( g \neq 0 \), we have (9).

3 Optimization

For a 1-rotational family \( \mathcal{P} = \{D_1, \ldots, D_q\} \), we define a \((v-1)\)-vector \( C = (c_g) \) associated with it, where \( c_g \) is the number of times that \( g \in G \setminus \{0\} \) is a difference arising from the \( q \) sets, and \( c_0 = k - 2 \). Note that a 1-rotational family is a 1-rotational difference family if and only if \( c_g = k - 1 \) for each \( g \neq 0 \pmod{q} \), and \( c_g = k - 2 \) for each \( g \equiv 0 \pmod{q} \).

Let us now formulate the problem of constructing 1-rotational \((v-1, k, k-1)\)-NRDF as an optimization problem. Here \( N = \{0, q, \ldots, (k - 1)q\} \). A feasible solution for our optimization problem is a 1-rotational family \( \mathcal{P} = \{D_1, \ldots, D_q\} \) such that the matrix \((n_{i,g}) = (|D_i \cap N_g|)\) satisfies (6)–(9). We
first define the function
\[
\lambda(g) = \begin{cases} 
  k - 2, & \text{if } g \equiv 0 \pmod{q} \\
  k - 1, & \text{otherwise.}
\end{cases}
\]

Then, we define the objective function as
\[
f(P) = \sum_{g=0}^{v-1} (c_g - \lambda(g))^2, \tag{10}
\]
where the vector \((c_g)\) is associated with the sets \(D_1, \ldots, D_q\). Clearly, the lower bound of the objective function is zero. Therefore, a 1-rotational near resolvable difference family \(P\) can be constructed if and only if the value of \(f(P)\) is 0 in a global minimum.

In our optimization approach, feasible solutions \(D_1, \ldots, D_q\) are generated as follows. First, we use a backtracking algorithm to find a \(q \times q\) non-negative matrix \((n_{i,g})\) satisfying (6)–(7). Second, for each \(1 \leq g \leq q - 1\), we generate randomly a partition \(S_{1,g}, \ldots, S_{q,g}\) of the translate set \(N + g\) with \(|S_{i,g}| = n_{i,g}\), and for \(g = 0\) we generate randomly a partition \(S_{1,0}, \ldots, S_{q,0}\) of the set \(N \setminus \{0\}\) with \(|S_{i,0}| = n_{i,0}\). Then, we define
\[
D_1 = \left( \bigcup_{g=0}^{q-1} S_{1,g} \right) \cup \{\infty\}
\]
\[
D_i = \bigcup_{g=0}^{q-1} S_{i,g} \quad \text{for } 2 \leq i \leq q.
\]

In consequence the size of the search space \(\mathcal{X}\) to construct 1-rotational \((v - 1, k, k - 1)\)-NRDFs is
\[
|\mathcal{X}| = \left[ \binom{k - 1}{n_{1,0}} \binom{k - 1 - n_{1,0}}{n_{2,0}} \binom{k - 1 - n_{1,0} - n_{2,0}}{n_{3,0}} \cdots \binom{n_{q,0}}{n_{q,0}} \right] \times \\
\left[ \prod_{g=1}^{q-1} \binom{k}{n_{1,g}} \binom{k - n_{1,g}}{n_{2,g}} \binom{k - n_{1,g} - n_{2,g}}{n_{3,g}} \cdots \binom{n_{q,g}}{n_{q,g}} \right].
\]

Examples. In the construction of \((46, 9, 8)\), \((51, 10, 9)\) and \((55, 9, 8)\)-NRDFs we use the NPCIMs.
This optimization approach for constructing 1-rotational \((v - 1, k, k - 1)\)-NRDFs is similar to the one used by Morales (2001) for constructing 1-rotational difference families. However, this new definition of the search space allows us to reduce the size of the search space. In Morales (2001), the search space was defined as the set of all partitions \(P = \{D_1, \ldots, D_q\}\). Therefore the size of this search space \(\mathcal{Y}\) is

\[
|\mathcal{Y}| = \binom{v-2}{k} \binom{v-k-2}{k} \cdots \binom{2k-1}{k} \binom{k-1}{k-1}.
\]

For example, if \(v = 46\) and \(k = 9\) we have \(|\mathcal{Y}| \approx 3.8 \times 10^{27}\). In contrast, with our new definition we have \(|\mathcal{X}| \approx 2.9 \times 10^{19}\).

In our approach a 1-rotational family \(D_1, \ldots, D_q\) is represented by the data structure: a \((v - 1)\)-array \(W = (w_j)\), where

\[
(w_1, \ldots, w_{v-1}) = (s^{(1)}_{1,0}, \ldots, s^{(n_1,0)}_{1,0}, \ldots, s^{(1)}_{1,q-1}, \ldots, s^{(n_1,q-1)}_{1,q-1}, \ldots, s^{(q,q-1)}_{q,q-1}, \ldots, s^{(n_q,q-1)}_{q,q-1}).
\]

Here \(\{s^{(1)}_{i,g}, \ldots, s^{(n_1,g)}_{i,g}\} = D_i \cap N_g\).

4 Scatter Search

Scatter search is an evolutionary algorithm that operates on a set of solutions, the reference set, by strategically combining. In SS the main mechanism for combining solutions is such that a new solution is created from the strategic combination of other solutions to explore the search space. For a detailed description of the SS methodology see (Glover et al., 2001).

Unlike a “population” in genetic algorithms, the reference set in scatter search tends to be small. In genetic algorithms, two solutions are randomly chosen from the population and a “crossover” or combination mechanism is applied to generate one or more offspring. A typical population size in a genetic algorithm consists of 100 elements, which are randomly sampled to create combinations. In contrast, scatter search chooses two or more elements of the

\[
\begin{bmatrix}
3 & 0 & 2 & 1 & 2 \\
2 & 1 & 1 & 2 & 3 \\
1 & 3 & 3 & 1 & 1 \\
1 & 3 & 2 & 1 & 2 \\
1 & 2 & 1 & 4 & 1
\end{bmatrix}, \quad
\begin{bmatrix}
2 & 1 & 2 & 2 \\
4 & 3 & 1 & 2 \\
3 & 2 & 2 & 1 \\
0 & 2 & 3 & 2 \\
0 & 2 & 2 & 3
\end{bmatrix}, \quad
\begin{bmatrix}
2 & 0 & 0 & 1 & 2 & 3 \\
2 & 1 & 1 & 2 & 2 & 1 \\
1 & 2 & 3 & 0 & 2 & 1 \\
1 & 4 & 0 & 2 & 1 & 1 \\
2 & 1 & 2 & 2 & 1 & 1 \\
0 & 1 & 3 & 2 & 1 & 2
\end{bmatrix},
\]

respectively.
reference set in a structured and intelligent way with the purpose of creating new solutions. Since the combination process considers at least all pairs of solutions in the reference set, there is a practical need for keeping the cardinality of the set small. Scatter search consists of five component processes:

1. Diversification Generation Method: generate a starting set of trial solutions to guarantee a critical level of diversity.
2. Improvement Method: given a trial solution as input, a heuristic process is applied to improve the solution.
3. Reference Set Update Method: builds and updates the reference set $\text{RefSet}$ consisting of a set of the best solutions found, widespread throughout the solution space.
5. Solution Combination Method: combines the subsets of solutions into one or more trial solutions.

The basic scatter search procedure starts generating a large set of diverse solutions, which is obtained using the Diversification Generation Method. This procedure creates the initial population $\text{Pob}$ ($\text{Pobsize} = |\text{Pob}|$), which must be a wide set consisting of diverse and good solutions. An Improvement Method is applied to each solution obtained by the previous method reaching a better solution, which is added to $\text{Pob}$.

A set of good representative solutions of the population is chosen to generate the reference set ($\text{RefSet}$). The good solutions are not limited to those with the best objective function values. The considered reference set consists of the union of two subsets, $\text{RefSet}_1$ and $\text{RefSet}_2$ ($|\text{RefSet}| = b = b_1 + b_2 = (|\text{RefSet}_1| + |\text{RefSet}_2|)$). The $b_1$ solutions with the best objective function values are included in the $\text{RefSet}_1$, which is the reference set of the “high quality” solutions. While the $\text{RefSet}_2$ is generated with the $b_2$ most diverse solutions in the population. With the purpose of evaluating the dispersion among solutions, a distance between them is considered.

In the iterative process of the SS algorithm, the first method used is the Subset Generation Method, which selects several subsets of solutions from the $\text{RefSet}$ to be combined by the Solution Combination Method that uses these subsets to produce new trial solutions. Then, the Improvement Method is applied to the result of the combination to get an improved solution. These improved solutions are placed in a solution pool, denoted by $\text{Pool}$. The best $b_1$ solutions are chosen from the union of the current $\text{RefSet}_1$ and the set $\text{Pool}$ to update $\text{RefSet}_1$. The set $\text{RefSet}_2$ is updated similar as the set $\text{RefSet}_1$, but according to the dispersion among solutions. The process is repeated a certain number, $\text{itermax} = 4$, of iterations while the set $\text{RefSet}$ converges.

Regarding SS implementation, the algorithm uses the objective function defined by (10). Our diversification method employs controlled randomization using frequency memory to generate a set of diverse solutions. This method generates the initial set $\text{Pob}$ taking care that the frequency that any pair of elements of $Z_{v-1} \setminus \{0\}$ appears in the same block of a 1-rotational family in
$Pob$ is most nearly the same for all pairs. This method may be written in pseudo-code as follows:

```plaintext
global $F.rec[g][p]$, $1 \leq g, p < v$
procedure $F.recSum(a, S)$
    $sum = 0$
    for each $g$ in $S$
        $sum = sum + F.rec[a][g]$
    return $sum$
main
Let $N = \{0, q, \ldots, (k - 1)q\}$, $N[0] = N \setminus \{0\}$, $N[j] = N + j$, $1 \leq j < q$
$Pob = \emptyset$, $F.rec[g][p] = 0$, $1 \leq g, p < v$
while ($|Pob| < Pobsize$) do
    $S[i][j] = \emptyset$, $1 \leq i \leq q$, $0 \leq j < q$
    for $i = 1$ to $q$ do
        for $j = 0$ to $q - 1$ do
            Select at random $n$ in $T = N[j] - \bigcup_{t=1}^{j} S[t][j]$ such that
            $F.recSum(n, S[i][j]) \leq F.recSum(g, S[i][j])$ for each $g$ in $T$
            $S[i][j] = S[i][j] \cup \{n\}$
            for each $g$ in $S[i][j]$ do
                $F.rec[n][g] = F.rec[g][n] = F.rec[n][g] + 1$
            endfor
        endfor
    endfor
    $Pob = Pob \cup \{S[1][0], \ldots, S[q][0], \ldots, S[1][q - 1], \ldots, S[q][q - 1]\}$
endwhile

The Improvement Method, is then applied to each solution obtained by the previous method. The Improvement method used in this work was a tabu search algorithm, which is described in the next section.

The distance between two solutions $W$ and $Z$ is defined as follows:

$$Dist(W, Z) = \sum_{i=1}^{v-1} |w_i - z_i|. \quad (11)$$

For each solution in $Pob$, the minimum of the distances to the solutions in $RefSet$ is computed. Then, the solution with maximum of these distances is selected. This solution is added to $RefSet_2$ and deleted from $P$ and the minimum distances are updated.

The next step is to combine the solutions of the reference set. Once $(x_1^i)$ and $(x_2^i)$ are selected in the reference set and we let $\{\ell, h\} = \{1, 2\}$, two types of combinations $T_1$ and $T_2$ are generated as follows:

Choose a $q$-vector $(\alpha_0, \ldots, \alpha_{q-1}) \neq (0, \ldots, 0)$, where $\alpha_j$ is a random number in $N = \{0, q, \ldots, (k - 1)q\}$ and there is at least some $\alpha_j = 0$. (The probability
of selecting an $\alpha_j \neq 0$ is given by the parameter $\text{prob}$). Then we define

$$T_1(\ell, h) : z_i^{(\ell, h)} = \begin{cases} x_i^{\ell} , & \text{if } x_i^{\ell} \equiv j \pmod{q} \text{ and } \alpha_j \neq 0, \\ x_i^{h} , & \text{if } x_i^{\ell} \equiv j \pmod{q} \text{ and } \alpha_j = 0, \end{cases}$$

$$T_2(\ell, h) : z_i^{(\ell, h)} = \begin{cases} x_i^{\ell} + \alpha_j \pmod{v} , & \text{if } x_i^{\ell} \equiv j \pmod{q} \text{ and } \alpha_j \neq 0, \\ x_i^{h} , & \text{if } x_i^{\ell} \equiv j \pmod{q} \text{ and } \alpha_j = 0. \end{cases}$$

Moreover, If $z_i^{(\ell, h)} = x_i^{\ell} + \alpha_j = 0$, then $z_i^{(\ell, h)} = \alpha_j$. It is not hard to see that if $(x_1^1)$ and $(x_1^2)$ are feasible solutions, then these two trial solutions are also feasible solutions.

For two solutions $(x_1^1)$ and $(x_2^1)$ in RefSet, we generate four solutions by applying $T_1(1, 2)$, $T_2(1, 2)$, $T_1(2, 1)$ and $T_2(2, 1)$. If $(x_1^1)$ and $(x_2^2)$ belong to RefSet_1 (RefSet_2), we select the parameter $\text{prob}$ equal to 0.55 ($\text{prob} = 0.45$). Finally, if neither $(x_1^1)$ nor $(x_2^2)$ is a member of RefSet_1, we take $\text{prob} = 0.5$.

Trial solutions that are constructed as combination of the reference set are placed in a solution pool, denoted by Pool. Then the solutions in the set Pool are evaluated and the best $b_1$ solutions are chosen from the union of the current RefSet_1 and the set Pool. The set RefSet_2 is updated in the same way as the set RefSet_1, but according to the dispersion, defined by (11), among solutions.

5 Tabu Search

The tabu search method is an iterative heuristic technique used for finding, in a set $X$ of feasible solutions, the solution that minimizes an objective function $f$ based on neighborhood search (NS).

In neighborhood search, each feasible solution $x$ has an associated set of neighbors, $N(x) \subset X$, called the neighborhood of $x$. It starts with a given initial feasible solution and checks the space $X$ by moving from one solution to another one in its neighborhood. At each iteration of the process, a subset $V$ of $N(x)$ is generated and we move from the current solution $x$ to the best one $x^*$ in $V$, whether or not $f(x^*)$ is better than $f(x)$.

In some optimization problems, including the problem at hand, when exploring the set $V$ we find that there are multiple optimal solutions. Then, an important feature in these problems is to make a random choice of the best solution $x^*$ of $V$. However, the main shortcoming of this simple heuristic optimization algorithm is by cycling.

Stopping rules must also be defined. In many cases (including our optimization problem) a lower bound $f^*$ of the objective function is known in advance. As soon as we have reached this bound, we may interrupt the algorithm. In general, however, $f^*$ is not available with sufficient accuracy; so a fixed maximum number of iterations is given.

The tabu search algorithm offers another interesting possibility for overcom-
ing the above-mentioned obstacle of the NS technique. To prevent cycling, a queue called the tabu list $T$ of length $|T| = t$ is provided. Its aim is to forbid moves between solutions that reinstate certain attributes of past solutions. After $t$ iterations they are removed from the list and are free to be reinstated. The tabu list is also called a short term memory function, because it stores information on the $t$ most recent moves. Likewise, a long-term memory is used to improve the search process as we shall see in the adaptation of this technique. This memory is used to diversify the search and move to unexplored regions, and is usually based on a frequency criterion.

Unfortunately, the tabu list may forbid certain interesting moves, such as those that lead to a better solution than the best one found so far. An aspiration criterion is introduced to cancel the tabu status of a move when this move is judged useful.

We now describe how we used tabu search to find 1-rotational near resolvable difference families.

The feasible solutions and the objective function were defined in Section 3.

The neighborhood structure is defined using swap moves: Two 1-rotational families $P$ and $P'$ are defined as neighbors if one of them is obtained by swapping two elements $x$ and $y$, with $x \equiv y \pmod{q}$, from different blocks. Note that the move that transforms $P$ into $P'$ is defined by the vector $(x, y)$.

The tabu list $T$ of length $|T|$ is constructed and updated circularly during the process. At each iteration we introduce the best found move $(x, y)$ into the tabu list. This means that for $|T|$ iterations of TS, $x$ and $y$ are points that cannot be exchanged from blocks. The length of the tabu list was adjusted experimentally depending on the 1-rotational difference family parameters. The best tabu length seems to be integers somewhere in the range 3 through 7.

The process stops if the objective function has reached the lower bound 0. However, since TS is a heuristic technique, it does not always guarantee reaching the theoretical minimum, and the search process will be stopped if the number of iterations used without improving the best solution is greater than a nimax = 850 limit.

The long term memory is a function that records moves taken in the past in order to penalize those which are non-improving. The goal is to diversify the search by compelling regions to be visited that possibly were not explored before [Glover 1989]. In our particular TS implementation, the long-term memory is a $q$ vector, which will be denoted $F$. This vector has zeroes at the beginning of the procedure. When a pair of points $(x, y)$ are swapped at a given iteration, the vector changes as follows: $F_g = F_g + q$, where $g = x \pmod{q}$ and $0 \leq g \leq q - 1$. (The entry $F_g$ is the frequency at which two points belong to set $N + g$ have been swapped). Then, the values of non-improving moves that switch two points in $N + g$ are increased by $F_g$. 

10
6 Results

The SS algorithm described above was implemented in C language and all computations were executed on a Dual64 computer with 2GHz. We consider the following set of parameters for our method for all runs: \( P_{\text{obs}ize} = 100, \ |RefSet| = 32, \ |RefSet_1| = 16 \) and \( |RefSet_2| = 16. \)

Our SS was used to construct 1-rotational \((v, k, k-1)\)-NRBDs on the set problem of designs with parameters \(22 \leq v \leq 55\) and \(7 \leq k \leq 10.\) Fortunately, the technique was able to construct all of them. However, we present here only the results on the three NRBDs whose existence remained unknown so far.

A \((45, 9, 8)\)-NRDF was found:

\[
\begin{align*}
D_1 &= \{4, 7, 13, 20, 25, 29, 35, 37, \infty\} \\
D_2 &= \{3, 5, 10, 24, 28, 31, 32, 34, 44\} \\
D_3 &= \{2, 11, 17, 21, 22, 23, 30, 36, 39\} \\
D_4 &= \{1, 12, 14, 15, 18, 19, 26, 27, 41\} \\
D_5 &= \{6, 8, 9, 16, 33, 38, 40, 42, 43\}
\end{align*}
\]

This and Theorem 2 prove the existence of a \((46, 9, 8)\)-NRBD. For these difference family parameters, an optimal solution was found after 30 runs.

A \((50, 10, 9)\)-NRDF was found:

\[
\begin{align*}
D_1 &= \{5, 7, 9, 15, 24, 26, 27, 38, 48, \infty\} \\
D_2 &= \{1, 3, 21, 25, 30, 33, 37, 40, 45, 46\} \\
D_3 &= \{10, 13, 14, 17, 20, 35, 36, 41, 47, 49\} \\
D_4 &= \{2, 6, 11, 12, 19, 28, 39, 42, 43, 44\} \\
D_5 &= \{4, 8, 16, 18, 22, 23, 29, 31, 32, 34\}
\end{align*}
\]

This and Theorem 2 prove the existence of a \((51, 10, 9)\)-NRBD. For these difference family parameters, an optimal solution was found after 200 runs.

A \((54, 9, 8)\)-NRDF was found:

\[
\begin{align*}
D_1 &= \{5, 15, 16, 17, 30, 40, 41, 48, \infty\} \\
D_2 &= \{3, 12, 19, 21, 22, 26, 36, 52, 53\} \\
D_3 &= \{2, 6, 8, 14, 31, 34, 35, 43, 46\} \\
D_4 &= \{1, 7, 10, 13, 18, 23, 25, 27, 33\} \\
D_5 &= \{9, 20, 24, 28, 37, 39, 42, 44, 47\} \\
D_6 &= \{4, 11, 29, 32, 38, 45, 49, 50, 51\}
\end{align*}
\]

This and Theorem 2 prove the existence of a \((55, 9, 8)\)-NRBD. For these difference family parameters, an optimal solution was found after 200 runs.
7 Conclusions

In this paper, we have formulated the problem for constructing near resolvable 1-rotational difference families as a discrete optimization problem, where 0 is the global minimum. Our new definition of feasible solutions allows us to reduce the search space for this optimization problem. First, we proposed an algorithm based on tabu search to tackle the problem. TS was able to construct many 1-rotational \((qk + 1, k, k - 1)\)-NRDFs for \(k \leq 8\) and \(q \leq 5\). However, the tabu search algorithm did not produce the theoretical optimal solution for the near resolvable difference family parameters \((46,9,8)\), \((51,10,9)\) and \((55,9,8)\). Then, to solve these NRBDFs, we developed an algorithm based on scatter search where the tabu search algorithm was used as the Improvement Method for SS algorithm. This conjunction of TS and SS algorithms was able to find theoretical optimal solution for these 1-rotational near resolvable difference family parameters.

Acknowledgment

Support for the research was provided by the grant IN119906-2 from PAPIIT-DGAPA-UNAM.

References