Journal of Statistical Software

August 2015, Volume 66, Issue 2. *http://www.jstatsoft.org/*

OptGS: An R Package for Finding Near-Optimal Group-Sequential Designs

James M. S. Wason
MRC Biostatistics Unit

Abstract

A group-sequential clinical trial design is one in which interim analyses of the data are conducted after groups of patients are recruited. After each interim analysis, the trial may stop early if the evidence so far shows the new treatment is particularly effective or ineffective. Such designs are ethical and cost-effective, and so are of great interest in practice. An optimal group-sequential design is one which controls the type-I error rate and power at a specified level, but minimizes the expected sample size of the trial when the true treatment effect is equal to some specified value. Searching for an optimal group-sequential design is a significant computational challenge because of the high number of parameters. In this paper the R package **OptGS** is described. Package **OptGS** searches for near-optimal and balanced (i.e., one which balances more than one optimality criterion) group-sequential designs for randomized controlled trials with normally distributed outcomes. Package **OptGS** uses a two-parameter family of functions to determine the stopping boundaries, which improves the speed of the search process whilst still allowing flexibility in the possible shape of stopping boundaries. The resulting package allows optimal designs to be found in a matter of seconds – much faster than a previous approach.

Keywords: group-sequential designs, optimal design, R.

1. Introduction

Traditionally a clinical trial is conducted by recruiting a pre-specified number of patients and then conducting a statistical test of some null hypothesis at the end of the trial. An alternative approach is to use a group-sequential design, in which the hypothesis is tested multiple times during accrual of patients. The advantage of a group-sequential design is that the trial may be stopped early if the data at a given interim analysis is sufficiently convincing to reject the null hypothesis or if the data suggests the experiment is unlikely to end in success (i.e., for futility). This possibility of stopping early will mean the expected sample size, that is the average sample size used if the trial were repeated many times, is lower. This is at the expense of a higher maximum sample size of the trial (i.e., the sample size used if the trial continues to the final analysis) compared to the fixed sample size trial.

Group-sequential designs have been widely researched and used for clinical trials. Their use is arguably more ethical, since trials will be stopped early if the new treatment is ineffective. They are also more efficient than fixed sample size trials, as fewer patients are used on average – this means the average cost of a trial is lower, and more trials can be supported from a limited number of patients. Group-sequential designs have also been used in other applications, for example acceptance sampling in quality control. Jennison and Turnbull (2000) provide an excellent overview of group-sequential methodology.

When multiple analyses are allowed, the number of parameters (i.e., the group size and stopping boundaries at each stage) is higher than the number of constraints set by the required type-I error rate and power. The traditional approach of designing a group-sequential trial is to constrain the stopping boundaries of the trial using some function, and thus reducing the number of parameters to equal the number of constraints. Commonly used stopping boundaries are those of Pocock (1977), O'Brien and Flemming (1979), and Whitehead and Stratton (1983).

An alternative approach to using a stopping boundary function is to choose the boundaries so that the design minimizes the expected sample size at some treatment effect, i.e., an optimal design. An optimal design is feasible, i.e., it meets required constraints on the type-I error rate and power, and also optimizes some specified criterion over all possible feasible designs. A commonly used example of such a criterion is the expected sample size under the null hypothesis. A method for finding optimal group-sequential designs using dynamic programming was proposed by Eales and Jennison (1992). The method cannot be used for all optimality criteria of interest, for example the maximum expected sample size; an alternative approach using simulated annealing was proposed to search for more general optimal designs (Wason, Mander, and Thompson 2012). The disadvantage of simulated annealing is that it is computationally intensive, and therefore may be difficult without substantial computational resources. A less computational demanding method of searching for optimal designs which retains the flexibility of simulated annealing is highly desirable.

The search process for an optimal design is time consuming because of the high number of parameters. Intermediate between a fixed group-sequential stopping boundary design and an optimal design is the power family of group-sequential tests (Emerson and Flemming 1989; Pampallona and Tsiatis 1994), in which the shape of the stopping boundaries is controlled by a single parameter. This is more complex than a fixed design, as the extra parameter allows infinitely many shapes. It is also less complex than finding an optimal design as the number of parameters to search over is reduced. In this paper I propose using a two-parameter stopping boundary function which provides greater flexibility in possible shapes. Using this function requires considerably less computation in comparison to finding an optimal design using simulated annealing and the resulting designs are very similar to optimal designs. The method is implemented in the R (R Core Team 2015) package **OptGS** (Wason 2015), which is available from the Comprehensive R Archive Network (CRAN) at `http://CRAN.R-project.org/package=OptGS` and which can be run on a single processor in a matter of seconds.

2. Methods

2.1. Notation

Consider a randomized two-arm group-sequential design with up to J analyses. The jth analysis takes place after $n_j = jn_1$ patients have been randomized to each arm, and their treatment response has been measured. Here, n_1 is called the group size. Outcomes are assumed to be normally distributed with known variance σ^2. The case of unknown variance will be addressed later in this paper. The mean difference in response between the treatment arm and control arm is labeled δ, with the hypothesis tested being $H_0 : \delta \leq \delta_0$. A design is required such that the probability of rejecting the null is at most α under H_0 and at least $1-\beta$ when $\delta \geq \delta_1$, where δ_1 is the clinically relevant difference (CRD). These two constraints are referred to as the type-I error and power constraints respectively. A design which meets both constraints is called feasible.

At a given interim analysis j, the z-statistic is calculated:

$$Z_j = \frac{\sqrt{n_j}\hat{\delta}_j}{\sqrt{2\sigma^2}}, \tag{1}$$

where $\hat{\delta}_j$ is the MLE of δ using data observed up to stage j. If $Z_j > e_j$, the trial stops for efficacy; if $Z_j \leq f_j$, the trial stops for futility. If it is between the two thresholds, the trial continues to stage $j+1$. The value of e_J is set to f_J to ensure that a decision is made at the last interim analysis.

2.2. Power family of group-sequential tests

The power family of group-sequential tests was proposed by Emerson and Flemming (1989) for symmetric tests (i.e., ones where $\alpha = \beta$). Pampallona and Tsiatis (1994) extended the family to allow non-symmetric tests ($\alpha \neq \beta$). The family is indexed by a parameter Δ, which allows the shape of the stopping boundaries to vary. In the notation above, the power family stopping boundaries are:

$$\begin{aligned} e_j &= C_e(J,\alpha,\beta,\Delta)(j/J)^{\Delta-0.5}, \\ f_j &= \delta\sqrt{\mathcal{I}_j} - C_f(J,\alpha,\beta,\Delta)(j/J)^{\Delta-0.5}, \end{aligned}$$

where $\mathcal{I}_j = 2n_j/\sigma^2$.

To ensure $e_J = f_J$, the final information level is set to:

$$\mathcal{I}_J = 2n_J/\sigma^2 = \frac{\{C_e(J,\alpha,\beta,\Delta) + C_f(J,\alpha,\beta,\Delta)\}^2}{\delta^2}. \tag{2}$$

For a specific value of Δ, values of $C_f(J,\alpha,\beta,\Delta)$ and $C_e(J,\alpha,\beta,\Delta)$ are found so that the design has correct type-I error rate and power. The value of Δ can be varied, with higher values generally giving designs with lower expected sample sizes, but higher maximum sample sizes.

2.3. Extended power family of group-sequential tests

Although the power family provides greater flexibility than fixed group-sequential designs such as those of Pocock (1977) or O'Brien and Flemming (1979), it does not provide sufficient flexibility to include optimal designs. For optimal designs, the shape of the efficacy stopping boundaries will generally differ from the shape of the futility stopping boundaries.

I propose a straightforward extension to the power family: introducing two shape parameters Δ_f and Δ_e so that the shape of the futility and efficacy boundaries may differ, and thus allow a greater flexibility in shape. The stopping boundaries will be:

$$\begin{aligned} e_j &= C_e(J,\alpha,\beta,\Delta_f,\Delta_e)(j/J)^{\Delta_e-0.5}, \\ f_j &= \delta\sqrt{\mathcal{I}_j} - C_f(J,\alpha,\beta,\Delta_f,\Delta_e)(j/J)^{\Delta_f-0.5}. \end{aligned}$$

Note that Equation 2 still ensures $e_J = f_J$.

2.4. Finding optimal extended power family designs

Given values of $(\Delta_f, \Delta_e, C_f, C_e)$, one can straightforwardly find the probabilities of stopping at each interim analysis, given a specified value for δ, by using the methods given in Chapter 19 of Jennison and Turnbull (2000). This allows one to calculate the type-I error rate, power, and expected sample size at any treatment effect of interest. One can also find the maximum expected sample size using an interval search technique, as described in Wason *et al.* (2012).

As in Pampallona and Tsiatis (1994), for each value of (Δ_f, Δ_e), values of C_f and C_e are required such that the design has correct type-I error rate and power. These values can be found by searching for the values of (C_f, C_e) that minimize the following function:

$$(\alpha^*(J,\Delta_f,\Delta_e,C_f,C_e)-\alpha)^2 + (\beta^*(J,\Delta_f,\Delta_e,C_f,C_e,\delta)-\beta)^2, \tag{3}$$

where $\alpha^*(.)$ and $\beta^*(.)$ are the type-I and type-II error rate for a specific design. The value of Equation 3 is 0 if and only if the type-I error rate and power of the design are as required.

In package **OptGS**, this minimization is performed using the Nelder-Mead algorithm (Nelder and Mead 1965), calling C++ code written by Burkardt (2008). A drawback of the algorithm is that it is not guaranteed to reach the global minimum. To overcome this, it can be repeatedly run using different starting values until values of C_f and C_e are found which give the correct type-I and type-II error rates.

The Nelder-Mead algorithm is also used to search for the optimal design over values of (Δ_f, Δ_e). Almost surely, the optimal value of (Δ_f, Δ_e) will imply a non-integer group size. Thus, a second optimization is run, with the constraint that the final group size is equal to the ceiling integer of that implied by the optimal (Δ_f, Δ_e). A third optimization is run using the floor integer instead of the ceiling. Of the designs found in the second and third runs, the one that is closer to optimal is picked as the final design.

Since in the second and third optimizations the group size is constrained to be a given value, C_e is determined from C_f by Equation 2. The function to be minimized is:

$$h(\Delta_f,\delta_e,C_f,n_j,J) + \nu\left\{(\alpha^*(J,\Delta_f,\Delta_e,C_f,n_j)-\alpha)^2 + (\beta^*(J,\Delta_f,\Delta_e,C_f,n_j,\delta)-\beta)^2\right\}, \tag{4}$$

where h is the optimality criterion of interest, and ν is a penalty factor to ensure the final design has the correct type-I and type-II error rates. In effect this step is to tweak the stopping boundaries slightly so that the error rate constraints are met for integer group size.

Package **OptGS** allows the user to specify one of three optimal designs to search for: 1) the null-optimal design, which minimizes $\mathbb{E}(N|\delta_0)$; 2) the CRD-optimal design, which minimizes $\mathbb{E}(N|\delta_1)$; or 3) the δ-minimax design, which minimizes $\max(\mathbb{E}(N))$.

2.5. Finding balanced designs

In Section 2.4, a single optimality criterion was of interest. Previous work has shown that if a design focuses on a single optimality criterion, the resulting design often performs poorly in terms of other criteria that may also be of interest (Jung, Lee, Kim, and George 2004; Wason *et al.* 2012). For example, all designs that are optimal for the expected sample size at some δ have a high maximum sample size. An alternative approach is to find a balanced design in which the design is chosen in order to minimize the weighted sum of two or more criteria of interest.

Package **OptGS** allows the user to find a design that balances the three optimality criteria of interest together with the maximum sample size. A vector of weights, $(\omega_1, \omega_2, \omega_3, \omega_4)$, is specified such that all entries are non-negative. Then the feasible design is found that minimizes the following function:

$$\omega_1 \mathbb{E}(N|\delta = \delta_0) + \omega_2 \mathbb{E}(N|\delta = \delta_1) + \omega_3 \max(\mathbb{E}(N)) + \omega_4 J n_1. \tag{5}$$

This design balances the three optimality criteria together with the maximum sample size. Note that one of ω_1, ω_2, and ω_3 must be strictly positive, because an infinite number of designs will exist with the lowest maximum sample size.

2.6. Unknown variance

The literature on group-sequential designs for normally distributed endpoints generally assumes known variance, but in practice this is a strong assumption to make. Once boundaries for known variance statistics are found, one can convert them to boundaries for t-test statistics using a recursive algorithm (Jennison and Turnbull 1991), or a more straightforward quantile substitution method (Whitehead, Valdes-Marquez, and Lissmats 2009). The latter method generally controls the type-I error rate at the correct level (Wason *et al.* 2012). Package **OptGS** provides the option of returning stopping boundaries for the unknown variance case using quantile substitution. A value of the treatment outcome standard deviation must still be given in order to power the trial. Note that if the true variance differs from the design value, the power of the design will vary from the required level.

3. Use of package OptGS

Package **OptGS** contains a single function, `optgs`. All arguments are fully documented within the package. The `weights` argument allows the user to put weights on the different optimality criteria. It is a four entry vector where the first entry is the weight put on $\mathbb{E}(N|\delta = \delta_0)$; the second entry is the weight put on $\mathbb{E}(N|\delta = \delta_1)$; the third entry is the weight put on $\max(\mathbb{E}(N))$; and the fourth entry is the weight put on the maximum sample size. As an example, here is the syntax used to find a four-stage δ-minimax design with type-I error rate 0.05, power 0.9 with a standardized clinically relevant difference (i.e., $\frac{\delta_1 - \delta_0}{\sigma}$) of 1/3:

```
R> optgs(J = 4, alpha = 0.05, power = 0.9, delta0 = 0, delta1 = 1,
+    sigma = 3, weights = c(0, 0, 1, 0))

Groupsize:  50
Futility boundaries  -0.26 0.65 1.29 1.82
```

```
Efficacy boundaries  2.32 2.05 1.91 1.82
ESS at null:     92.9
ESS at CRD:      105
Maximum ESS:     122.1
Max sample-size: 200
```

The output gives the design and operating characteristics. The sample size required per arm per stage is 50, the futility boundaries are $(-0.2644, 0.6469, 1.2940, 1.8189)$, and the efficacy boundaries are $(2.3188, 2.0537, 1.9129, 1.8189)$. If the user preferred a design that puts weight on the maximum sample size, then they could change the `weights` argument. For example:

```
R> optgs(J = 4, alpha = 0.05, power = 0.9, delta0 = 0, delta1 = 1,
+    sigma = 3, weights = c(0, 0, 0.75, 0.25))

Groupsize:  43
Futility boundaries  -0.95 0.32 1.1 1.69
Efficacy boundaries  3.22 2.33 1.93 1.69
ESS at null:     98.2
ESS at CRD:      112.7
Maximum ESS:     125.7
Max sample-size: 172
```

Notice that the expected sample sizes have risen compared to the first design. The group size has fallen from 50 to 43, which would considerably reduce the maximum sample size of the design. By varying the `weights` argument, the user can search for designs which put different weights on the maximum sample size and relevant expected sample size. It is recommended that some weight is always put on the maximum sample size, as a small value can reduce the maximum sample size noticeably without increasing the expected sample size more than a negligible amount.

Values for all entries in the `weights` argument can be provided. As an example, firstly the $(1, 1, 1, 1)$-balanced design is found, and then a design that puts more weight on the expected sample size at $\delta = \delta_0$ and less on the expected sample size at $\delta = \delta_1$.

```
R> optgs (J = 4, alpha = 0.05, power = 0.9, delta0 = 0, delta1 = 1,
+    sigma = 3, weights = c(1, 1, 1, 1))

Groupsize:  42
Futility boundaries  -1.22 0.21 1.05 1.68
Efficacy boundaries  3.53 2.43 1.96 1.68
ESS at null:     100.9
ESS at CRD:      114.6
Maximum ESS:     127.2
Max sample-size: 168

R> optgs(J = 4, alpha = 0.05, power = 0.9, delta0 = 0, delta1 = 1,
+    sigma = 3, weights = c(2, 0.5, 1, 1))
```

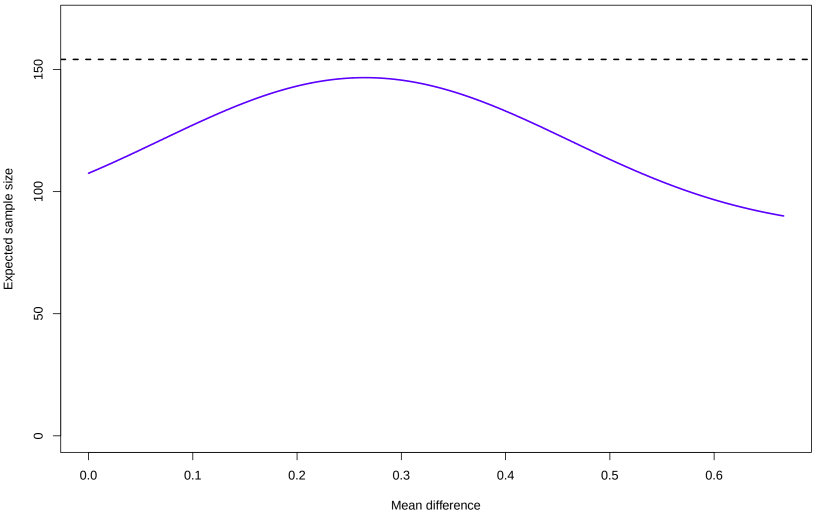

Figure 1: Output from `plot(optgs())`.

```
Groupsize:  43
Futility boundaries  -0.69 0.37 1.08 1.62
Efficacy boundaries  4.17 2.6 1.97 1.62
ESS at null:     94.4
ESS at CRD:      119.2
Maximum ESS:     128.1
Max sample-size: 172
```

Note that the second design has lower expected sample size at $\delta = \delta_0$ and higher expected sample size at $\delta = \delta_1$, as one might expect. Although the weight put on the maximum expected sample size and maximum sample size have not changed, the group size and maximum expected sample size of the second design are different to the first design. This is to be expected, as varying one of the operating characteristics will have an effect on the others too.

The `plot` function can be used on an object containing the output of `optgs`. For example, `plot(optgs())` will give the plot shown in Figure 1.

The `sd.known` argument in `optgs` can be set to `FALSE`, in order to convert the stopping boundaries to unknown variance boundaries, as discussed in Section 2.6. For example:

```
R> optgs(J = 2, delta1 = 1, sigma = 1)

Groupsize:  9
Futility boundaries  0.4 1.6
Efficacy boundaries  3.17 1.6
ESS at null:     12.1
ESS at CRD:      16.3
Maximum ESS:     16.5
Max sample-size: 18

R> optgs(J = 2, delta1 = 1, sigma = 1, sd.known = FALSE)
```

	$\mathbb{E}(N\|\delta_0)$			Time taken	
J	Average from 10 SA runs	Minimum from 10 SA runs	From **OptGS**	Average SA run	**OptGS**
2	108.1	107.5	107.5	2.5s	0.50s
3	95.0	94.7	94.8	19.6s	1.06s
4	89.0	88.8	89.1	35.0s	1.71s
5	85.6	84.9	85.8	64.0s	4.62s

Table 1: Comparison of run-time and expected sample size at $\delta = \delta_0$ of designs found from simulated annealing (SA) and package **OptGS**.

```
Groupsize:  9
Futility boundaries  0.4 1.65
Efficacy boundaries  3.81 1.65
ESS at null:     12.1
ESS at CRD:      16.3
Maximum ESS:     16.5
Max sample-size: 18
```

Note that only the stopping boundaries have changed, and not the group size or expected sample sizes. These unchanged quantities still assume the variance is known – in practice the required group size may have to be increased in order to ensure the power constraint is correct. In addition, the expected sample sizes will differ when the standard deviation is estimated from the data.

4. Results

The closest equivalent to package **OptGS** is the simulated annealing method as discussed in Wason *et al.* (2012). Table 1 shows the time taken to find J-stage null-optimal designs using simulated annealing and using package **OptGS** (note that both methods require the final design to have an integer group size). Both methods were carried out on a single core of a Intel 3rd generation Core i7 processor. Because simulated annealing is a stochastic process, results may vary between runs. Therefore for each value of J, I carried out 10 independent simulated annealing searches. The average and minimum expected sample size under the null over the ten processes are shown in Table 1.

Interestingly, for most values of J, the optimal design found by package **OptGS** is close to the best of 10 runs of simulated annealing. This is despite the shape constraint imposed by use of the extended power family. Only for $J = 5$ does simulated annealing show some improvement over package **OptGS**. This could indicate that as J increases, the shape constraint has a larger effect. The process that package **OptGS** uses is substantially faster than even one simulated annealing run. All designs found by package **OptGS** met the type-I error rate and power constraints required. Clearly, there are substantial advantages to using package **OptGS** over simulated annealing.

Table 2 shows the optimal values of $\Delta_f, \Delta_e, C_f, C_e$ for the three types of optimal design implemented in package **OptGS** as well as the $(1,1,1,1)$-balanced design, i.e., the balanced design that puts equal weight on all four operating characteristics. Generally, it is clear that

Design	J	Δ_f	Δ_e	C_f	C_e	$\mathbb{E}(2N\|\delta_0)$	$\mathbb{E}(2N\|\delta_1)$	$\max(\mathbb{E}(2N))$	$\max(2N)$
Null-optimal	2	0.46	−0.39	1.57	1.50	215.0	290.8	297.2	340
	3	0.51	−0.46	1.53	1.66	189.5	272.4	280.0	366
	4	0.51	−0.35	1.53	1.73	178.1	257.7	269.4	384
	5	0.53	−0.46	1.51	1.83	171.7	262.0	271.9	400
CRD-optimal	2	−0.15	0.45	1.83	1.26	238.5	234.6	275.5	344
	3	−0.13	0.48	1.95	1.26	229.0	214.8	264.6	372
	4	−0.21	0.47	2.01	1.25	227.0	205.6	260.5	384
	5	0.13	0.47	2.04	1.38	195.6	199.8	245.6	420
δ-minimax	2	0.33	0.31	1.73	1.42	220.6	239.5	266.6	356
	3	0.36	0.34	1.79	1.50	196.3	219.5	251.9	390
	4	0.32	0.32	1.82	1.51	185.7	210.0	244.2	400
	5	0.31	0.32	1.84	1.53	179.8	204.2	239.4	410
Balanced	2	0.03	0.04	1.66	1.34	224.0	248.5	273.3	324
	3	0.00	−0.02	1.66	1.34	210.3	237.6	261.8	330
	4	0.00	−0.04	1.68	1.38	201.9	229.2	254.3	336
	5	0.13	−0.02	1.68	1.44	188.2	222.8	247.3	350

Table 2: Optimal design parameters $(\Delta_f, \Delta_e, C_f, C_e)$ for various optimality criteria and number of stages. "Balanced" corresponds to the $(1,1,1,1)$-balanced design. Note that the expected and maximum sample sizes shown are for both treatment arms.

allowing Δ_f to differ from Δ_e is necessary to allow optimal designs to be found – the null-optimal and CRD-optimal designs have Δ_f and Δ_e designs with opposite signs. Interestingly, a design close to the δ-minimax design would be found using the original power family, as Δ_f and Δ_e are very close in value.

The $(1,1,1,1)$-balanced design has good operating characteristics – none of the expected sample sizes are too large compared to those of the optimal designs, but the maximum sample size is generally substantially lower.

In some scenarios, it may not be desirable to stop a clinical trial early for efficacy. For example, if it is of interest to estimate the treatment effect with a high precision, or to gather additional information about side-effects. In these cases, one should not put weight on the expected sample size at the clinically relevant difference, or the maximum expected sample size. Instead, one might choose to just put weight on the expected sample size under the null hypothesis and the maximum sample size. This will lead to larger efficacy boundaries, and small probabilities of stopping early for efficacy.

A drawback of package **OptGS** is its reliance on the Nelder-Mead algorithm, for which results can be sensitive to the starting value of (Δ_f, Δ_e). For example, when looking for the four-stage δ-minimax design, starting the Nelder-Mead algorithm at $(0.3, 0.3)$ gives a design with maximum expected sample size 122.11, but starting it at $(-0.5, -0.5)$ gives a design with a maximum expected sample size of 126.73 – a considerable difference. Using the results in Table 2, default starting values for the Nelder-Mead algorithm were set to be $(0.4, -0.4)$ for the null-optimal design, $(-0.2, 0.4)$ for the CRD-optimal design, and $(0.3, 0.3)$ for the δ-minimax design. For balanced designs, the starting values used are the weighted sum of the three previous starting values (where the weights are ω_1, ω_2 and ω_3 from Equation 5), and $(-0.5, -0.5)$ times the weight put on the maximum sample size. I have found that this tends to perform well, but a user may want to change the starting values used before picking a design. An argument named `initial` allows the user to override the default starting values.

5. Discussion

Optimal and balanced group-sequential designs for continuous outcomes have been discussed in several papers (Eales and Jennison 1992; Brittain and Bailey 1993; Eales and Jennison 1995; Chang 1996; Barber and Jennison 2002; Anderson 2007; Wason and Mander 2012; Wason *et al.* 2012). Balanced designs have excellent operating characteristics and are a very flexible class of design. Up to now, they have not been used in practice because freely-available software to implement them has not yet been available. Package **OptGS** is a R package that provides an automated search procedure using an extension to the power family of group-sequential tests (Pampallona and Tsiatis 1994) to find optimal and balanced group-sequential designs. The extended power family has two parameters which determine the relative shape of the futility and efficacy stopping boundaries. Previous methods to find optimal designs include: a grid search technique, which is infeasible for more than two stages; a dynamic programming approach which can be used for certain optimality criteria, but not others, and also not for designs that balance different optimality criteria; and simulated annealing, which allows searching for all optimality and balancing criteria, but is slow. In comparison, package **OptGS** yields designs which are often better than those found from an average simulated annealing run and in a much faster time.

The R package **gsDesign** (Anderson 2014) implements a large range of group-sequential designs. One type of design implemented is the Hwang, Shih and DeCani error spending function (Hwang, Shih, and DeCani 1990), which has been used to find designs that are optimal with respect to the integral of the expected sample size over a normal distribution (Anderson 2007). Thus, package **gsDesign** could be modified to search for near-optimal designs. A current advantage of package **OptGS** is that the search procedure is automated, allowing a very flexible range of optimal and balanced designs to be found. It also allows stopping boundaries to be modified to take into account unknown variance. However, a disadvantage of the current version of **OptGS** is that the sample size at each interim analysis is assumed to be equally spaced, whereas package **gsDesign** allows the user to modify designs to take into account unequally spaced analyses. In practice, analyses are unlikely to be exactly evenly spaced, even if designed to be. Some patients may drop out of the trial, or practical considerations may have determined that the interim analysis must be at a certain time. Stopping boundaries can be modified to take into account the actual number of observations at a given analysis. Jennison and Turnbull (2000) describe a method to adapt stopping boundaries from the one-parameter power family to allow different numbers of patients at each analysis, which could be straightforwardly generalized to the extended power family. Fixed stopping boundaries from an optimal or balanced group-sequential design can be interpolated into an error spending function, as described by Kittelson and Emerson (1999). Both of these approaches control the overall type-I error, but not necessarily the power.

Package **OptGS** is currently just implemented for normally distributed outcomes, and cannot be used directly to find optimal designs for binary or time-to-event outcomes. However, typical test statistics for both types of outcome (such as the estimate of a binary proportion and the log-rank test) are both asymptotically normally distributed. Thus, the methods implemented in package **OptGS** could be extended to allow group-sequential designs for other endpoint types, although the operating characteristics would be valid only asymptotically. In the case of treatment outcomes that take a long time to observe, group-sequential designs are generally less useful. Since patients will generally be recruited continuously, by the time the treatment effect on the first group of patients has been observed, most patients in the trial will have been recruited. Thus, optimal or balanced designs are more relevant for shorter term endpoints, although with some modification, they may still be useful for optimizing the time taken by the trial when the endpoint is a long-term one.

A type of trial that has recently started to garner more attention and study is a multi-arm multi-stage (MAMS) trial (Sydes, Parmar, James *et al.* 2009; Magirr, Jaki, and Whitehead 2012). Using multiple new treatment arms in a trial increases efficiency over separately testing each treatment because just one control group is needed. It also means a direct comparison can be made which may be problematic when conducting several separate trials. Some work has been done on optimal multi-arm multi-stage clinical trials for normally distributed outcomes (Wason and Jaki 2012) which involves using simulated annealing. The methods underlying package **OptGS** could be extended to optimal design of group-sequential trials, although it may be that the extended power family is no longer sufficiently flexible to include optimal MAMS designs.

Acknowledgments

I would like to thank Dr. Thomas Jaki for his helpful comments on both the R package and the manuscript. In addition, I would like to thank the two reviewers for their constructive and insightful comments, which helped improve the manuscript. This work is funded by the UK Medical Research Council (grant number G0800860).

References

Anderson K (2014). ***gsDesign**: Group Sequential Design.* R package version 2.9-3, URL http://CRAN.R-project.org/package=gsDesign.

Anderson KM (2007). "Optimal Spending Functions for Asymmetric Group Sequential Designs." *Biometrical Journal*, **49**(3), 337–345.

Barber S, Jennison C (2002). "Optimal Asymmetric One-Sided Group Sequential Tests." *Biometrika*, **89**(1), 49–60.

Brittain EH, Bailey KR (1993). "Optimization of Multistage Testing Times and Critical Values in Clinical Trials." *Biometrics*, **49**(3), 763–772.

Burkardt J (2008). *C++ Implementation of Nelder-Mead Algorithm.* URL http://people.sc.fsu.edu/~jburkardt/cpp_src/asa047/asa047.html.

Chang MN (1996). "Optimal Designs for Group Sequential Clinical Trials." *Communications in Statistics – Theory and Methods*, **25**(2), 361–379.

Eales JD, Jennison C (1992). "An Improved Method for Deriving Optimal One-Sided Group Sequential Tests." *Biometrika*, **79**(1), 13–24.

Eales JD, Jennison C (1995). "Optimal Two-Sided Group Sequential Tests." *Sequential Analysis*, **14**(4), 273–286.

Emerson SS, Flemming TR (1989). "Symmetric Group Sequential Designs." *Biometrics*, **45**(3), 905–923.

Hwang IK, Shih WJ, DeCani JS (1990). "Group Sequential Designs Using a Family of Type I Error Probability Spending Fuctions." *Statistics in Medicine*, **9**(12), 1439–1445.

Jennison C, Turnbull BW (1991). "Exact Calculations for Sequential t, χ^2 and F Tests." *Biometrika*, **78**(1), 133–141.

Jennison C, Turnbull BW (2000). *Group Sequential Methods With Applications to Clinical Trials*. Chapman and Hall.

Jung SH, Lee T, Kim K, George SL (2004). "Admissible Two-Stage Designs for Phase II Cancer Clinical Trials." *Statistics in Medicine*, **23**(4), 561–569.

Kittelson JM, Emerson SS (1999). "A Unifying Family of Group Sequential Test Designs." *Biometrics*, **55**(3), 874–882.

Magirr D, Jaki T, Whitehead J (2012). "A Generalized Dunnett Test for Multiarm-Multistage Clincial Studies with Treatment Selection." **99**(2), 494–501.

Nelder JA, Mead R (1965). "A Simplex Method for Function Minimization." *The Computer Journal*, **7**(4), 308–313.

O'Brien PC, Flemming TR (1979). "A Multiple-Testing Procedure for Clinical Trials." *Biometrics*, **35**(3), 549–556.

Pampallona S, Tsiatis AA (1994). "Group Sequential Designs for One-Sided and Two-Sided Hypothesis Testing with Provision for Early Stopping in Favor of the Null Hypothesis." *Journal of Statistical Planning and Inference*, **42**(1–2), 19–35.

Pocock SJ (1977). "Group Sequential Methods in the Design and Analysis of Clinical Trials." *Biometrika*, **64**(2), 191–199.

R Core Team (2015). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/.

Sydes MR, Parmar MKB, James ND, *et al.* (2009). "Issues in Applying Multi-Arm Multi-Stage Methodology to a Clinical Trial in Prostate Cancer: The MRC STAMPEDE Trial." *Trials*, **10**(39).

Wason J (2015). ***OptGS**: Near-Optimal and Balanced Group-Sequential Designs for Clinical Trials with Continuous Outcomes*. R package version 1.1.1, URL http://CRAN.R-project.org/package=OptGS.

Wason JMS, Jaki T (2012). "Optimal Design of Multi-Arm Multi-Stage Trials." *Statistics in Medicine*, **31**(30), 4269–4279.

Wason JMS, Mander AP (2012). "Minimising the Maximum Expected Sample Size in Two-Stage Phase II Clinical Trials with Continuous Outcomes." *Journal of Biopharmaceutical Statistics*, **22**(4), 836–852.

Wason JMS, Mander AP, Thompson SG (2012). "Optimal Multi-Stage Designs for Randomised Clinical Trials with Continuous Outcomes." *Statistics in Medicine*, **31**(4), 301–312.

Whitehead J, Stratton I (1983). "Group Sequential Clinical Trials with Triangular Continuation Regions." *Biometrics*, **39**(1), 227–236.

Whitehead J, Valdes-Marquez E, Lissmats A (2009). "A Simple Two-Stage Design for Quantitative Responses with Application to a Study in Diabetic Neuropathic Pain." *Pharmaceutical Statistics*, **8**(2), 125–135.

Affiliation:

James Wason
MRC Biostatistics Unit
Institute of Public Health
Robinson Way
Cambridge, CB2 0SR, United Kingdom
E-mail: james.wason@mrc-bsu.cam.ac.uk
URL: http://sites.google.com/site/jmswason/

Journal of Statistical Software http://www.jstatsoft.org/
published by the American Statistical Association http://www.amstat.org/

Volume 66, Issue 2 *Submitted:* 2012-02-14
August 2015 *Accepted:* 2014-10-10