# 红包算法说明

## 随机生成本次红包总额M1

设总金额为M元，本次游戏难度系数为C，难度对应金额映射表F

* 通过难度系统C，查询F得到本次红包总额的最大最小浮动百分比
* 根据最大/最小百分比乘以M得到浮动的最大/最小值
* 随机取的最大/最小值得一个数值作为本次红包总额M1

## 根据红包总额，生成随机的红包列表

设红包总额红包M1，总个数N个

#### 数学解决方案

设每个红包平均金额为Z，则Z=M1/N

假设X为偏移量，则每个红包分布在 Z-X到Z+X之间，保证每个红包金额更均衡

那么，当前总金额为 (Z1+Z2+.....Zn) = M'

差值D = M' - M1

当D<0时，则对每个红包加 D/N的金额

当D=0时，红包金额不变

当D>0时，则对每个红包减去D/N的金额

所以，只要保证任意Zn大于D/N即能保证红包金额正确

公式:

当只有一个最小值红包，其余红包都为最大值时，计算偏移量X,即:

把Z=M1/N代入计算后得到公式:

#### 程序实现

1. 代入公式计算出偏移值X
2. 计算出平均值Z
3. 循环N次，每次随机数在(Z-X,Z+X)之间
4. 结束循环，计算出总金额M'
5. 计算需要补全值 Z' = (M'-M)/N,值可能为负数
6. 循环每个红包值，执行 Zn-Z'后截取小数点后前两位，得到新的红包值

## 生成本次领取红包具体数额

设随机生成的红包列表为L，用户接收红包个数R，微信基本红包数额为B(当前B=1)

* 取L的前R个红包作为红包具体数额列表L1
* 检车L1总和是否小于B
* 如果小于B，则计算B-L1得到差额D
* 将D添加到L1的第一个红包数额上