

# 基于遗传算法的手机键盘字母布局的研究

陈彩云 李治国

(南开大学组合数学研究中心,天津 300071)

E-mail: chencaiyn@eyou.com

**摘要** 该文探讨了手机键盘上的拼音字母的布局问题,给出了此问题的数学模型。运用遗传算法将手机键盘上的拼音字母进行重组,在对常用汉字进行模拟的过程中采用了多种优化策略,不断地对遗传算法进行优化,最后得到了理想的键盘字母布局。实验表明,该种字母布局大大提高了拼音输入法的效率,方便了用户的使用。

**关键词** 手机 字母布局 遗传算法 组合优化

文章编号 1002-8331-(2003)27-0077-03 文献标识码 A 中图分类号 TP391

## Rearranging of characters on Cell Phone Based on Genetic Algorithm

Chen Caiyun Li Zhiguo

(Center for Combinatorics of Nankai University, Tianjin 300071)

**Abstract:** This paper discusses the characters layout on the cell phone, builds up a mathematical model for the problem and then rearranges the characters by the Genetic Algorithm. During the simulation of words in common use, authors constantly optimize Genetic Algorithm by adopting some optimum strategy, the result is proved efficient in Chinese inputting.

**Keywords:** Cell Phone, Characters Layout, Genetic Algorithm, Combination Optimization

### 1 介绍

目前许多信息技术产品,如手机已实现了数字键盘的汉字输入功能。而且手机作为人们日常通信联系的重要工具之一,越来越受到人们的喜爱。据中国信息产业部公布,到2002年4月末移动电话用户已增至1亿519.8万户。在各种输入法中,拼音法由于其简单易学上手能用,而成为使用最广的输入法。现在在手机上使用得比较多的是智能拼音输入法,这种输入法不用学习,而且比传统的拼音输入法要快很多。虽然如此,由于目前手机键盘上面拼音字母的排列顺序,使得这种输入法没有发挥出更大的效率,经常是许多很常用的汉字,还得经过多次翻页才能找到。如果能够合理地安排数字键盘上面拼音字母的布局,或许可以较好地解决这个问题。目前还没有专门的文献研究这一方面的工作,该文初步探索了手机键盘上的拼音字母重组问题,取得了较好的结果。在实际模拟中得到,用改进后的字母布局进行汉字输入,翻页的次数比原来减少了36%之多,特别是一些常用字的输入,一般不需要翻页就能找到,大大提高了输入的效率。

以三星N628手机的智能输入法为例,该输入法以拼音字母所在的数字键来代表拼音进行输入。比如要输入“李”字,需键入拼音li,因为l在5号数字键上,i在4号数字键上,所以此时按下的键的顺序为5、4。但是与字母l同在5号键上的还有字母j,k,与字母i同在4号键上的还有字母g,h,因此,如果依次按下5、4号键,还会产生拼音ji,从而所有满足ji和li的汉字都会被选择出来,由于手机的显示屏的限制(三星N628手机每屏只能显示5个汉字),可能还需要多次翻页才能找到所需要的字,这样汉字输入的速度自然就很慢。而该文是将手机

键盘上的拼音字母进行适当的组合排列,尽可能地减少输入汉字时翻页的次数,这样能够大大提高手机拼音输入法的速度,同时也减少了输入时的错误。事实上,要减少翻页的次数就必须把这样的一种键盘输入对应多种不同的拼音拼法的数量减少到最小,也就是说要让键盘的输入与一种拼音拼法一一对应的数量达到最大,这样才能有效地减少翻页的次数,提高速度。

### 2 建立数学模型

记 $A$ 为一个字母集 $A=\{a..z\}$ ,在这里 $A$ 是26个字母的集合。 $W$ 为一个汉字拼音的集合, $W=\{\alpha_1\alpha_2\dots\alpha_m|\alpha_j\in A,j=1\dots m,m=1,2,\dots\}$ ,其中 $|W|<\infty$ ,且 $W$ 中的元素互不相同。在这里 $W$ 即是所有汉字拼音的集合,例如ai jian等等都是 $W$ 的元素。 $W$ 中每个元素都有一个权重,称之为字频,字频越大说明这个汉字拼音越常用。文中用一个字频函数 $h(\omega)$ 来表示每一个 $\omega\in W$ 的权重。

关于字频的计算,笔者搜集了含有244818个汉字的文本,粗略统计了每个汉字出现的概率,比如说“的”字出现的概率是3.095%;“是”字出现的概率为1.659%,再把相同拼音的汉字的概率加起来,得到“shi”出现的概率为4.10;“de”出现的概率为3.55%等等,那么记“shi”的字频 $h(\text{'shi'})=4.10$ ;“de”的字频 $h(\text{'de'})=3.55$ ,这样就可以得到每个拼音的字频了。

设 $D$ 为一个2到9的数字集合, $D=\{2..9\}$ , $D$ 中的元素就是目前手机键盘上面用来放字母的数字键。笔者的目的是要找到一个从 $A$ 到 $D$ 的映射 $f:A\rightarrow D$ ,满足如下条件:任给 $W$ 中的一个元素 $\omega$ ,那么 $\omega$ 可以写成 $\alpha_1\alpha_2\dots\alpha_m$ 的形式,有:

基金项目:教育部核心数学与组合数学重点实验室和国家973重点基础研究发展规划项目(编号:G19980306)资助

作者简介:陈彩云(1975-),女,博士生,研究方向:组合数学与数据挖掘。李治国(1977-),男,硕士生,研究方向:组合数学与应用。

$$f(\omega) = f(\alpha_{i_1} \alpha_{i_2} \dots \alpha_{i_n}) = f(\alpha_{i_1}) \times f(\alpha_{i_2}) \dots f(\alpha_{i_n})$$

由  $W$  中所有元素的像集组成了一个可重集合  $W'$ ,  $W'$  中相同元素出现的次数这里称为该元素在  $W'$  中的频率, 再将频率为 1 的元素重新组合成集合  $W'_1$ , 定义  $W'_1$  和  $W'$  中元素的字频比为:

$$R(W') = \frac{\sum_{\omega_i \in W'_1} h(\omega_i)}{\sum_{\omega_i \in W'} h(\omega_i)}$$

显然  $R(W') \in [0, 1]$ .

作者的目的是要找到使得  $R(W')$  尽可能大的映射  $f$ , 这样就可以做到最常用的字输入最快, 从而最大程度地减少翻页次数, 提高汉字输入速度。

由上面的分析可以知道, 大约有  $10^{21}$  多种不同的拼音字母的布局, 所以单纯的遍历算法是不现实的。于是想到用遗传算法<sup>[1]</sup>来解决该组合优化问题。

### 3 手机键盘上拼音字母布局问题的遗传算法设计

遗传算法是在 70 年代初期由美国 Michigan 大学的 Holland 教授最先提出的<sup>[2]</sup>, 后经由 DeJong(1975), Goldberg(1989) 等归纳总结所形成的一类模拟进化算法。它是一种借鉴生物界自然选择和自然遗传学机理上的迭代自适应概率性搜索算法。它是基于生物的进化与选择机制的优化算法, 通过维持一个群体, 并按个体的适应度大小重复地进行选择、交叉、变异等遗传操作来实现群体内个体结构的重组, 将性能良好的解结构遗传下去, 提高后代的适应能力, 从而进化到最优解或次优解。

遗传算法将问题的解表达成染色体(chromosome), 通过对染色体的选择(selection)、交叉(crossover)及变异(mutation)的操作, 产生更适应环境(即问题)的新染色体, 这样一代代地进化下去, 直到满足某些约束为止。它主要应用来解决复杂的组合优化问题, 是具有通用性的全局最优算法。

下面根据遗传算法来设计手机键盘上拼音字母的布局。

遗传算法的定义:

基本遗传算法可以定义一个 8 元组  $(C, E, P_0, M, \Phi, \Gamma, \Psi, Y)$

式中:

$C$ -个体的编码方法;

$E$ -个体适应度的评价函数;

$P_0$ -初始群体;

$M$ -种群大小;

$\Phi$ -选择算子;

$\Gamma$ -交叉算子;

$\Psi$ -变异算子;

$Y$ -遗传运算终止条件。

下面开始遗传算法的设计。

第一步: 设计编码方案(编码长度记为  $L$ )

在遗传算法中编码是解决问题的先决条件, 也是 GA 设计的关键步骤。该文要把 26 个字母映射到 8 个数字上面, 可以用一个编码长度  $L$  为 78 位的二进制数来表示一个染色体。在该染色体中, 每三位二进制数的值依次表示字母 a, b, ..., z 对应的数字, 如果三位数字为 000, 则表示在 2 号键上, 001 表示在 3 号键上, 依次类推。例如, 一个染色体为 001110101..., 可知, 这个映射把字母 a 映射到 3 号键上, 字母 b 映射到 7 号键上,

字母 c 映射到 5 号键上等等。可见, 每一个这样的染色体都对应着一个映射  $f$ 。

第二步: 确定种群规模  $M$

种群规模  $M$  取为 500, 根据经验知道规模太小不利于寻求最优解, 而规模过大会影响速度, 这里取为 500 虽然运行时间稍长, 但是它有利于更好地发现最优解, 而且也会避免早熟。初始种群  $P_0$ , 随机生成初始染色体  $\vec{X}(0) = (X_1(0), X_2(0), \dots, X_M(0)) \in H_M^L$ 。

第三步: 选取交叉算子与变异算子

交叉算子体现了遗传算法的局部搜索性能, 交叉概率( $P_c$ )的最佳范围为 0.4-0.7。过大, 高性能的模式被破坏; 过小, 搜索陷入迟钝状态。变异算子体现了群体的多变性, 变异概率( $P_m$ )过大, 搜索趋于纯粹的随机搜索; 过小, 将陷于局部最小。在这里, 交叉算子取标准的单点交叉, 其交叉概率  $P_c=0.7$ , 变异算子选取标准的单点变异, 变异概率  $P_m=0.01$ 。

第四步: 构造适应度函数

在前面求的是  $R(W')$  极大化的问题, 而且  $0 \leq R(W') \leq 1$ , 所以可直接选取  $R(W')$  作为适应度函数。

第五步: 设定终止条件

算法优化 1000 次(这里 1000 是经过多次试验得到的), 取出适应度最大的解作为最优解, 结束程序。

### 4 优化策略

为了使遗传算法更快地收敛到全局最优解, 同时使得得到的结果在实际中更适用, 在计算过程中采用了下面的一些优化策略。

(1) 运用杰出者选择与“父子混合”选择相结合的策略

为了使遗传算法能尽可能快地收敛到全局最优解, 采用了杰出者选择和“父子混合”选择相结合的策略。杰出者选择策略和“父子混合”选择策略都是遗传算法保证收敛的执行策略<sup>[3]</sup>。算法描述如下:

“父子混合”的杰出者选择遗传算法

步 1(初始化) 置  $t \leftarrow 0$ , 随机产生初始种群

$$\vec{X}(0) = (X_1(0), X_2(0), \dots, X_M(0)) \in H_M^L$$

步 2(种群进化)

2.1 独立地从当前种群  $\vec{X}(t)$  中选择出  $M$  对母体;

2.2 独立地对所选  $M$  对母体执行交叉, 运算获得  $M$  个中间个体;

2.3 独立地对交叉后所产生的  $M$  个中间个体执行变异得到  $M$  个个体

$$\vec{X}'(t+1) = (X'_1(t+1), X'_2(t+1), \dots, X'_M(t+1))$$

2.4 在  $\vec{X}(t) \cup \vec{X}'(t+1)$  中选择出  $M-1$  个个体  $X_1(t+1), X_2(t+1), \dots, X_{M-1}(t+1)$  作为  $t+1$  代的前  $M-1$  个个体, 同时选择一个最优个体  $X^*(t)$ , 并令  $t+1$  代种群的第  $M$  个个体  $X_M(t+1) = X^*(t)$ , 从而得到第  $t+1$  代种群为:

$$\vec{X}(t+1) = (X_1(t+1), X_2(t+1), \dots, X_{M-1}(t+1), X_M(t+1))$$

步 3(终止检验) 如果终止准则满足, 则输出  $X^*(t+1)$  作为问题的近似解, 否则, 置  $t \leftarrow t+1$  转步 2。

(2) 运用强个体策略

这里利用了遗传算法的遗传及变异等特性,把前面产生出来的次优解,放到初始种群中重新执行遗传算法,以期通过遗传、变异、交叉等算子把次优解进一步优化。这与单纯增加优化次数是不同的,前者利用了初始种群的多样性把次优解的性能进一步地改善,而后者则达不到如此功效,因为在遗传算法后期种群中的个体“性质”都比较接近,想通过交叉和遗传来继续优化已经不太可能了。

该策略也可以在一定程度上避免遗传算法的早熟现象。

### (3) 运用均衡策略

由遗传算法得到的解,尽管适应度函数很大,但在实际应用中不一定适用,原因是实际中不仅仅要求效率高,而且这些字母要标在手机键盘上,每个数字键上的字母的个数是有限的,不能排得太多,最多只能排4个。所以不得不把得到的结果进行改进,使每一个数字键上面的字母个数最多为4个,而又尽可能地少减少其适应值,文中采用如下算法对字母布局进行调整。分别用  $K_i, i=2..9$  表示每个数字键,该算法可以如下描述:

步1 如果存在某个  $i, 2 \leq i \leq 9$ ,使得  $|K_i| > 4$ ,则转步2,否则转步3;

步2 对每个字母  $c \in K_i$ , 分别从  $K_i$  中取出并依次放到  $K_j$  中  $j \neq i$ ,同时计算  $R(W')$ ,计算后再将  $c$  放回到  $K_i$  中。设  $R(W')$  取得最大时取出的字母为  $c'$ ,另外的数字键为  $K_j'$ ,把  $c'$  从  $K_i$  中取出,放到  $K_j'$  中,转步1;

步3 把数字键的结果输出。

## 5 拼音字母布局方案

传统的手机键盘布局是将二十六个英文字母从第一排开始,依序在每个键上放三或四个字母,一直放到第三排第三列。如表1所示:

表1

数字	2	3	4	5	6	7	8	9
拼音字母	abc	def	ghi	jkl	mno	pqrs	tuv	wxyz

这种布局只是把英文的布局照搬过来,在进行拼音输入时效果并不是很好。用这个布局来计算  $F(W')$ ,得到结果为  $F(W')=0.226$ 。也就是说只有22.6%的汉字拼音与键盘的输入是一一对应的。笔者用遗传算法得到的结果如表2所示。

在这种布局下,可以得到  $F(W')=0.629$ ,即有62.9%的汉

字拼音与键盘输入是一一对应的,大大地提高了键盘的利用率。对这两个布局分别用计算机模拟输入前面所说的244818个汉字,结果为:传统的键盘布局需要59403次翻页,而改进后的拼音字母布局只要38174次翻页,减少了36%。可见新的布局有效地增加了键盘的效率,提高了拼音输入的速度。

表2

数字	2	3	4	5	6	7	8	9
拼音字母	mnst	cqvw	orxz	di	ag	uy	hlbp	efjk

事实上,在实际中可以对该布局进一步调整。因为数字键和拼音字母组合不仅仅只有上面一种搭配,例如“mnst”字母组不一定只放到2号键上,也可以放到3号4号键上。在实际操作中尽量使得输入一个字母后,下一个要输入的字母就在该数字键周围的键上面。这就需要调整数字键与拼音字母组合的相对位置。仍然利用前面的汉字进行统计计算,最终得到如表3的布局:

表3

数字	2	3	4	5	6	7	8	9
拼音字母	orxz	cqvw	di	gnst	al	efjk	uy	bhpm

经过统计,这样的布局可以做到:输入一个字母以后,下一个要输入的字母以72%的概率出现在他周围的数字键上,这在一定的程度上方便了用户的输入。

## 6 结束语

该文用遗传算法初步探讨了手机、电话等通信终端的键盘字母布局问题,即用遗传算法给出了较好的结果。在实际模拟中说明了使用该种字母布局的手机键盘进行汉字输入比现在流行的最快的汉字输入法减少了36%次翻页,提高了汉字的输入速度。可见,这个结果在实际中是有效的,也说明了遗传算法在解决手机键盘字母布局中是可行的。(收稿日期 2002年11月)

## 参考文献

- 1.周明,孙树栋.遗传算法原理与应用[M].北京:国防工业出版社,1999
- 2.Holland John H.Adaptation in Natural and Artificial System[M].Ann Arbor:The University of Michigan Press,1975
- 3.徐宗本,聂赞坎,张文修.父代种群参与竞争遗传算法的几乎必然收敛性[J].应用数学学报,2002,25(1):167~175
- 6.R Melhem ,D Chiarulli S Levitan.Space multiplexing of waveguides in optically interconnected multiprocessor systems[J].The Computer Journal,1989,32(4):362~369
- 7.C Qiao ,R Melhem ,D Chiarulli et al.Optical multicasting in linear arrays[J].International Journal of Optical Computing,1991,2(1):31~48
- 8.C Qiao ,R Melhem ,D Chiarulli.Time-division optical communications in multiprocessor arrays[J].IEEE Transactions on Computers,1993,42(5):577~590
- 9.Keqin Li ,Yi Pan Si Qing Zheng.Fast and Processor Efficient Parallel Matrix Multiplication Algorithms on a Linear Array With a Reconfigurable Pipelined Bus System
- 10.U J J Le Verrier.Sur les variations seculaires des elements elliptiques des sept planetes principales[J].J Math Pures Appl,1840,5:220~254
- 11.J L Trahan ,A G Bourgeois ,Y Pan et al.An Optimal and Scalable Algorithm for Permutation Routing on Reconfigurable Linear Arrays with Optically Pipelined Buses[J].Journal of Parallel and Distributed Computing,2000,60(9):1125~1136

(上接76页)

## 参考文献

- 1.T Leighton.Introduction to Parallel Algorithms and Architectures:Arrays·Trees·Hypercubes[M].Morgan Kaufmann San Mateo,CA,1992
- 2.K Li Y Pan S Q Zheng.Parallel matrix computations using a reconfigurable pipelined optical bus[J].Journal of Parallel and Distributed Computing,1999,59(1):13~30
- 3.Y Pan ,K Li.Linear array with a reconfigurable pipelined bus system—concepts and applications[J].Information Sciences,1998,106(3/4):237~258
- 4.Yi Pan.Basic data movement operations on the LARPBS model[C].In:K Li ,Y Pan S Q Zheng eds.Parallel Computing Using Optical Interconnections ,Boston,USA Kluwer Academic Publishers,1998~10
- 5.Z Guo ,R Melhem ,R Hall et al.Pipelined communication in optically interconnected arrays[J].Journal of Parallel and Distributed Computing,1991,269~282