Placing Chinese Phonetic Symbols on Mobile Phones Keyboard Using Branch-and-bound Strategy

Chin-Chen Chang [1,2], Chi-Shiang Chan [1] and Kai-Jung Shih[1]

[1]Department of Computer Science and Information Engineering

National Chung Cheng University

Chiayi, Taiwan 621, R.O.C.

E-mail: {ccc, cch, jason}@cs.ccu.edu.tw

[2]Department of Information Engineering and Computer Science,
Feng Chia University, Taichung, Taiwan, 40724, R.O.C.

E-mail: ccc@cs.ccu.edu.tw

***Abstract*—In 2002, Tung et al. introduced the so-called intellectual Chinese phonetic symbol input system for the placement of Chinese phonetic symbols on mobile phones keyboard. In order to come to good placements, Thung et al. and Chang et al. have proposed their finding algorithms. In this paper, we shall also present a new method we have developed by using the branch-and-bound strategy. In our method, the tree structure of candidate solutions is built up first. Then, the branch-and-bound algorithm is applied. Experimental results show that our method can get better results than Thung et al.'s method and Chang et al.'s method.**

Key words: Chinese input system, Chinese phonetic symbols, mobile phone, branch-and bound strategy

I. INTRODUCTIOIN

Due to the astonishing growth of computer technology, data communication has become easier and easier. The most obvious change we have been making these years is the amazingly convenient mobile phones in our hands. Through the mobile phone, people can chat with others everywhere in the world only if they know what number to dial or have kept that number in his phone book. Nowadays, for a mobile phone company to stay competitive, besides being fashionably shaped, its products must be as small and delicate as possible. To make mobile phones small and delicate, one of the ways is to make the buttons only a few and the size of the buttons small. However, in addition to phone call making and receiving, the services modern mobile phones have to provide also include email (or text message) sending and receiving, which of course covers Chinese characters in the Chinese-speaking world. Up to the present time, in the literature concerned, there have been quite some researches devoted to the field of Chinese characters inputting in the computer system, for example, Chinese system [6] and Chinese document revision [5]. Besides, there have also been many different Chinese input systems developed, implemented, and studied, such as "Chinese Phonetic Symbols Input System [9]," "Chinese Speaking Input System [7]," and so on [1][2].

If we want to use Chinese phonetic symbols as our Chinese character input system, we must face the problem that the number of Chinese phonetic symbols is much greater than the number of buttons on the mobile phone. A simple, intuitive way to deal with this problem is to assign several Chinese phonetic symbols to each button on the mobile phone, leaving each button in charge of more than one phonetic symbol. But which Chinese phonetic symbols should be put together under the control of the same button? Traditionally, Chinese phonetic symbols are assigned to the buttons by the order we have long been used to, first few assigned to Button 1 and the next few assigned to Button 2 and so on [9]. In such systems, to key in the required Chinese characters, it takes a lot of time to pick out the required phonetic symbols and put them together. In order to speed up this tedious typing process, the intellectual Chinese phonetic symbol input system [8] has been created.

Formatted: Bullets and Numbering

The intellectual Chinese phonetic symbol input system can show the possible feasible Chinese phonetic symbol combinations according to the buttons the user presses. Because of the implementation of the intellectual Chinese phonetic symbol input system, the placement of the Chinese phonetic symbols becomes more and more important, more and more crucial to the efficiency of the system. A good placement can efficiently reduce the number of feasible Chinese phonetic symbol combinations. As a result, the times the buttons need to be pressed become only a few. Recently, Tung et al. [8] have proposed their method to decide the placement of the Chinese phonetic symbols statistically. To make a difference, Chang et al. [3] have also proposed their method by using a genetic algorithm.

In order to reduce the number of feasible Chinese phonetic symbol combinations further, in this paper, we shall propose our new method with a branch-and-bound strategy. First of all, we build up a tree structure of candidate solutions. Then, the branch-and-bound algorithm is applied on the tree structure. The upper bound of each node in the tree

1-4244-0220-4/$20.00? 2006 IEEE

structure must be calculated. When the cost of the current node is greater than its upper bound, the children of the current node can be eliminated. This way, we can save the time that would be wasted calculating unsuitable candidate solutions otherwise.

The rest of this paper is organized as follows. To begin with, we shall review some related works in Section 2, including a brief introduction to the structure of the intellectual Chinese phonetic symbol input system and a short tour around Chang et al.'s proposed method. Then, we will continue to present our method in Section 3, where the way to build the tree structure of candidate solutions is described first, followed by the way to apply the branch-and-bound algorithm on the tree structure to get better results. Then we shall offer our experimental results in Section 4, and finally the conclusions will be given in Section 5.

II. RELATED WORKS

In this section, let's have a look at the problem that we want to solve and some attempts that have been tried to solve it. First of all, we have the intellectual phonetic symbol input system [8]. With the system, the question here is which Chinese phonetic symbols should be put together onto the same buttons. Some researches [3, 8] have been brought up in an attempt to answer the question, the latest one being Chang et al.'s method [3] with a genetic algorithm to solve the placement problem of Chinese phonetic symbols. Their method will be described in the following paragraphs.

In the intellectual phonetic symbol input system, there are three components involved. One is the mobile phone, another is the Chinese phonetic symbol combination database, and the other is the Chinese vocabulary database. The placement planning is discussed in the mobile phone component. According to the decided placements of Chinese phonetic symbols, the system can find feasible Chinese phonetic combinations by referring to the Chinese phonetic symbol combination database. After selecting one feasible Chinese phonetic combination, the homophonic Chinese characters are shown by referring to the Chinese vocabulary database. The structure of the intellectual Chinese phonetic symbol input system is shown in Fig. 1.

For example, if we want to input the Chinese character "這", then we must press proper buttons to get this character. First of all, the Chinese character "這" can be traced down by the combination of the phonetic symbols "ㄓ" and "ㄜ". According to the placement of Chinese phonetic symbols on the mobile phone in Fig. 1, Button 5 and Button 7 should be pressed sequentially. There exist 16 different kinds of combinations. They are "ㄓㄚ", "ㄓㄛ", "ㄓㄜ","ㄓㄝ", "ㄔㄚ", "ㄔㄛ", "ㄔㄜ", "ㄔㄝ", "ㄕㄚ", "ㄕㄛ", "ㄕㄜ"," ㄕㄝ", "ㄖㄚ", "ㄖㄛ", "ㄖㄜ"and "ㄖㄝ". However, not all of those combinations have their corresponding Chinese characters. For example, there is a Chinese

Formatted: Bullets and Numbering

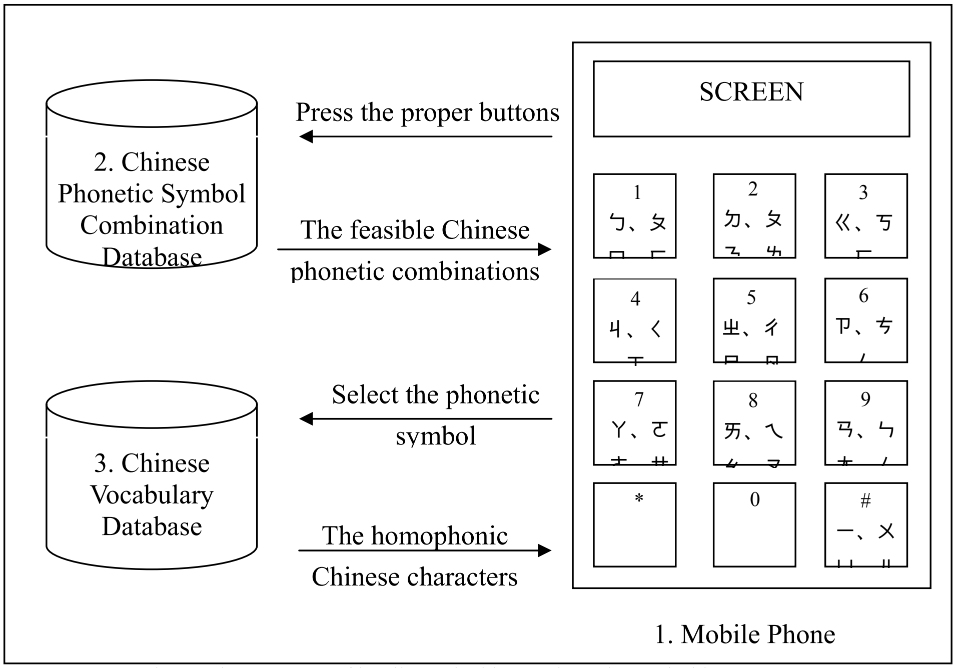


Fig. 1. The structure of intellectual Chinese phonetic symbol input system

character “這” for the combination “ㄓㄜ,” however, there is no Chinese character whatsoever for the combination “ㄓㄛ”. We call a combination with one or more corresponding Chinese characters by the name of a feasible Chinese phonetic symbol combination. Therefore, “ㄓㄜ” is a feasible Chinese phonetic symbol combination. After pressing Buttons 5 and 7, we can get six feasible Chinese phonetic symbol combinations, and the screen of the mobile phone will reveal these feasible Chinese phonetic symbol combinations. They are “ㄓㄚ”, “ㄓㄜ”, “ㄔㄚ”, “ㄔㄜ”, “ㄕㄚ” and “ㄕㄜ”.

To get the target Chinese character “這”, we have to pick out the proper combination from the feasible Chinese phonetic symbol combinations the two buttons indicate. The proper combination is “ㄓㄜ,” and what we can do is to press the selection button two times to select it from the candidate combinations. After pressing the button **OK**, the homophonic Chinese characters can be shown. The desired Chinese character can be selected from those homophonic Chinese characters.

It is obvious that the number of feasible Chinese phonetic symbol combinations directly relates to the placement of Chinese phonetic symbols on the mobile phone. If we can find a good placement of Chinese phonetic symbols on the mobile phone, the total numbers of feasible Chinese phonetic symbol combinations can be decreased efficiently. In order to find a good placement of Chinese phonetic symbols on the mobile phone, Chang et al. [3] then proposed their genetics-based selection algorithm.

Before going any further, let’s decide the number of phonetic symbols each button should have. The total number of phonetic symbols is thirty-seven, and the number of buttons on which we intend to put phonetic symbols is nine. One special fact that deserves special attention, however, is that there exits a special phonetic symbol called “ㄦ”. This special phonetic symbol does not have any feasible Chinese phonetic symbol combinations with any other phonetic symbol. Because of this special property, it should be left alone outside of the placement procedure. The special phonetic symbol can simply be assigned when all the other symbols are done. Therefore, what we have to do here is to try to place thirty-six phonetic symbols onto nine buttons, which means each button contains four phonetic symbols on the average.

Chang et al.’s way of solving the placement problem is by using a genetics-based algorithm. Generally speaking, when a genetic algorithm is used to solving a certain kind of complex problem, it first converts some candidate solutions to the problem into chromosomes. In Chang et al.’s method, they set the initial chromosome as containing a fixed number of genes. Each chromosome is in the form of $C=\{g_1, g_2, \ldots, g_9\}$, where $g_1, g_2, \ldots, g_9$ represent the genes. The genes can be viewed as the buttons on the mobile phone. Each gene contains four phonetic symbols. According to the definition above, the chromosome and the placement of phonetic symbols on the mobile phone can be transformed to each other.

At first, the method produces several chromosomes randomly. After such operations as reproduction, crossover and mutation, we can get generations and generations of chromosomes. Those new chromosomes are the candidate solutions to the problem, and they may be better or worse than the original candidate chromosomes. Therefore, we need a way to judge the quality of the new chromosomes. In a genetic algorithm, a function called the fitness function is used to do the judgment.

In Chang et al.’s method, the fitness function is defined for each button G_i $(1\square i\square 9)$ as follows.

$$\begin{aligned} C_i = f_1(\{G_i\}) + + \sum_{j=1 to i} f_2(\{G_i, G_j\}) \\ + \sum_{\substack{j=1 to i \\ k=1 to i}} f_3(\{G_i, G_j, G_k\}) \end{aligned} \quad , \qquad (1)$$

$G_1, G_2, \ldots, G_9$ mean the buttons containing phonetic symbols. To make it clear, we use the symbol (G_i, G_j, G_k) to represent the order when we press the buttons on the mobile phone. For example, (G_4, G_1, G_9) means we press the fourth button, the first button, and the ninth button in that order. Therefore, the parameter $\{G_i, G_j\}$ represents conditions (G_i, G_j) and (G_j, G_i). Moreover, function f_m $(1\square m\square 3)$ can be represented as Formula (2).

$$f_m(\{\qquad\}) = \sum_{n=1}^{N} n, \qquad (2)$$

where N is the number of feasible Chinese phonetic symbols of f_m $(\{\})$. For example, the function $f_1(\{G_i\})$ happens in a case where we only press one button, G_i. By the same token, $f_2(\{G_i, G_j\})$ means two buttons, G_i and G_j, are pressed. The order by which they are pressed may be either (G_i, G_j) or (G_j, G_i). Likewise, $f_3(\{G_i, G_j, G_k\})$ reveals the fact that three buttons, G_i, G_j and G_k, are pressed. According to the descriptions above, the value-of-fitness function for a chromosome can be calculated by taking $g_1, g_2, \ldots, g_9$ as parameters in Formula (1).

Using the fitness function, we can keep only the good chromosomes for the next generation while canceling the bad ones. That is, the survivor chromosomes will go through reproduction, crossover and mutation operations to get new chromosomes for the next generation. After going through several iterations, better chromosomes than the initials can be found.

III. THE PROPOSED METHOD

In this section, we shall present our method with the branch-and-bound strategy [4]. To begin with, we will briefly explain what a branch-and-bound strategy is about. Then, we will continue to apply the branch-and-bound strategy to solve the placement problem.

The branch-and-bound strategy is basically a state space search method. That is, all possible solutions will be represented as a tree structure. Each node of the tree has either a lower bound or an upper bound or both. In the searching process, if there exists any node whose current cost is out of range, the costs of its children need not be calculated because the solutions containing the node and its children will also be out of range. That is, these candidate solutions would not be our final solution. This way, a branch-and-bound algorithm can eliminate impossible solutions very quickly.

In this paper, we intend to use a branch-and-bound strategy to solve the Chinese phonetic symbol placement problem. Before going further, let's discuss the appearances of the candidate solutions. In [8], there is a constraint that each button must include two consonant symbols and one vowel symbol at least. In [3], however, Chang et al.'s method did not go by the same constraint. In their model of genetic algorithm, a button can contain four consonants or vowels. Chang et al.'s results turn out to be very theatrical. Of course Chang et al.'s results also include the two-consonants-and-one-vowel-at-least case. In the light of the observation above, in our method, we obey the [8] constraint when executing the branch-and-bound algorithm so as to reduce time and computation consumption.

Formatted: Bullets and Numbering

In our method, there are three steps to take. For the first step, we present the possible solutions in the form of a tree structure. The number of levels there are is the number of buttons, that is, nine. We give a simple example to show how the tree structure of solutions looks like. Assume that we have six consonants and three vowels in our example. The six consonants are "*A*", "*B*", "*C*", "*D*", "*E*" and "*F*", and the three vowels are "α", "β" and "γ". The number of buttons is three, and the new constraint is that each button contains two consonants and one vowel in this example. The solution tree structure is shown in Fig. 2.

As Fig. 2 shows, level one represents the first button. The possible cases at this level are under the condition that consonant "*A*" will be included. That means each case contains the consonant "*A*" and we name consonant "*A*" the leading consonant. Level one always takes consonant "*A*" as leading consonant. On the other hand, which consonant should be taken as the leading consonant for other levels depends on its parent node. For example, the node "*AB*α" at level one owns two consonants "*A*" and "*B*". Therefore, its children at second level can only take the third consonant "*C*" as their leading consonant. According to the rule, we can build up the tree structure of candidate solutions.

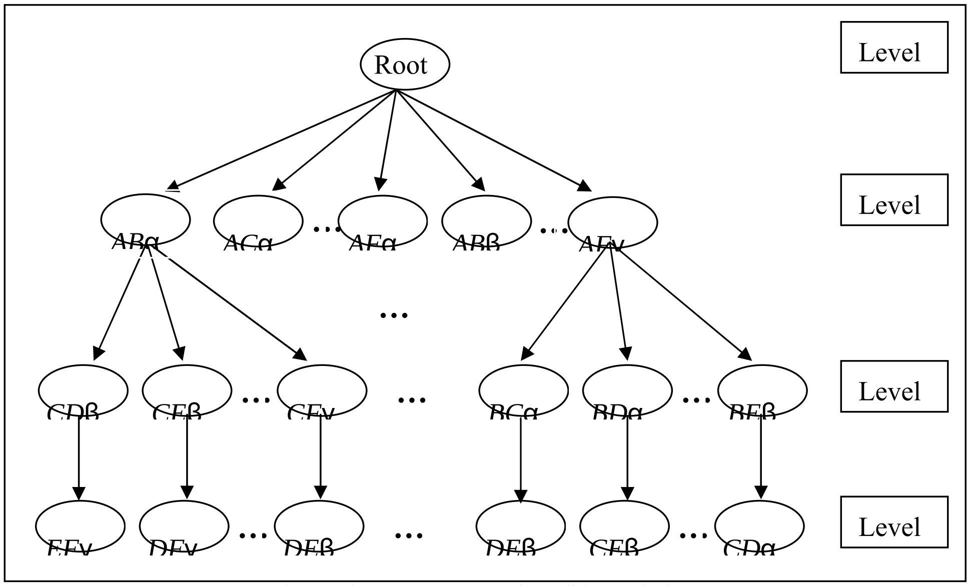


Fig. 2. The tree structure of candidate solutions

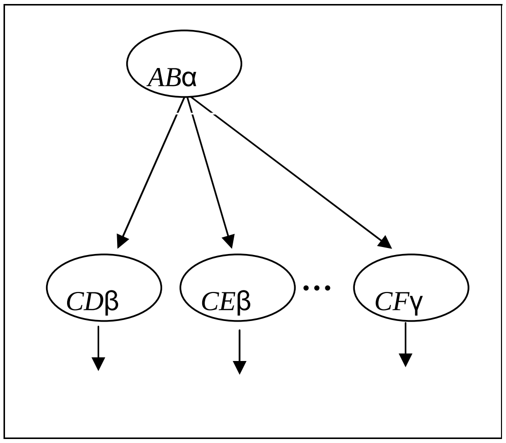


Fig. 3. The sub- tree structure of the tree in Fig. 2

The second step of our method is to calculate the current cost for each node. Note that the total cost of the current node comes from adding the cost of its parent and "the cost of itself". The cost of the tree root is set to be zero. When calculating the cost of the "*AB*α" node at level one, for example, we treat the node as the first button, that is, G_1. According to Formula (1), therefore, the cost of the node is $f_1(\{G_1\}) + f_2(\{G_1, G_1\}) + f_3(\{G_1, G_1, G_1\})$.

When calculating the cost of the node "*CD*β" at level two, the node is treated as the second button, that is, G_2. The total cost of the current node, G_2, comes from the cost of its parent node and "the cost of itself". "The cost of itself" means the additional cost caused by adding the current node into the mobile phone. Therefore, the cost of itself is $f_1(\{G_2\}) + f_2(\{G_2, G_1\}) + f_2(\{G_2, G_2\}) + f_3(\{G_2, G_1, G_1\}) + f_3(\{G_2, G_1, G_2\}) + f_3(\{G_2, G_2, G_2\})$. Note that $f_3(\{G_2, G_1, G_2\})$ and $f_3(\{G_2, G_2, G_1\})$ are both combinations of G_1, G_2, G_2. They are (G_1, G_2, G_2), (G_2, G_1, G_2) and (G_2, G_2, G_1). Adding costs of the parent node and "the cost of itself", we can attain the total cost of the current node. Here, the result is the same as that calculated by using Formula (1).

It goes without saying that the current node can be calculated if the cost of its parent node has been figured out. Moreover, if the cost of the current node is greater than the upper bound (threshold), then the costs of its children will also be greater than the upper bound. In this case, the current node and its children should be discarded because there does not exist any desired solution in this sub-tree.

In order to further reduce computation load, the cost of the common part shared by the current node and its sibling(s) can be calculated first. This means that we will add the phonetic symbols and calculate the costs one by one. Therefore, the sub-tree can be discarded immediately when the total cost of adding phonetic symbols is greater than the upper bound. For example, Fig. 3 is a sub-tree that takes the node "*AB*α" as the root. The common parts of the current node "*CD*β" and its sibling are drawn out as a common node first. The outlook of the sub-tree with common nodes is illustrated in Fig. 4.

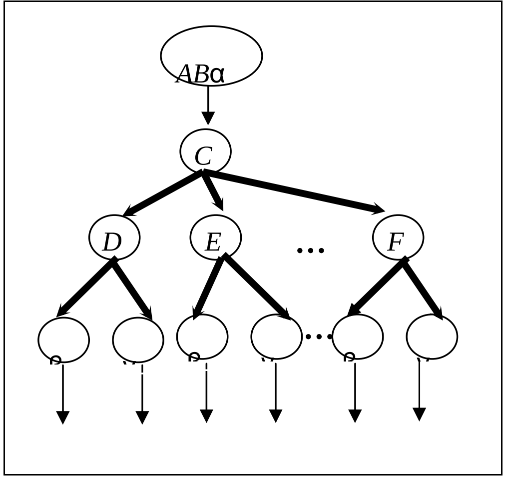


Fig. 4. The sub- tree structure with common part

For example, if we want to calculate the cost of the node "*CD*β", we first add phonetic symbol "*C*" to button 3 and calculate its total cost after adding it. Then the phonetic symbols "*D*" and "β" are added to button 3 sequentially. If the total cost is greater than the upper bound when the phonetic symbol "*D*" is added, for example, then we do not need to calculate the costs of its children. We then continue to add "*E*" to button 3 and calculate its total cost.

For the third step, if the procedure reaches the leaf node and the total cost is smaller than the upper bound, it implies that the total cost of the current placement of phonetic symbols is better than the current cost, which is used to be the upper bound. In this case, we find a better placement of phonetic symbols than the current placement.

In this paper, we shall not discuss much as to how to calculate the value of the upper bound. In Chang et al.'s method, the resultant value of the fitness function is 672, and we take it as the basic upper bound. For each added phonetic symbol *i*, its own new upper bound value, NB_i, can be calculated from the basic upper bound. The new upper bound can be calculated as follows.

$$NB_i = (627 - (TFN - FN_i)), \qquad (3)$$

where *TFN* is the total number of feasible Chinese phonetic symbol combinations in the Chinese phonetic symbol combination database. *TFN* is a constant value—406. The variable symbol FN_i is the number of feasible Chinese phonetic symbol combinations that come from combining the phonetic symbols extracted from the Chinese phonetic symbol combination database, that is, adding phonetic symbol *i* and its ancestors. Note that the value of FN_i can be directly figured out from the number of feasible Chinese phonetic symbol combinations without using Formula (2). According to Formula (3), we can eliminate the sub-tree immediately when the cost of the current added phonetic symbol is greater than the upper bound value.

IV. EXPERIMENTAL RESULTS

In this section, we shall show our experimental results and evaluate the performance of our method. Our performance will be compared with those of the Tung et al.'s method and Chang et al.'s method.

Let's look at the Chinese phonetic symbol combination database first. It keeps cases of feasible Chinese phonetic symbol combinations. The total number of Chinese characters is greater than ten thousand with half of them commonly used. Although there are so many Chinese characters, most of them share the same phonetic symbol combinations with many others, with or without the same tones.

The total number of feasible Chinese phonetic symbol combinations in the database is four hundred and six. This number is the total cost in the case where each phonetic symbol is placed onto a different button. On the other hand, if we want to assign all the phonetic symbols to nine buttons, then the feasible Chinese phonetic symbol combinations must be more than four hundred and six according to Formula (2). Therefore, four hundred and six is the lower bound, and the costs of placement methods must be greater than it in the case where we have to put thirty-six phonetic symbols onto nine buttons.

Since the Chinese phonetic symbol combination database is readymade, we can run our procedure by referring to it. That is, we first build up the tree structure of candidate solutions. Then, the branch-and-bound strategy is applied on the tree structure to get better results. Moreover, the upper bound can be calculated for each node according to Formula (3). The final placement results of different methods are shown separately. The general placement is used in most mobile phones, and Chang et al. call this placement the general placement (see Fig. 5). The placements of Tung et al.'s method and Chang et al.'s method are illustrated in Fig. 6(a) and Fig. 6(b), separately. The placement of our method is shown in Fig. 7.

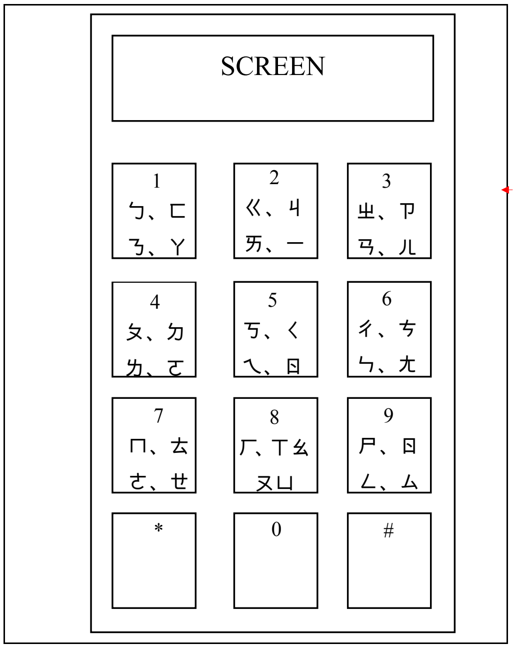
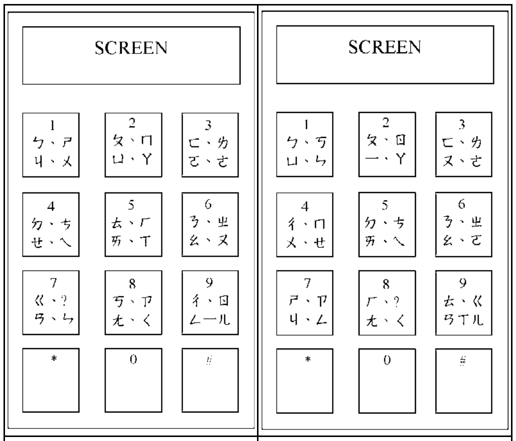


Fig. 5. The general placement of phonetic symbols on a mobile phone

Formatted: Bullets and Numbering



(a) Tung et al.'s method (b) Chang et al.'s method

Fig. 6. The placements of phonetic symbols from different methods

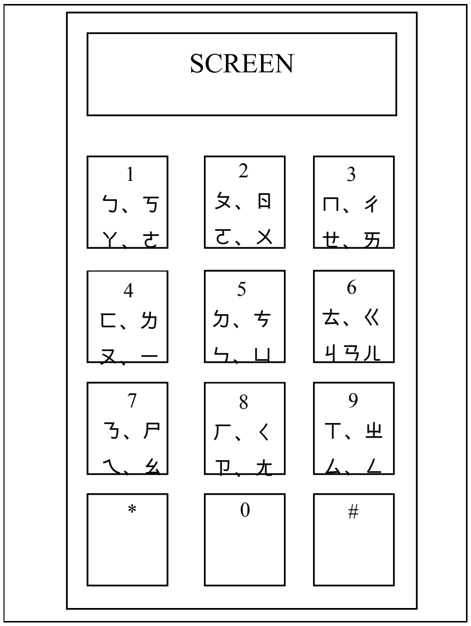


Fig. 7. The placement of phonetic symbols from our method

The total costs, which are calculated by using Formula (1), of different methods are shown in Table 1.

TABLE 1. THE TOTAL COST OF DIFFERENT PLACEMENT

Results	The general placement	The placement of Tung et al.'s method	The placement of Chang et al.'s method	The placement of our method
Total cost	789	759	672	658

Compared with the other placement methods, our method has the lowest cost. That means the total number of times the buttons have to be pressed in our method is the fewest.

V. CONCLUSION

There are several methods available to deal with the placement problem of Chinese phonetic symbols on the mobile phone keyboard. In this paper, we propose a new method of Chinese phonetic symbol placement by using the branch-and-bound strategy. The branch-and-bound strategy eliminates sub-trees when the cost of the current node is greater than the upper bound, which avoids wasting time and computation costs on calculating the unsuitable nodes. Therefore, our method can work quite efficiently.

According to the experimental results, the cost of our method is the lowest among the methods proposed preciously. With our method, the produced placement of Chinese phonetic symbols can reduce the time needed to press the buttons on the mobile phone. Moreover, once we get the placement of Chinese phonetic symbols, it can be used in the input system of the mobile phone. Therefore, it is worth taking a little time to find a better placement of Chinese phonetic symbols. In a word, our new scheme is quite a practical one to find a good Chinese phonetic symbol placement.

REFERENCES

[1] C. C. Chang, T. S. Chen and Y. Lin, " An Efficient Accessing Technique of Chinese Characters Using Boshiamy Chinese Input System, " *Proceedings of the International Workshop on Information Retrieval for Asian Language*, Hong Kong, 2000, pp. 61-67.

[2] C. C. Chang and C. F. Lee, " Retrieving Chinese Characters Using DAYI Chinese Input Method, " *Proceedings of the 17th International Conference on Computer Processing of Oriental Languages*, Hong Kong, 1997, pp. 387-391.

[3] C. C. Chang, K. J. Shih and I. C. Lin, " An Efficient Placement of Chinese Phonetic Symbols for an Intellectual Chinese Phonetic Symbol Input System in Mobile Phones Using Genetic Algorithm, " *Proceedings of 2003 IEEE International Conference on Natural Language Processing and Knowledge Engineering*, NLP-KE'03, Beijing China, Oct. 2003, pp506-511.

[4] E. Horiowitz, S. Sartaj and S. Rajasekaran, "Branch and Bound" *Computer Science Press*, New York, pp. 379-416.

[5] J. J. Kuo and M. C. Ke, " Development of Automatic Chinese Document Revision System Using Phonetic Information and Mask Technology, " *Proceedings of National Computer Symposium*, Taiwan, vol. 2, 1997, pp. D8-D13.

[6] M. C. Lee, " Chinese System Research, " *Flag Publishing Co.*, Taipei, Taiwan 2000.

[7] J. H. Liu and B. Y. Tang, " Chinese Symbol Phonetic Input, " *King Information Inc.*, Taipei, Taiwan, 2000.

[8] C. L. Tung, T. S. Chen and Y. C. Chiang, " An Intellectual Chinese Input System for Mobile Phones Based upon Chinese Phonetic Symbols, " *International Journal of Computer Processing of Oriental Languages*, vol. 15, no. 4, 2002, pp. 395-406.

[9] S. H. Yan, " Chinese Speaking Input, " *Translator Publishing Co.*, Taipei, Taiwan, 1998.