JOURNAL OF INDUSTRIAL AND PRODUCTION ENGINEERING
https://doi.org/10.1080/21681015.2018.1508080

Taylor & Francis
Taylor & Francis Group

ARTICLE

Hybrid estimation of distribution algorithms for solving a keyboard layout problem

T.G. Pradeepmon, Vinay V. Panicker and R. Sridharan

Department of Mechanical Engineering, National Institute of Technology, Calicut, India

ABSTRACT

The use of smartphones and handheld devices in our daily activities has sharply increased. The added features of wireless technology and related applications on these devices make it possible to write emails, notes, and long text. Even though the most commonly used electronic input device is a keyboard, very little work has been dedicated for finding an optimal layout for this device. In this research, the aim is to propose a better layout for the single-finger keyboard in terms of rapid typing. The keyboard layout problem can be formulated as a quadratic assignment problem, which is one of the hardest combinatorial optimization problems. Some well-known literary works in English are chosen for estimating the keying-in-time. A variant of genetic algorithm, namely, the estimation of distribution algorithms is used to find a better layout. The new layout is found to be efficient compared with some of the existing prominent keyboard layouts.

ARTICLE HISTORY
Received 9 December 2016
Accepted 11 June 2018

KEYWORDS
Single-finger keyboard; estimation of distribution algorithm; quadratic assignment problem; keyboard layout problem; population based incremental learning; univariate marginal distribution algorithm

1. Introduction

The most commonly used input device for computers and other text processing electronic devices is the keyboard. However, a novice computer user will get surprised by the seemingly arbitrary arrangement of keys in the standard keyboard. The most widely used keyboard is the "QWERTY" keyboard – named after the first six characters in the top row of letters – or the Sholes keyboard – named after its inventor, Christopher Latham Sholes. This keyboard layout was designed in 1868 by Sholes, for use in a mechanical typewriter in which the typing is done with both the hands. In this keyboard, the letters are arranged in a special, non-alphabetical order to improve the speed of typing by minimizing the jamming of keys in the typewriter. Since then, the QWERTY layout had become popular and is still the most accepted device used to input data into personal computers. However, the recent spread of smartphones, Portable Data Assistants (PDA), Tablet PCs, etc., has necessitated the redesign of the keyboard layout. Since the digital keys do not jam, it is possible to place any combination of keys nearby. Many of these devices are held in one hand and only a single finger of the other hand is used for pressing the keys to input data (hence, the name "single-finger (s-finger) keyboard"), which makes the s-finger keyboard a necessity.

The QWERTY keyboard was developed by Christopher Latham Sholes and he obtained patent for the same in 1968. He continued to develop on his typing machine and one of the problems faced by him was the jamming of the type bars when certain combinations of keys were struck in a very close succession. In order to solve this problem, Sholes arranged the keys in such a way that the keys most likely to be struck in close succession approached the type point from opposite sides of the machine. Thus, the layout was not actually designed for increasing the speed of typing, but to decrease the jamming of the keys during typing. In 1873, the patent was sold to E. Remington and Sons who added further mechanical improvements and began the commercial production of the typewriters using this keyboard [1]. The QWERTY keyboard was accepted as a standard and people using it outperformed others in typing competitions held in those years. Thus, the QWERTY keyboard became popular among people using typewriters.

In the year 1936, August Dvorak designed a new layout for the keyboard and obtained patent for the same. This keyboard layout is known as Dvorak Simplified Keyboard. It claimed to reduce the finger movements necessary for typing by equally dividing the load between the hands and loading the stronger fingers more heavily. After the introduction of Dvorak keyboard, there were many debates on the efficiency of the two layouts. Some authors even claimed that the QWERTY keyboard actually reduced the typing speed and, by doing so, avoided the jamming of keys. The Dvorak keyboard claimed the advantages of greater speed, reduced fatigue, and easier learning. Even with all the claims, the Dvorak keyboard failed to find much acceptance.

CONTACT Vinay V. Panicker vinay@nitc.ac.in Department of Mechanical Engineering, National Institute of Technology, Calicut, India

The competition between the two keyboards was of interest to economists also, and they used this example to demonstrate how the standards are being set and how it is difficult to replace an already set standard with a new one even if the new one is better. Paul A. David started this discussion in 1985 through a paper in which he claimed that the Dvorak keyboard is better than QWERTY keyboard [2]. But Liebowitz and Margolis rejected the claim through their paper "The fable of Keys" [1]. The battle on the superiority of the keyboards continued through a series of papers [3]. During this period, some researches tried to find out a better layout for the keyboard than the QWERTY and Dvorak. Many of such works focused on developing customized keyboards for special purposes.

Eggers et al. [4] introduced an abstract representation of a keyboard and an evaluation function taking account of ergonomic criteria is proposed. The resulted optimization problem was named the keyboard arrangement problem and based on the generic framework of Ant Colony optimization; an algorithm is developed and applied to this problem.

Kwon et al. [5] compared the performance and the subjective ratings of the conventional finger touch text-entry method and the regional error-correction method, for a touch-screen QWERTY keyboard. The regional error correction reduced both the time and the number of touches required to complete text entry when keys were small, but no difference was observed when keys were large.

Cardinal and Langerman [6] investigated the problem of placing symbols of an alphabet onto the minimum number of keys of a small keyboard so that any word of a given dictionary could be recognized unambiguously only by looking at the corresponding sequence of keys. The paper also provides optimized incomplete keyboards with 6, 8, and 12 keys. Curran et al. [7] conducted a survey to establish which mobile phone text input method best suits the requirements of a select group of target users. The survey used a diverse range of users to compare devices that are in everyday use by most of the adult population. The targeted group preferred larger keypad/keyboard devices to their smaller equivalents and the standard keypad layout to the non-standard layout on mobile phones.

Yin and Su [8] proposed a Cyber Swarm method considering multiple objectives and it accommodates ergonomic criteria and disambiguation/prediction effectiveness simultaneously. Experimental results proved that the proposed keyboard outperforms several benchmark keyboards and other competing algorithms. The work also gives an illustrative example for preliminary keyboard shape design, which could be very useful in customized keyboard production for motor-impaired users whose physical capacity has been evaluated a priori.

Alswaidan et al. [9] proposed a genetic algorithm to find an optimal layout for the s-finger Arabic keyboard. To adapt the problem to the requirements of optimizing the s-finger Arabic keyboard, two measures, namely, the keyboard row weight and the hit direction of the finger were added to the objective function of the Quadratic Assignment Problem (QAP). The resulting keyboard was compared with the Arabic keyboard layouts developed for m-fingers in Khorshid et al. [10], Malas et al. [11], and the iPhone Operating System (iOS) Arabic keyboard and was proved to be better in terms of improving in the objective function value. Govind and Panicker [12] proposed a Genetic Algorithm for the optimization of the s-finger keyboard problem. In the paper, the problem was considered a multiple attribute decision-making problem and Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) is applied to find out the best layout. The attributes selected were to minimize the flow distance, learning percentage, and maximize the typing speed.

Although many works related to text entry on mobile devices had been conducted in the human–computer interface zone, it seems that not many of them treat the problem of redefining the actual keyboard layout. Amico et al. [13] consider a s-finger keyboard layout to be a generalization of the well-known QAP. In this paper, a s-finger keyboard layout is considered as a QAP and an optimal layout is determined for the keys in terms of typing time. Since the QAP is proved to be an NP-hard problem [14], there is a need to develop heuristic algorithm to solve it to near-optimality. McKendall and Li [15] proposed a generalization of QAP and proposed a tabu search for solving it. Other heuristic and metaheuristic methods for solving QAPs were proposed by Tosun et al. [16], Hafiz and Abdennour [17], Zhu et al. [18,19], etc. In this work, the Estimation of Distribution Algorithms (EDAs), which are relatively new evolutionary algorithms, are employed for solving QAPs. In order to improve the effectiveness of the algorithms, two local search methods are also employed. Similarly, in this work, a keyboard layout problem is formulated as a QAP. The size of the QAP is 27, as the language considered is English and the punctuations are omitted for simplicity. A summary of literature survey is provided in Table 1.

In the proposed research, the experiments are carried out in three stages. In the first stage, four algorithms are developed combining two EDAs and two local search methods. The effectiveness of the algorithms in solving QAPs is verified by solving 57 benchmark problems taken from QAP Library. In the second stage, a test instance is generated using seven literary works in English and word frequency list of English language. Using this test instance, optimized layouts for least typing time and least finger movement are found out. These layouts are employed for validation experiments in the third stage.

The rapid increase in the usage of handheld electronic data entry devices demands the relaying out of the on-screen keyboards. For faster data input using only a single finger, the best layout for the keyboard may not be the one which is optimized for all the 10 fingers. This is the motivation behind this research. The proposed layouts are promising in terms of typing time and this direction of relaying out the keyboard is a significant contribution of this work. The algorithms derived from EDAs and local searches are employed for solving QAPs, which are one of the hardest of combinatorial optimization problems. The possibility of employing EDAs for solving QAPs has not been much explored in the past. Thus, the objectives of this work are as follows:

- Develop algorithms combining EDAs and local searches for solving QAPs.
- Establish the effectiveness of the algorithms using benchmark instances taken from QAP Library.
- Generate new layouts for keyboard, optimized for typing time and finger movement.
- Validate the layouts by conducting physical experiments on the optimized layouts.

The rest of the paper is composed as follows: Section 2 presents a brief review of currently existing keyboards. The general description of the s-finger keyboard problem is described in Section 3. Section 4 explains the concepts of algorithms used in this study, namely, EDAs and 2-opt local search. Section 5 provides the details of the computational experiments carried out and the results obtained. The modified keyboard layout is also presented in Section 5. Section 6 concludes the paper.

2. Keyboards

A keyboard is a typewriter-like device, which is used to enter data to an electronic device using an arrangement of buttons or keys. A keyboard has characters inscribed or printed on the keys and each key press corresponds to a single written symbol. However, some symbols require pressing or holding of several keys all together or in specific sequence. Most of the keys on the keyboard produce characters, but other keys or simultaneous key presses may produce actions or execute computer commands. Even though there are a number of other input devices, such as mouse, touch screen, pen devices, character recognition and voice recognition, the keyboard remains the most commonly used device for the direct input of alphanumeric data into computers.

2.1. n-finger keyboards

These keyboards use more than one finger for typing. Most of the n-finger keyboards are designed for typing with all the fingers on both hands.

2.1.1. QWERTY keyboard

This keyboard layout was designed by Christopher Latham Sholes in 1878 for use in a typewriter with both the hands. The letters were arranged in a special, non-alphabetical order so as to improve the speed of typing by minimizing the jamming of the typewriter. Figure 1 shows the layout of QWERTY keyboard. The QWERTY layout is the most popular alpha-numeric data input device for personal computers. However, this layout suffers from the following drawbacks [20]:

- Many common letter combinations require awkward finger motions or a finger to jump over the home row or need to be typed with one hand.
- In this keyboard, most of the typing is done with left hand, but most people are dominant on their right hand.
- Moreover, about 16% of typing is done on the lower row, 52% on the top row, and only 32% on the home row.

2.1.2. Dvorak keyboard

The Dvorak layout was patented in 1936 by Dvorak and Dealey. Figure 2 shows the keyboard layout proposed by Dvorak and Dealey. This layout claims to provide lesser finger motions, increased typing rate, and reduced errors compared to the standard QWERTY keyboard. This leads to faster rates of typing while reducing repetitive strain

Figure 1. QWERTY keyboard.

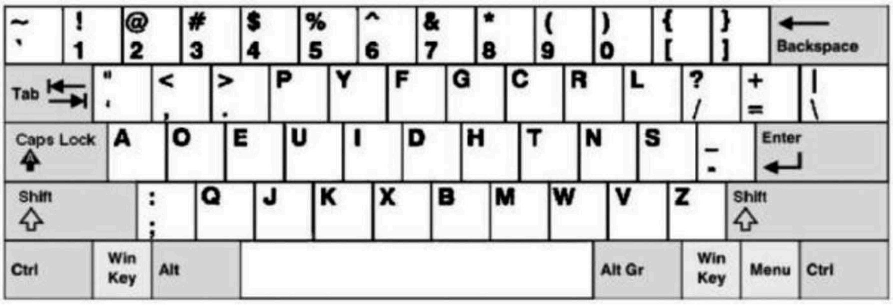

Figure 2. The Dvorak keyboard.

injuries. Nevertheless, these advantages have been questioned by Liebowitz and Margolis [1]. Over the years, several slight variations were designed by the team led by Dvorak or by American National Standards Institute (ANSI); however, it failed to replace the QWERTY keyboard.

Many researchers have tried to compare the behavior of various keyboard layouts [21]. Recently, researchers have started applying quantitative methods to design optimal layouts for keyboards. Egger et al. [4] has considered the problem of assigning characters to keys arranged in a pre-specified layout structure.

2.2. s–finger keyboards

All the aforementioned layouts are used for the purpose of data entry to personal computers. With the recent spread of mobile phones and PDAs, the necessity for small keyboard layouts has been identified by researchers in this field. Users of such devices normally use just one finger to type in the data to be entered while holding the device in the other hand. This situation gave way to the concept of s-finger keyboard layouts. The size has to be reduced so as to include all the characters in a limited space. Many different layouts are proposed by various researchers, which can be classified into two main categories: complete and incomplete keyboards.

2.2.1 Complete keyboards

These are the keyboard layouts similar to QWERTY or Dvorak layouts, but the size is reduced and the keys are arranged in such an order that only one finger is being used for the purpose of text entry. An example of such a keyboard is the FITALY keyboard patented by Textware solutions. It has been designed for English on the basis of the corresponding words frequency. The ABC layout, OPTI, Metropolis, Hooke, Lewis, and many more layouts are available. MacKenzie and Soukoreff [22] provides a complete review of such layouts.

2.2.2. Incomplete keyboards

In an incomplete keyboard, many characters are incorporated into the same key and repetitive pressing gives these characters in order. An example of this type of keyboard layout is the 12-key keyboards commonly used in mobile phones. The problem of finding an optimal layout for an incomplete keyboard (called KEYBOARD) by Cardinal and Langerman [6] is a combinatorial optimization problem with an aim of finding the minimum size partition of an alphabet, allowing the users to type any word of a given dictionary so that each word is recognized without ambiguity. Cardinal and Langerman [6] report that the optimal layout determination problem is NP-hard even if it is decided that two keys are sufficient.

3. Problem description

This paper considers the s-finger complete keyboard layout problem with the following assumptions:

- All keys or locations except one in the position of spacebar in QWERTY keyboard are unit squares and arranged and numbered as shown in Figure 3. The unit squares represent a "key" or a possible "location."
- Each key or location must contain exactly one of the different symbols or characters used in the given language, say English in this case.

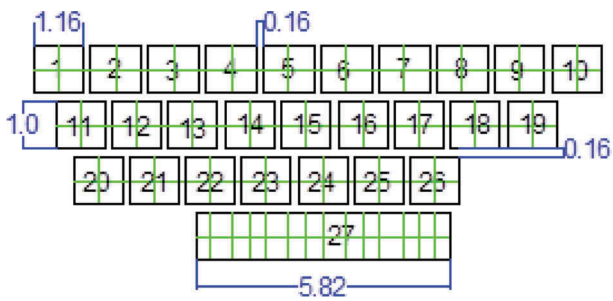

Figure 3. Dimensions and positions keys in the keyboard.

The first assumption is that the keyboard considered for the experiments is a normal keyboard with QWERTY layout, which can be transformed into other layouts by rearranging the positions of the keys. The second assumption implies that the keyboard cannot contain duplicate symbols. Now, the problem is to assign the symbols or characters to the locations while minimizing the average time required to type a statement in the given language.

The average time to type a statement using an s-finger keyboard can be computed by considering the frequency in which each ordered pair of symbols appears in the language and the time needed to move the s-finger between the keys accommodating these symbols. The average time can be obtained by summing up the product of the frequency with the corresponding time over all the ordered pairs of symbols.

Fitts's law experiment is often applied when studying the performance of typing devices like keyboards. The Fitts's experiment proved that movement time is related to movement distance and target size [23]. The effort or time taken for typing (T_{ik}) two symbols i and k, consecutively is given by the Fitts's law [24]

$$T_{ik} = \theta + \phi \log_2\left(\frac{D_{ik}}{W_k} + 1\right) \tag{1}$$

where θ and ϕ are constants having pre-fixed values, D_{ik} is the distance between the keys assigned with symbols i and k, and W_k is the width of the key assigned with symbol k. The contribution to the overall writing time due to the repetition of a symbol can be omitted, as it is independent of the assignment of the symbol to a location. Moreover, the equal sized key assumption in our problem enforces that the objective function only depends on the distance between the centers of two locations. In this work, $\theta = 0$ and $\phi = 10/49$ in accordance with Soukoreff [24].

The s-finger keyboard layout problem can be stated as to find the permutation φ from the set of all permutations of the number of symbols P, which minimizes

$$Z = \min_{\varphi \in P} \sum_{i=1}^{N} \sum_{k=1}^{N} a_{ik} b_{\varphi(i)\varphi(k)} \tag{2}$$

where N is the number of symbols or characters to be assigned to N locations or keys, a_{ik} is the number of occurrences of the symbol k immediately after symbol i in the text, $b_{\varphi(i)\varphi(k)}$ is the value of Fitts's function, computed with respect to the Euclidean distance from the center of key assigned with symbol $\varphi(i)$ to the center of key assigned with symbol $\varphi(k)$. The value of Z provides the total time in seconds to type the text with the given ordered symbol pair frequencies. This formulation invariably represents a QAP of size N. In addition to the Fitts's time taken for keying in the text, the total distance covered by the finger is minimized. This is achieved by replacing the b_{jl} values with the Euclidean distance from the center of location j to the center of location l.

QAP is a well-known NP-hard problem and hence the s-finger keyboard problem, stated earlier, is also an NP-hard problem. QAP is one of the most challenging optimization problems and benchmark instances proposed by Nugent et al. [25] in the late sixties have been solved exactly only recently.

4. Solution methodology – algorithms

This work aims to develop four algorithms combining two EDAs namely Univariate Marginal Distribution Algorithm (UMDA) and Population-Based Incremental Learning Algorithm (PBILA) with the recursive versions of 2-opt local search denoted by S2opt and R2opt. The following four algorithms are developed combining two EDAs and two local search methods as follows:

(1) UMDA with S2opt,
(2) UMDA with R2opt,
(3) PBILA with S2opt and
(4) PBILA with R2opt.

The details of the algorithms employed in this work are explained in the following sections.

4.1. *Estimation of distribution algorithms (EDAs)*

EDAs, also known as Probabilistic Model Building Genetic Algorithms (PMBGAs), incorporate probability into heuristic search procedures. EDAs are new development in genetic and evolutionary computation research. The algorithms exploit a feasible probabilistic model built around superior solutions so far found within the problem domain. There are three basic steps in EDAs as follows:

(1) Select superior candidate solutions from an initially randomly generated population.
(2) Estimate the probability distribution from the selected solutions.
(3) Generate new candidate solutions or offspring from the estimated probability distribution.

It may be noted that the second and third steps differentiate the EDAs from Genetic Algorithms by replacing the crossover and mutation operators. EDAs create a probabilistic model of the sample population and the new generation is based on this model. The main challenge faced by the EDAs involves estimating an accurate distribution and thus creating a probabilistic model that can represent the structure of the given problem effectively. This has led to the development of various categories of EDAs, like UMDA, PBILA, Compact Genetic Algorithm, Bivariate Marginal Distribution Algorithm, Mutual Information Maximization for Input Clustering Algorithm, Bayesian Optimization Algorithm, etc. [26]. EDAs were used for solving a variety of problems such as scheduling [27–29], control design, [30] etc.

While solving QAP, EDA maintains during each generation t, a population of m solutions $\Phi_{(t)} = \{\varphi^1_{(t)}, \varphi^2_{(t)}, \varphi^3_{(t)}, \ldots, \varphi^m_{(t)}\}$ and a probability matrix $\Pi_{(t)} = \begin{pmatrix} \pi_{11(t)} & \cdots & \pi_{1N(t)} \\ \vdots & & \vdots \\ \pi_{N1(t)} & \cdots & \pi_{NN(t)} \end{pmatrix}$ which models the distribution of solutions in $\Phi_{(t)}$. Figure 4 gives the pseudo-code of general EDA.

In this work, two EDAs namely, UMDA and PBILA are used for developing algorithms.

4.1.1. Univariate marginal distribution algorithm

UMDA was proposed by Muehlenbein and Paaß [31], and it is one of the early works in the field of EDAs. This algorithm assumes that all the variables are independent, i.e. the value of variable *Xi* does not depend on the state of any other variable. It does not carry forward the probability of model in the previous generations. During each generation, a new population is produced based upon the model generated from the current population [32]. Figure 5 gives the pseudo-code of general UMDA.

1. Generate a population Φ of m number of solutions
2. Select set Θ from Φ consisting of n promising solutions, where $n <= m$
3. Estimate univariate marginal probabilities $\Pi(x_i)$ from Θ for each x_i
4. Sample $\Pi(x_i)$ to generate m new individuals and replace Φ
5. Go to step 2 until termination criteria are met

Figure 5. Pseudo-code of UMDA.

4.1.2. Population-based incremental learning algorithm

PBILA is another simple EDA, and this also assumes no dependence among the variables. The statistical model in use is a real-valued vector with each element independently representing the probability of assigning value 1 to each corresponding bit in a binary string (candidate solution). PBILA starts with a probability vector with all elements set to 0.5, which means that each bit in a generated individual is set to 0 or 1 with equal probability. During evolution, the value of each element is updated using the best individual in the population and drifts away from 0.5, modifying its estimation of the structure of good individuals. Typically, PBILA will converge to a vector with each element close to 0 or 1. During each generation, the probability vector is updated as per the equation $\boldsymbol{\Pi_{(t+1)} = (1-\alpha)\,.\Pi_{(t)} + \alpha \cdot \pi_{(t)}^{Best}}$. But, instead of using best solution, the proposed algorithm uses all solutions in the current generation to update the probability vector. This is continued till the probability vector when rounded, becomes a valid solution [33]. Figure 6 gives the pseudo-code of general PBILA.

There are two decision parameters to be made, namely, (i) the value of the learning rate parameter, alpha (α) and (ii) the number of elite individuals (n) used to update the vector. In the current study, alpha is set to 0.1 and n is set to 10.

4.2. 2-opt local search

In simple 2-opt local search method [34], the algorithm searches the set of all possible solutions resulting from swapping two distinct elements of the current solution and the best solution among all the resulting solutions replaces the current solution.

1. $\Phi_0 \leftarrow$ Generate the initial population (m individuals)
2. Evaluate the population Φ_0
3. $t = 1$
4. Repeat
   a. $\Theta_{t-1} \leftarrow$ Select $n \le m$ individuals from Φ_{t-1}
   b. Estimate / learn a new model Π from Θ_{t-1}
   c. $\Phi_{new} \leftarrow$ Sample m individuals from Π
   d. Evaluate Φ_{new}
   e. $\Phi_t \leftarrow$ Select m individuals from $\Phi_{t-1} \cup \Phi_{new}$
   f. $t = t + 1$

   Until stop condition

Figure 4. Pseudo-code of General EDA.

1. Initialize a probability vector $\Pi = \{\pi_1, \pi_2, \ldots, \pi_N\}$ with 0.5 at each position. Here, each π_i represents the probability of 1 for the i^{th} position in the solution.
2. Generate population Φ of m solutions by sampling probabilities in Π
3. Select set Θ from Φ consisting of n promising solutions, where $n <= m$
4. Estimate univariate marginal probabilities $\Pi(\pi_i)$ for each π_i
5. For each i, update π_i using $\pi_i = \pi_i + \alpha\,(\Pi(\pi_i) - \pi_i)$
6. Go to step 2 until termination criteria are met.

Figure 6. The PBIL pseudo code.

However, in the recursive version of the 2-opt local search, if a better solution is found, it replaces the original solution and the search is restarted with the replaced solution as the central solution. This is continued till a solution with no better solution in its 2-opt neighborhood is found. In this paper, a simple as well as recursive versions of 2-opt local search denoted by S2opt and R2opt are applied.

4.3. The proposed algorithm for the optimal layout of keyboard

The proposed algorithms start with m random solutions. Each of the m solutions then undergoes 2-opt local search (S2-opt or R2-opt) and the resulting population is the initial population, $\Phi_0 = \{\pi^1{}_{(0)}, \pi^2{}_{(0)}, \pi^3{}_{(0)}, \ldots, \pi^N{}_{(0)}\}$. From Φ_0, n solutions are selected and the probability matrix Π_0 is calculated. For the next generation, the population, Φ_1 is modeled using Π_0. This process is repeated until a termination criterion is reached. There are two algorithms and two local searches, which result in four different algorithms, namely, UMDA-S2opt, UMDA-R2opt, PBILA-S2opt, and PBILA-R2opt. The four algorithms are coded in MATLAB and first tested for efficiency by solving benchmark problems taken from QAP Library [35]. The algorithms are then used to solve a test instance generated using the frequency list of English alphabets obtained from http://www.wiktionary.org [36] and seven famous books in English literature.

4.4. Parameter tuning

In order to find the best parameter combinations for the algorithms, a parametric optimization study has been conducted using Taguchi's robust design methodology. The parameters for the algorithms are number of maximum generations, population size, and elite population size for UMDA and maximum generations, population size, elite population size and learning rate (alpha) for PBILA. Since the local search does not have any parameter, only the base algorithm has to be optimized. For consistency, the parametric optimization is done for PBILA and the same values are applied for UMDA. The list of parameters with corresponding levels is given in Table 2.

alpha – learning rate, MG – maximum generation, PS – Population Size, EPS – Elite Population Size and Av – Average Solution

The orthogonal array for the DOE is L18 in which there is one parameter with six levels and all other three parameters with three levels each. In order to conduct the Taguchi's experiments an average-sized problem namely, esc16a from QAP Library is selected. Each experiment in the L18 array is done 10 times and the average is reported. Table 3 gives the result of the experiments.

Table 1. Summary of literature survey.

Author	Work	Methodology
Lesher and Moulton [1]	Optimization of s-finger keyboard	Swap heuristics
Eggers et al. [2]	Optimization of keyboard	Ant Colony Optimization
Cardinal and Langerman [3]	Incomplete keyboard design	
Clarkson et al. [4]	Mini-QWERTY keyboard design	Empirical study
Clawson et al. [5]	Mini-QWERTY keyboard learning rates, typing ability in limited visual feedback	Empirical study
Curran et al. [6]	Survey on preference of input devices on mobile devices	Survey
Clarkson et al. [7]	Mini-QWERTY keyboard validation	Empirical study
Kwon et al. [8]	Comparison of conventional finger touch text entry method and the regional error correction method in touch screen QWERTY keyboard	Survey
Yin and Su [9]	General keyboard arrangement	Cyber swarm
Behbahan [10]	Optimal layout for Farsi keyboard	Hybrid GA and SA
Alswaidan et al. [11]	Optimization of single finger Arabic keyboard layout	Genetic Algorithm
Govind and Panicker [12]	Optimization of s-finger keyboard problem	Genetic Algorithm and TOPSIS

Table 2. List of Parameters and levels.

Parameter	Number of Levels	Level Values
Learning rate (alpha)	6	0.1, 0.2, 0.3, 0.4, 0.5, 0.6
Maximum Generation	3	N*5, N*10, N*15
Population size (PS)	3	N/2, N, N*2
Elite population size	3	PS/4, PS/2, PS

Table 3. Result of Taguchi's experiments.

alpha	MG	PS	EPS	Av
0.1	1	1	1	81.8
0.1	2	2	2	81.2
0.1	3	3	3	81.2
0.2	1	1	2	76.6
0.2	2	2	3	72.6
0.2	3	3	1	78.6
0.3	1	2	1	77.2
0.3	2	3	2	76.8
0.3	3	1	3	83.6
0.4	1	3	3	75.6
0.4	2	1	1	75.4
0.4	3	2	2	74.8
0.5	1	2	3	84.6
0.5	2	3	1	82.8
0.5	3	1	2	79.2
0.6	1	3	2	80.0
0.6	2	1	3	78.8
0.6	3	2	1	82.4

From Figures 7 and 8, we can find out the optimal combination of parameters as alpha = level 5 = 0.4, maximum generation = level 2 = N*10, Population Size = level 2 = N and Elite Population Size = level 2 = N/2.

Table 4: Data for illustrations

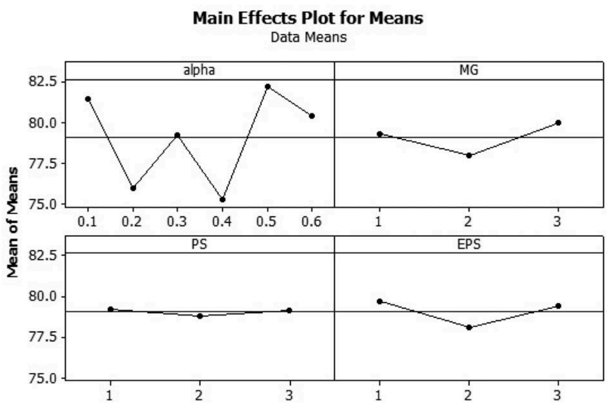

Figure 7. Main effects plot for means from Taguchi's experiments.

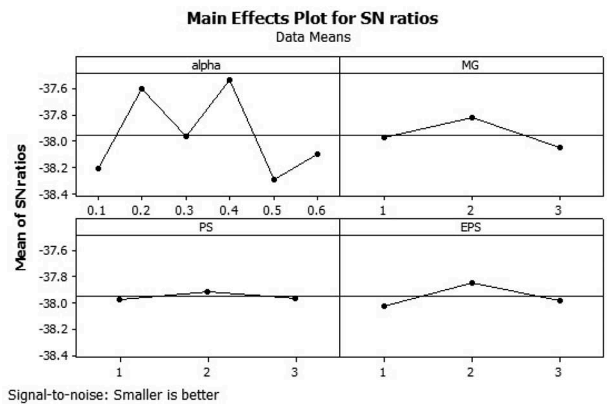

Figure 8. Main effects plot for SN ratios from Taguchi's experiments.

Table 4. Distance Matrix. Flow Matrix.

From \ To	1	2	3	4	5	6
1	0	1	2	1	2	3
2	1	0	1	2	1	2
3	2	1	0	3	2	1
4	1	2	3	0	1	2
5	2	1	2	1	0	1
6	3	2	1	2	1	0
1	0	90	689	194	165	494
2	668	0	1324	811	241	206
3	631	387	0	125	281	375
4	80	495	615	0	222	221
5	276	204	1127	490	0	676
6	109	409	1780	394	200	0

5. Illustrative example

For a step-by-step illustration of the proposed algorithms, we consider a QAP with six facilities. This problem is derived from the benchmark problems on Dynamic Facility Layout problems provided by Balakrishnan and Cheng [37]. Table 4 gives the distance matrix of the departments and the material handling costs between departments.

5.1. Working of the proposed UMDA with S2opt

(1) Generate population $\Phi_0 = \left\{\begin{matrix} 5\ 2\ 1\ 4\ 6\ 3 \\ 2\ 3\ 5\ 6\ 1\ 4 \\ \vdots \\ 3\ 5\ 6\ 2\ 1\ 4 \end{matrix}\right\}$ of m random solutions.

(2) Run S2opt local search on each of the m solutions.

$$\Phi_0 = \left\{\begin{matrix} 3\ 1\ 2\ 4\ 6\ 5 \\ 4\ 6\ 5\ 3\ 1\ 2 \\ \vdots \\ 6\ 5\ 2\ 4\ 1\ 3 \end{matrix}\right\}$$

(3) Calculate the objective function value as the material handling cost or transportation cost.

$$\Phi_{TC} = \left\{\begin{matrix} 20253 \\ 20253 \\ \vdots \\ 20361 \end{matrix}\right\}$$

(4) Select set Θ from Φ consisting of n promising solutions, where $n \le m$.

$$\Theta = \left\{\begin{matrix} 1\ 3\ 2\ 6\ 4\ 5 \\ 4\ 6\ 5\ 3\ 1\ 2 \\ \vdots \\ 6\ 4\ 5\ 1\ 3\ 2 \end{matrix}\right\}$$

(5) Estimate univariate marginal probabilities $\pi_{(i,\ j)}$ for each (i, j) using Θ

For calculating univariate marginal probabilities, convert each solution in Θ to the corresponding permutation matrix. For example, $[1\ 3\ 2\ 6\ 4\ 5] = \begin{bmatrix} 1&0&0&0&0&0 \\ 0&0&1&0&0&0 \\ 0&1&0&0&0&0 \\ 0&0&0&0&0&1 \\ 0&0&0&1&0&0 \\ 0&0&0&0&1&0 \end{bmatrix}$

Add the permutation matrices and divide it by n to get Π'

i.e. $\Pi' = \frac{1}{12}\begin{bmatrix} 2&0&5&2&0&3 \\ 5&0&2&3&0&2 \\ 0&7&0&0&5&0 \\ 3&0&2&5&0&2 \\ 2&0&3&2&0&5 \\ 0&5&0&0&7&0 \end{bmatrix}$

(6) Replace Π with Π' and generate new population based on Π.

(7) Repeat steps 2 to 6 until termination criteria are satisfied. The termination criteria selected is the number of iterations and its value is 10 × number of facilities.

(8) Save the solution with the lowest cost. The solution for the aforementioned problem is (1 3 2 5 6 4) with a total cost of 20,253.

5.2. Working of the proposed PBILA with S2opt

(1) Initialize a probability matrix Π as N x N matrix with all values set to $1/N$. Here, each (i, j) element in the matrix represents the probability that i^{th} department is located in j^{th} location.

$$\Pi = \begin{bmatrix} 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \end{bmatrix}$$

(2) Generate population $\Phi = \left\{ \begin{matrix} 5\ 2\ 1\ 4\ 6\ 3 \\ 2\ 3\ 5\ 6\ 1\ 4 \\ \vdots \\ 3\ 5\ 6\ 2\ 1\ 4 \end{matrix} \right\}$ of m solutions by sampling probabilities in Π.

(3) Run S2opt local search on each of the m solutions.

$$\Phi = \left\{ \begin{matrix} 3\ 1\ 2\ 4\ 6\ 5 \\ 4\ 6\ 5\ 3\ 1\ 2 \\ \vdots \\ 6\ 5\ 2\ 4\ 1\ 3 \end{matrix} \right\}$$

(4) Calculate the objective function value.

$$\Phi = \left\{ \begin{matrix} 20253 \\ 20253 \\ \vdots \\ 20361 \end{matrix} \right\}$$

(5) Select set Θ from Φ consisting of n promising solutions, where $n \leq p$.

$$\Theta = \left\{ \begin{matrix} 1\ 3\ 2\ 6\ 4\ 5 \\ 4\ 6\ 5\ 3\ 1\ 2 \\ \vdots \\ 6\ 4\ 5\ 1\ 3\ 2 \end{matrix} \right\}$$

(6) Estimate univariate marginal probabilities $\pi'_{(i,\ j)}$ for each (i, j) using Θ

For calculating univariate marginal probabilities, convert each solution in Θ to the corresponding permutation matrix. For example, $[1\ 3\ 2\ 6\ 4\ 5] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$

Add the permutation matrices and divide it by n to get Π'

i.e.

$$\Pi = \frac{1}{12} \begin{bmatrix} 2 & 0 & 5 & 2 & 0 & 3 \\ 5 & 0 & 2 & 3 & 0 & 2 \\ 0 & 7 & 0 & 0 & 5 & 0 \\ 3 & 0 & 2 & 5 & 0 & 2 \\ 2 & 0 & 3 & 2 & 0 & 5 \\ 0 & 5 & 0 & 0 & 7 & 0 \end{bmatrix}$$

(7) For each (i, j), update Π using $\pi_{(i,\ j)} = \pi_{(i,\ j)} + \alpha\ (\pi'_{(i,\ j)} - \pi_{(i,\ j)})$

$$\Pi = \begin{bmatrix} \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} \\ \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} \\ \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} \\ \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} \\ \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} \\ \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} \end{bmatrix} + 0.4 \left(\begin{bmatrix} \frac{1}{6} & 0 & \frac{5}{12} & \frac{1}{6} & 0 & \frac{1}{4} \\ \frac{5}{12} & 0 & \frac{1}{6} & \frac{1}{4} & 0 & \frac{1}{6} \\ 0 & \frac{7}{12} & 0 & 0 & \frac{5}{12} & 0 \\ \frac{1}{4} & 0 & \frac{1}{6} & \frac{5}{12} & 0 & \frac{1}{6} \\ \frac{1}{6} & 0 & \frac{1}{4} & \frac{1}{6} & 0 & \frac{5}{12} \\ 0 & \frac{5}{12} & 0 & 0 & \frac{7}{12} & 0 \end{bmatrix} - \begin{bmatrix} \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} \\ \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} \\ \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} \\ \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} \\ \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} \\ \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} \end{bmatrix} \right)$$

$$= \begin{bmatrix} \frac{1}{6} & \frac{1}{10} & \frac{4}{15} & \frac{1}{6} & \frac{1}{10} & \frac{1}{5} \\ \frac{4}{15} & \frac{1}{10} & \frac{1}{6} & \frac{1}{5} & \frac{1}{10} & \frac{1}{6} \\ \frac{1}{10} & \frac{1}{3} & \frac{1}{10} & \frac{1}{10} & \frac{4}{15} & \frac{1}{10} \\ \frac{1}{5} & \frac{1}{10} & \frac{1}{6} & \frac{4}{15} & \frac{1}{10} & \frac{1}{6} \\ \frac{1}{6} & \frac{1}{10} & \frac{1}{5} & \frac{1}{6} & \frac{1}{10} & \frac{4}{15} \\ \frac{1}{10} & \frac{4}{15} & \frac{1}{10} & \frac{1}{10} & \frac{1}{3} & \frac{1}{10} \end{bmatrix}$$

(8) Repeat steps 2 to 7 until termination criteria are satisfied. Repeat steps 2 to 6 until termination criteria are satisfied. The termination criteria selected is the number of iterations and its value is 10 × number of facilities.

(9) Save the solution with the lowest cost. The solution for the aforementioned problem is (1 3 2 5 6 4) with a total cost of 20,253.

The logic of UMDA R2opt and PBILA R2opt can be derived from the aforementioned illustrations by replacing the S2opt local search with R2opt local search method.

6. Computational study

The computational experiments are carried out in three stages. First of all, the efficiency of the proposed algorithms is demonstrated by solving benchmark instances taken from QAP Library. In the second stage, the algorithms are applied to determine new layouts for the s-finger keyboard by solving a test instance generated from frequency list of letter pairs for English language. Finally, from the stage two, a keyboard with optimal layouts is obtained and experiments are conducted to validate the results obtained. The detailed description of each of these stages is given below.

6.1. Results and analysis – stage 1 experiments

To demonstrate the efficiency of the proposed algorithms, we consider 57 problems of size varying from 12 to 32 from QAP Library. All the problems were solved using the four proposed algorithms. Each problem was solved by each algorithm 10 times and the best, worst, and average solutions are reported. Table 5 gives the results of the efficiency tests along with the problem name and known best solution.

Table 6 gives the percentage variation of the best, worst, and average solutions from the known best solution. From the results presented in Tables

Table 5. Best, worst, and average solutions for the benchmark problems.

			PBIL-S2opt			PBIL-R2opt			UMDA-S2opt			UMDA-R2opt		
Sl. No.	Problem	Known Minimum	Best	Worst	Average	Best	Worst	Average	Best	Worst	Average	Best	Worst	Average
1	bur26a	5,426,670	5,426,670	5,450,502	5,438,540	5,435,268	5,436,784	5,435,706	5,428,397	5,454,119	5,441,013	5,434,116	5,436,784	5,435,032
2	bur26b	3,817,852	3,818,577	3,836,540	3,828,302	3,817,852	3,825,757	3,824,538	3,826,845	3,837,964	3,829,658	3,824,564	3,825,782	3,825,219
3	bur26c	5,426,795	5,427,230	5,439,926	5,430,426	5,426,993	5,428,090	5,427,197	5,426,801	5,467,484	5,441,433	5,426,993	5,428,356	5,427,565
4	bur26d	3,821,225	3,821,772	3,828,872	3,822,811	3,821,225	3,821,779	3,821,336	3,821,555	3,847,474	3,828,225	3,821,225	3,821,225	3,821,225
5	bur26e	5,386,879	5,387,166	5,412,911	5,391,406	5,386,879	5,387,318	5,387,135	5,388,192	5,446,720	5,400,593	5,386,879	5,388,258	5,387,556
6	bur26f	3,782,044	3,782,577	3,796,053	3,784,327	3,782,044	3,782,044	3,782,044	3,782,702	3,818,797	3,794,432	3,782,044	3,782,068	3,782,046
7	bur26g	10,117,172	10,118,819	10,162,248	10,128,170	10,117,172	10,118,177	10,117,664	10,118,812	10,164,318	10,139,818	10,117,172	10,118,634	10,118,070
8	bur26h	7,098,658	7,099,677	7,145,479	7,112,961	7,098,658	7,098,658	7,098,658	7,099,218	7,159,636	7,121,299	7,098,658	7,099,677	7,098,966
9	esc16a	68	68	72	69	68	68	68	68	72	69	68	68	68
10	esc16b	292	292	292	292	292	292	292	292	292	292	292	292	292
11	esc16c	160	160	160	160	160	160	160	160	166	161	160	160	160
12	esc16d	16	16	18	17	16	16	16	16	18	16	16	16	16
13	esc16e	28	28	32	29	28	28	28	28	34	31	28	30	29
14	esc16g	26	26	28	27	26	26	26	26	28	27	26	26	26
15	esc16h	996	996	996	996	996	996	996	996	996	996	996	996	996
16	esc16i	14	14	14	14	14	14	14	14	14	14	14	14	14
17	esc16j	8	8	8	8	8	8	8	8	12	9	8	8	8
18	esc32a	130	140	154	149	130	144	136	146	168	154	130	146	139
19	esc32b	168	188	216	199	168	192	184	188	248	209	168	196	184
20	esc32c	642	642	646	642	642	642	642	642	646	642	642	642	642
21	esc32d	200	200	216	205	200	202	200	200	218	206	200	208	201
22	esc32e	2	2	2	2	2	2	2	2	2	2	2	2	2
23	esc32g	6	6	6	6	6	6	6	6	6	6	6	6	6
24	esc32h	438	438	446	442	438	440	438	438	464	446	438	440	440
25	had12	1652	1652	1678	1665	1652	1660	1654	1652	1692	1674	1652	1660	1654
26	had14	2724	2724	2744	2735	2724	2724	2724	2724	2780	2742	2724	2730	2725
27	had16	3720	3720	3774	3736	3720	3722	3721	3722	3796	3740	3720	3722	3720
28	had18	5358	5366	5420	5389	5358	5370	5364	5360	5422	5395	5358	5370	5365
29	had20	6922	6930	6968	6956	6922	6948	6930	6930	7086	6999	6922	6958	6927

			PBIL-S2opt			PBIL-R2opt			UMDA-S2opt			UMDA-R2opt		
Sl. No.	Problem	Known Minimum	Best	Worst	Average	Best	Worst	Average	Best	Worst	Average	Best	Worst	Average
30	kra30a	88,900	91,500	94,850	93,418	88,900	91,070	90,140	91,750	95,710	93,674	88,900	92,320	90,654
31	kra30b	91,420	92,520	97,330	95,231	91,490	92,410	92,028	92,590	97,100	94,241	91,490	92,880	92,162
32	kra32	88,700	90,600	94,780	93,190	88,700	91,220	89,613	90,670	96,510	94,056	88,700	91,250	90,318
33	lipa20a	3683	3715	3784	3759	3683	3753	3702	3760	3797	3778	3683	3763	3712
34	lipa20b	27,076	27,076	31,326	29,754	27,076	27,076	27,076	27,076	31,573	30,226	27,076	27,076	27,076
35	lipa30a	13,178	13,394	13,426	13,408	13,178	13,178	13,178	13,377	13,454	13,418	13,178	13,355	13,196
36	lipa30b	151,426	151,426	176,772	165,675	151,426	151,426	151,426	151,426	177,378	171,257	151,426	151,426	151,426
37	nug12	578	586	608	593	586	588	586	590	626	602	586	594	589
38	nug14	1014	1016	1062	1038	1014	1032	1019	1018	1062	1039	1014	1034	1024
39	nug15	1150	1152	1202	1182	1150	1174	1155	1160	1224	1182	1150	1174	1157
40	nug16a	1610	1622	1708	1647	1610	1622	1614	1632	1718	1660	1610	1638	1629
41	nug16b	1240	1240	1350	1292	1240	1240	1240	1266	1320	1300	1240	1268	1247
42	nug17	1732	1742	1796	1776	1734	1748	1742	1760	1826	1792	1732	1758	1742

(*Continued*)

Table 5. (Continued).

Sl. No.	Problem	Known Minimum	PBIL-S2opt			PBIL-R2opt			UMDA-S2opt			UMDA-R2opt		
			Best	Worst	Average	Best	Worst	Average	Best	Worst	Average	Best	Worst	Average
43	nug18	1930	1948	2016	1981	1938	1966	1950	1982	2050	2002	1930	1954	1942
44	nug20	2570	2570	2686	2652	2570	2598	2579	2620	2692	2658	2570	2604	2588
45	nug21	2438	2444	2558	2501	2444	2476	2460	2476	2570	2516	2442	2482	2457
46	nug22	3596	3606	3782	3673	3596	3652	3615	3656	3816	3707	3596	3666	3618
47	nug24	3488	3560	3676	3622	3488	3520	3503	3542	3720	3617	3488	3538	3511
48	nug25	3744	3752	3914	3849	3744	3782	3754	3788	3930	3854	3744	3774	3753
49	nug27	5234	5246	5404	5335	5234	5304	5254	5234	5372	5316	5234	5338	5273
50	nug28	5166	5176	5392	5302	5176	5238	5214	5234	5456	5325	5176	5274	5211
51	nug30	6124	6162	6390	6293	6128	6168	6146	6218	6442	6330	6124	6192	6155
52	rou12	235,528	238,134	248,590	243,520	235,528	240,906	237,849	235,528	249,738	242,443	235,528	243,498	238,499
53	rou15	354,210	362,518	377,932	369,004	354,210	364,746	358,555	362,518	384,274	372,592	354,210	371,322	361,692
54	rou20	725,522	728,724	759,424	743,383	728,512	737,734	732,042	747,980	762,236	754,088	726,920	736,602	731,398
55	scr12	31,410	31,410	33,310	32,324	31,410	31,410	31,410	31,410	35,436	33,154	31,410	32,696	31,831
56	scr15	51,140	51,140	57,798	54,865	51,140	53,826	51,409	53,808	58,476	55,500	51,140	55,460	52,577
57	scr20	110,030	113,768	117,768	115,611	110,030	112,156	110,828	113,452	123,716	117,544	110,030	113,084	111,214

5 and 6, it is evident that the proposed algorithms perform well while solving QAP. Out of the 57 problems solved using UMDA-S2opt, for 21 problems, the solution obtained is the same as that of the best-known solution. The corresponding number for UMDA-R2opt is 49. For PBILA-S2opt and PBILA-R2opt, the corresponding numbers are 24 and 47, respectively. For most problems which resulted in inferior solutions, the variation from known optimal solution is less than two percentage, which shows the efficiency of the proposed algorithms. Out of the four algorithms proposed, those with the R2opt local search perform better compared to those with S2opt local search.

6.2. Results and analysis – stage 2 experiments

We generated a test instance for determining a better layout for the s-finger keyboards. The test instance used in this study is the frequency lists of letter pairs taken from seven well-known books in English literature and the 10,000 most frequently used words in English language obtained from http://www.wiktionary.org. We have merged the frequency lists of all the seven books and the frequently used words to form a single list, which invariably represents the frequency list for English language. The books considered are as follows.

- The Alchemist by Paulo Coelho
- Alice's Adventures in Wonderland by Lewis Carroll
- Animal Farm by George Orwell
- Anna Karenina by Leo Tolstoy
- Othello by William Shakespeare
- The Odyssey by Homer
- Time Machine by H. G. Wells

These books are available to download in text format from https://www.gutenberg.org/[38]. The frequency list of letter pairs are calculated for each book separately and then all are added together to get the final list. While finding out the frequencies, punctuations and new paragraph characters are removed for simplicity. A script was coded in MATLAB to find out the frequency list of letter pairs of the selected books and words. Table 7 gives the frequency table for letter pairs.

The Fitts's times for the test instances in QWERTY keyboard and the ABC keyboard are given as 2,070,670.2 and 2,097,922.2, respectively. The corresponding Euclidean distances are 27,272,066.6 and 28,950,018.9, respectively. The results obtained for the different algorithms for the objective of minimizing the Fitts's time and the Euclidean distance are given in Table 8.

Table 6. Percentage deviation of best, worst, and average solutions from the known optimal solutions for the benchmark problems.

			PBIL-S2opt			PBIL-R2opt			UMDA-S2opt			UMDA-R2opt		
Sl. No.	Problem	Known Minimum	Best	Worst	Average	Best	Worst	Average	Best	Worst	Average	Best	Worst	Average
1	bur26a	5,426,670	0.00	0.44	0.22	0.16	0.19	0.17	0.03	0.51	0.26	0.14	0.19	0.15
2	bur26b	3,817,852	0.02	0.49	0.27	0.00	0.21	0.18	0.24	0.53	0.31	0.18	0.21	0.19
3	bur26c	5,426,795	0.01	0.24	0.07	0.00	0.02	0.01	0.00	0.75	0.27	0.00	0.03	0.01
4	bur26d	3,821,225	0.01	0.20	0.04	0.00	0.01	0.00	0.01	0.69	0.18	0.00	0.00	0.00
5	bur26e	5,386,879	0.01	0.48	0.08	0.00	0.01	0.00	0.02	1.11	0.25	0.00	0.03	0.01
6	bur26f	3,782,044	0.01	0.37	0.06	0.00	0.00	0.00	0.02	0.97	0.33	0.00	0.00	0.00
7	bur26g	10,117,172	0.02	0.45	0.11	0.00	0.01	0.00	0.02	0.47	0.22	0.00	0.01	0.01
8	bur26h	7,098,658	0.01	0.66	0.20	0.00	0.00	0.00	0.01	0.86	0.32	0.00	0.01	0.00
9	esc16a	68	0.00	5.88	0.88	0.00	0.00	0.00	0.00	5.88	0.88	0.00	0.00	0.00
10	esc16b	292	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
11	esc16c	160	0.00	0.00	0.00	0.00	0.00	0.00	0.00	3.75	0.63	0.00	0.00	0.00
12	esc16d	16	0.00	12.50	3.75	0.00	0.00	0.00	0.00	12.50	1.25	0.00	0.00	0.00
13	esc16e	28	0.00	14.29	4.29	0.00	0.00	0.00	0.00	21.43	9.29	0.00	7.14	2.14
14	esc16g	26	0.00	7.69	2.31	0.00	0.00	0.00	0.00	7.69	2.31	0.00	0.00	0.00
15	esc16h	996	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
16	esc16i	14	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
17	esc16j	8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	50.00	15.00	0.00	0.00	0.00
18	esc32a	130	7.69	18.46	14.62	0.00	10.77	4.62	12.31	29.23	18.15	0.00	12.31	6.77
19	esc32b	168	11.90	28.57	18.33	0.00	14.29	9.52	11.90	47.62	24.29	0.00	16.67	9.29
20	esc32c	642	0.00	0.62	0.06	0.00	0.00	0.00	0.00	0.62	0.06	0.00	0.00	0.00
21	esc32d	200	0.00	8.00	2.70	0.00	1.00	0.10	0.00	9.00	3.20	0.00	4.00	0.50
22	esc32e	2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
23	esc32g	6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
24	esc32h	438	0.00	1.83	1.00	0.00	0.46	0.09	0.00	5.94	1.78	0.00	0.46	0.41
25	had12	1652	0.00	1.57	0.81	0.00	0.48	0.15	0.00	2.42	1.31	0.00	0.48	0.13
26	had14	2724	0.00	0.73	0.40	0.00	0.00	0.00	0.00	2.06	0.65	0.00	0.22	0.02
27	had16	3720	0.00	1.45	0.44	0.00	0.05	0.02	0.05	2.04	0.55	0.00	0.05	0.01
28	had18	5358	0.15	1.16	0.57	0.00	0.22	0.11	0.04	1.19	0.70	0.00	0.22	0.13
29	had20	6922	0.12	0.66	0.50	0.00	0.38	0.11	0.12	2.37	1.11	0.00	0.52	0.08
30	kra30a	88,900	2.92	6.69	5.08	0.00	2.44	1.39	3.21	7.66	5.37	0.00	3.85	1.97
31	kra30b	91,420	1.20	6.46	4.17	0.08	1.08	0.67	1.28	6.21	3.09	0.08	1.60	0.81
32	kra32	88,700	2.14	6.85	5.06	0.00	2.84	1.03	2.22	8.80	6.04	0.00	2.87	1.82
33	lipa20a	3683	0.87	2.74	2.06	0.00	1.90	0.50	2.09	3.10	2.57	0.00	2.17	0.79
34	lipa20b	27,076	0.00	15.70	9.89	0.00	0.00	0.00	0.00	16.61	11.63	0.00	0.00	0.00
35	lipa30a	13,178	1.64	1.88	1.75	0.00	0.00	0.00	1.51	2.09	1.82	0.00	1.34	0.13
36	lipa30b	151,426	0.00	16.74	9.41	0.00	0.00	0.00	0.00	17.14	13.10	0.00	0.00	0.00
37	nug12	578	1.38	5.19	2.66	1.38	1.73	1.45	2.08	8.30	4.19	1.38	2.77	1.83
38	nug14	1014	0.20	4.73	2.39	0.00	1.78	0.49	0.39	4.73	2.47	0.00	1.97	1.03
39	nug15	1150	0.17	4.52	2.80	0.00	2.09	0.40	0.87	6.43	2.77	0.00	2.09	0.63
40	nug16a	1610	0.75	6.09	2.30	0.00	0.75	0.24	1.37	6.71	3.12	0.00	1.74	1.17
41	nug16b	1240	0.00	8.87	4.18	0.00	0.00	0.00	2.10	6.45	4.87	0.00	2.26	0.53
42	nug17	1732	0.58	3.70	2.54	0.12	0.92	0.59	1.62	5.43	3.45	0.00	1.50	0.57
43	nug18	1930	0.93	4.46	2.64	0.41	1.87	1.04	2.69	6.22	3.74	0.00	1.24	0.63
44	nug20	2570	0.00	4.51	3.18	0.00	1.09	0.34	1.95	4.75	3.43	0.00	1.32	0.69
45	nug21	2438	0.25	4.92	2.59	0.25	1.56	0.92	1.56	5.41	3.19	0.16	1.80	0.79
46	nug22	3596	0.28	5.17	2.15	0.00	1.56	0.54	1.67	6.12	3.10	0.00	1.95	0.62
47	nug24	3488	2.06	5.39	3.85	0.00	0.92	0.44	1.55	6.65	3.71	0.00	1.43	0.66
48	nug25	3744	0.21	4.54	2.81	0.00	1.01	0.27	1.18	4.97	2.94	0.00	0.80	0.23
49	nug27	5234	0.23	3.25	1.93	0.00	1.34	0.39	0.00	2.64	1.57	0.00	1.99	0.74
50	nug28	5166	0.19	4.37	2.63	0.19	1.39	0.93	1.32	5.61	3.07	0.19	2.09	0.87
51	nug30	6124	0.62	4.34	2.75	0.07	0.72	0.35	1.53	5.19	3.36	0.00	1.11	0.50
52	rou12	235,528	1.11	5.55	3.39	0.00	2.28	0.99	0.00	6.03	2.94	0.00	3.38	1.26
53	rou15	354,210	2.35	6.70	4.18	0.00	2.97	1.23	2.35	8.49	5.19	0.00	4.83	2.11
54	rou20	725,522	0.44	4.67	2.46	0.41	1.68	0.90	3.10	5.06	3.94	0.19	1.53	0.81
55	scr12	31,410	0.00	6.05	2.91	0.00	0.00	0.00	0.00	12.82	5.55	0.00	4.09	1.34
56	scr15	51,140	0.00	13.02	7.28	0.00	5.25	0.53	5.22	14.34	8.53	0.00	8.45	2.81
57	scr20	110,030	3.40	7.03	5.07	0.00	1.93	0.73	3.11	12.44	6.83	0.00	2.78	1.08

The layout corresponding to minimum value for Fitts's time is represented as ***L1*** and the layout corresponding to the minimum value for Euclidean distance is represented as ***L2*** in Table 8. The corresponding layouts are shown in Figures 9 and 10, respectively.

Table 9 shows the comparison of keyboard layouts ***L1*** and ***L2*** with QWERTY and ABC layouts in terms of Fitts's time and total Euclidean distance moved by the finger.

For both these cases, the Fitts's time as well as the Euclidean distance is very much lesser than that of QWERTY keyboard and ABC keyboard. This means that theoretically there is a chance of improvement in typing speed for single finger used text entry, if we change the layout of the keyboard. In order to validate this finding, we conducted physical experiments also, the details of which are presented in the following sub-section.

Table 7. The combined frequency list for letter pairs.

	A	B	C	D	E	F	G	H	I	J	K	L	M
A	48	9272	16,345	26,441	596	3809	8831	463	22,014	148	7132	37,127	12,123
B	5901	433	16	50	30,663	0	0	5	2703	699	0	9307	91
C	17,142	0	2554	11	25,362	0	0	26,642	6675	0	6650	5048	4
D	6655	10	13	2379	29,747	201	1000	68	16,760	50	84	1983	661
E	37,763	724	13,760	56,943	23,025	7684	4074	1288	9292	188	694	25,875	15,912
F	9102	1	0	6	11,865	5603	0	1	9910	0	0	2796	2
G	7370	0	4	81	14,440	5	896	16,750	5799	0	2	3199	122
H	66,117	156	12	100	191,520	194	0	12	58,786	1	76	397	468
I	6418	2668	19,249	18,906	12,416	9173	11,606	163	104	1	3233	20,514	20,024
J	629	0	1	0	1391	0	0	0	51	0	0	0	0
K	1628	4	4	1	14,524	89	24	10	7330	0	3	520	73
L	18,523	138	525	18,261	39,801	4709	140	10	24,645	0	1987	34,507	971
M	23,604	2716	16	6	40,040	285	0	35	12,168	0	0	201	2264
N	8670	426	13,814	80,927	32,956	1844	52,920	338	10,871	452	3917	3752	365
O	2560	2815	3793	7868	1496	53,606	1693	790	4544	98	6140	12,418	25,957
P	11,296	30	0	0	18,160	7	11	1913	5517	0	15	9739	133
Q	0	0	0	0	0	0	0	0	0	0	0	0	0
R	18,681	685	3467	9735	80,110	1227	3324	536	24,523	8	4477	3684	5358
S	17,891	788	4269	412	43,671	603	188	24,841	18,467	4	4075	2956	2475
T	17,528	35	3384	9	44,032	377	5	198,145	32,778	0	21	6657	538
U	3150	2657	6443	2958	4785	737	9249	41	4155	0	184	18,122	3377
V	4365	0	0	2	42,808	0	7	0	11,138	0	3	31	0
W	31,203	144	4	287	21,344	222	1	27,766	25,998	0	86	685	2
X	1443	0	1294	0	1403	29	0	127	816	0	0	1	0
Y	1124	241	150	60	6637	60	25	30	2127	0	0	316	518
Z	255	1	0	14	1274	0	0	281	277	0	0	104	22
Sp	151,229	62,290	50,019	40,867	27,441	53,813	26,018	111,551	95,540	4781	9429	36,444	58,164

	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	Sp
A	107,127	116	7123	0	43,499	49,767	62,928	4879	13,023	4926	282	14,983	667	16,787
B	9	9951	0	0	5641	1291	820	11,655	140	3	0	7505	0	551
C	7	26,891	0	274	4827	152	10,661	4288	0	0	0	883	0	1901
D	1053	13,173	12	14	6188	6311	44	3000	679	198	0	3168	3	163,815
E	59,585	2533	7665	1077	99,641	44,218	19,175	809	14,294	5008	7966	10,733	150	274,396
F	48	23,899	6	0	10,799	158	4870	4876	0	2	0	207	0	59,315
G	1567	10,352	1	0	7811	2813	495	2942	4	0	0	346	2	41,311
H	362	29,757	1	8	3918	627	11,440	4353	69	90	0	2090	0	39,691
I	113,104	18,438	2440	110	15,256	51,511	57,140	431	9557	2	671	0	1068	20,300
J	0	1871	0	0	9	0	0	2540	0	0	0	0	0	0
K	5044	614	0	124	49	1940	8	48	2	84	0	1784	0	12,987
L	167	19,347	795	0	648	3968	4015	3267	1346	1083	0	21,130	1	34,750
M	443	15,494	6443	0	1706	4116	46	5014	0	0	0	7373	0	24,752
N	3979	30,576	127	341	170	14,781	34,899	2182	1446	273	249	5468	96	105,229
O	64,485	16,492	7532	60	52,343	11,891	25,533	66,017	10,455	22,736	427	2168	271	69,319
P	2	12,323	5810	0	13,127	1932	3370	3694	0	116	0	1023	0	8013
Q	0	0	0	0	0	0	0	5385	0	0	0	0	0	77
R	6389	30,306	1243	24	5407	16,697	13,842	5013	2203	669	0	12,387	3	81,239
S	1247	20,166	7853	280	73	16,746	43,810	11,131	147	2405	0	1094	0	136,313
T	402	59,621	23	0	14,254	9155	10,701	8064	63	3360	0	7556	45	142,154
U	17,352	392	8493	6	22,955	21,514	25,440	0	9	0	73	137	116	11,988
V	572	2574	0	0	883	213	24	69	0	0	0	278	0	339
W	5990	14,480	0	0	1627	1606	49	35	0	0	0	114	0	14,607
X	0	32	2142	72	0	2	1399	81	5	20	6	8	0	741
Y	76	18,929	426	0	302	4608	1407	4	1	227	2	0	12	76,506
Z	26	93	0	0	0	0	0	37	8	1	0	63	133	35
Sp	32,698	97,513	38,742	3035	27,370	102,226	231,449	16,169	9263	104,611	39	20,738	78	0

Table 8. Results of running algorithms for minimizing Fitts's time and Euclidean distance.

		Fitts's Time			Euclidean Distance		
		Min	Max	Average	Min	Max	Average
PBIL	**S2opt**	1,761,417.3	1,771,456.3	1,764,971.7	17,065,804.4	17,496,388.6	17,262,091.3
	R2opt	**1,756,435.4(*L1*)**	1,760,777.2	1,757,498.7	16,945,647.1	17,036,164.7	16,986,735.7
UMDA	**S2opt**	1,761,417.3	1,768,580.3	1,764,136.3	17,020,481.0	17,430,285.5	17,213,642.2
	R2opt	1,757,461.0	1,762,283.5	1,758,975.5	**16,894,181.5(*L2*)**	17,037,802.4	16,964,732.0

6.3. Results and analysis – stage 3 validation experiments

In order to validate the effectiveness of the new layouts, we conducted experiments by rearranging physical keyboards conforming to the obtained new layouts using Keytweak 2.3 software, which is free for use. The text used for the experiments is a small poem "Stopping by Woods on a Snowy Evening" by Robert Frost. The punctuation marks and newline characters were removed from the poem and only the alphabets and spaces were keyed in during the experiment. The experiments were done in two steps, one with a person who is

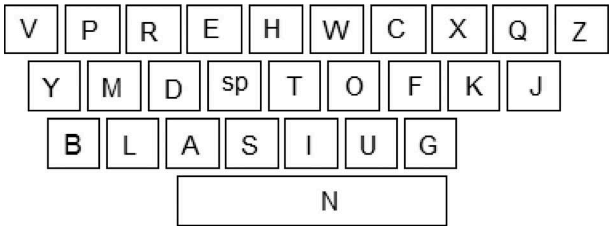

Figure 9. Layout giving the best Fitts's time (*L1*).

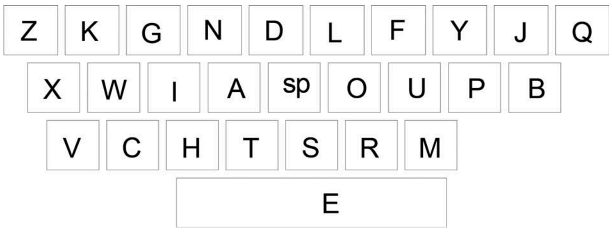

Figure 10. Layout giving the best Euclidean distance (*L2*).

Table 9. Comparison of selected layouts with QWERTY and ABC layouts.

Layout	Fitts's Time	Euclidean Distance
L1	1,756,435.4	17,309,171.3
L2	1,758,936.3	16,894,181.5
QWERTY	2,070,670.3	27,272,066.6
ABC	2,097,922.2	28,950,018.9

familiar with computers and is using the QWERTY keyboard for quite a long time, and other with a child who is not using computers and hence has no idea about the layout of the keyboards. The first person was given the new layouts and allowed to practice on the new keyboards for some time to make the layout somewhat familiar. After this familiarization period, the time for typing is noted and these values are shown in Table 10.

The time for typing using the QWERTY keyboard, in this case, is much lower than that for other layouts due to the fact that the person who did the experiments has been using the keyboard for a long time and is familiar with it.

In order to negate the effect of familiarity with QWERTY keyboard, we repeated the experiments with a 10-year-old child who is not familiar with computers. In this case also, the child was given time to practice in a layout for some time. The typing times obtained are shown in Table 11.

It is evident that the layouts ***L1*** and ***L2*** involve lesser time when compared with QWERTY and ABC layouts.

Table 10. Typing time of the selected text in seconds (QWERTY is familiar).

Exp. No.	Layout L1	Layout L2	QWERTY Layout	ABC Layout
1	445	553	200	759
2	442	433	209	647
3	406	425	260	591
4	386	402	196	564
5	353	369	270	513
6	341	362	260	476
7	330	351	194	464
8	326	356	230	446
9	312	354	210	439
10	319	347	250	442
Average	**366**	**395**	**228**	**534**

Table 11. Typing time of the selected text in seconds (QWERTY is not familiar).

Exp. No.	Layout L1	Layout L2	QWERTY Layout	ABC Layout
1	593	649	710	721
2	584	631	699	715
3	580	626	693	709
4	577	595	675	667
5	502	592	665	650
6	486	588	654	627
7	472	587	636	602
8	468	570	598	600
9	460	531	586	584
10	456	514	565	555
Average	**518**	**588**	**648**	**643**

7. Conclusions

The popular electronic text entry devices, such as QWERTY and Dvorak keyboards, are optimized for 10 finger usages. Handheld electronic devices such as PDAs, smartphones, and Tablet PCs require revised layouts for the on-screen keyboard. In this work, new layouts for s-finger typing are introduced. Optimization algorithms combining EDAs and local search methods are developed for solving the QAPs. The effectiveness of the algorithms is verified using 57 benchmark instances of QAP taken from QAP Library. A test instance is developed using seven popular literary works in English and the word frequency list for English language. Based results of optimization of the test instances, two layouts are selected. The proposed layouts are validated by keying in a popular poem, "Stopping by Woods on a Snowy Evening" by Robert Frost.

From the results obtained, it is evident that the QWERTY and ABC layouts for the keyboards are not suitable for applications where s-finger typing is involved. These layouts are suitable for applications where all the 10 fingers are employed for pressing the keys. In devices such as mobile phones, PDAs, and Tablet PCs, it is better to use an improved layout to reduce the typing time using only a single finger. The proposed layouts ***L1*** and ***L2*** are promising layouts for the aforementioned devices. Moreover, the rearranging of keys does not require any effort on these devices and for each language used, the layout can be different also.

The research reported in this paper is limited to solve the single keyboard layout problem using only 27 keys, i.e. all the 26 alphabets and the space key. The present research can be extended for solving keyboard layout problem including other keys, such as numerals and symbols. Another research area could be incorporating multiple fingers of the same hand while holding the device in the other hand, which require more ergonomic aspects also to be considered.

Acknowledgments

The authors are thankful to the referees and the editor for their constructive suggestions and comments which have immensely helped to bring this paper to the present form.

Disclosure statement

No potential conflict of interest was reported by the authors.

Notes on contributors

T.G. Pradeepmon is a research scholar in the Department of Mechanical Engineering. He is pursuing research on the development of heuristics for solving QAPs. His areas of interest include supply chain management, combinatorial optimization, metaheuristics, and simulation.

Vinay V. Panicker is an Assistant Professor from the National Institute of Technology Calicut (NITC), Kerala, India, where he teaches courses on Supply Chain Management, Work System Design, and Statistics for Management. He has done research in the area of supply chain management for his doctoral degree from the NITC. His research interests focus in the areas of supply chain management, logistics management, and ergonomics. He has published technical papers in the referred international journals and proceedings of international and national conferences.

R. Sridharan is a Professor of Industrial Engineering in the Department of Mechanical Engineering at NITC, India. He received his PhD in 1995 from the Department of Mechanical Engineering from the Indian Institute of Technology, Bombay, India. His research interests include modeling and analysis of decision problems in supply chain management, job shop production systems, and flexible manufacturing systems. He has published papers in the referred international journals and proceedings of international and national conferences. He has been conferred with the Fellowship award by the National Council of Indian Institution of Industrial Engineering in 2017 for the outstanding contribution to the field of Industrial Engineering and the Institution.

ORCID

T.G. Pradeepmon http://orcid.org/0000-0002-1524-5831
Vinay V. Panicker http://orcid.org/0000-0003-2167-3653
R. Sridharan http://orcid.org/0000-0002-0186-6442

References

[1] Liebowitz SJ, Margolis SE. The fable of the keys. J Law Econ. 1990;23:1–26.
[2] David PA. Clio and the Economics of QWERTY. Am Econ Rev. 1985;75:332–7.
[3] Lewin P. The market process and the economics of QWERTY : two views. The Review of Austrian Economics. 2001;14:65–96.
[4] Eggers J, Feillet D, Kehl S, Wagner MO, Yannou B. Optimization of the keyboard arrangement problem using an Ant Colony algorithm. Eur J Oper Res. 2003;148:672–686.
[5] Kwon S, Lee D, Chung MK. Effect of key size and activation area on the performance of a regional error correction method in a touch-screen QWERTY keyboard. Int J Ind Ergon. 2009;39:888–893.
[6] Cardinal J, Langerman S. Designing small keyboards is hard. Theor Comput Sci. 2005;332:405–415.
[7] Curran K, Woods D, Riordan BO. Investigating text input methods for mobile phones. Telematics and Informatics. 2006; 23(1): 1–21.
[8] Yin P-Y, Su E-P. Cyber Swarm optimization for general keyboard arrangement problem. Int J Ind Ergon. 2011;41:43–52.
[9] Alswaidan N, Hosny MI, Najjar AB. A genetic algorithm approach for optimizing a single-finger arabic keyboard layout. In: Arai K, Kapoor S, Bhatia R, editors. Intelligent systems in science and information. London: Springer; 2014. p. 261–277.
[10] Khorshid E, Alfadli A, Majeed M. A new optimal Arabic keyboard layout using genetic algorithm. Int J Des Eng. 2010;3:25–40.
[11] Malas TM, Taifour SS, Abandah GA," Toward optimal Arabic keyboard layout using genetic algorithm," *Proceedings of the 9th International Middle Eastern Multiconference on Simulation and Modeling (MESM 2008)*, Aug 26-28, Amman, Jordan, 1–5 (2008).
[12] Govind M, Panicker VV. Optimization of a single finger keyboard layout using genetic algorithm and TOPSIS. Int J Scientific Eng Res. 2016;7:102–5.
[13] Amico MD, Diaz JCD, Iori M, Montanari R. The single-finger keyboard layout problem. Computers and Operations Research. 2009;36:3002–12.
[14] Sahni S, Gonzalez T. P-Complete approximation problems. J Assoc Comput Machinery. 1976;23:555–65.
[15] McKendall A, Li C. A tabu search heuristic for a generalized quadratic assignment problem. J Ind Production Eng. 2016;34:221–31.
[16] Tosun U, Dokeroglu T, Cosar A. A robust island parallel genetic algorithm for the quadratic assignment problem. Int J Production Res. 2013;51:4117–33.
[17] Hafiz F, Abdennour A. Particle Swarm Algorithm variants for the Quadratic Assignment Problems-A probabilistic learning approach. Expert Syst Appl. 2016;44:413–31.
[18] Zhu W, Curry J, Marquez A. SIMD tabu search for the quadratic assignment problem with graphics hardware acceleration. Int J Production Res. 2010;48:1035–47.
[19] Zhu W, Curry J, Marquez A. GPU-accelerated SIMT tabu search for the quadratic assignment problem. Trans Am Manufacturing Res Institution SME. 2009;37:435–42.
[20] Light LW, Anderson PG. Typewriter keyboards via simulated annealing. In: *AI Expert* **8**. San Francisco: Miller Freeman Publishers; 1993.
[21] Card SK, Newell A, Moran TP. The psychology of human-computer interaction. Hillsdale - NJ: L. Erlbaum Associates Inc.; 1983.
[22] MacKenzie S, Soukoreff W. Text entry for mobile computing: models and methods, theory and practice. Human-Computer Interaction. 2002;17:147–198.
[23] Lin CJ, Liu C-N, Hwang J-L, Shiang WJ. Designing a handled trackball for seated computer tasks. J Chin Inst Ind Engineers. 2009;26:1–10.

[24] Soukoreff RW, *Text Entry for Mobile Systems: Models, Measures and Analyses for Text Entry Research*, Masters thesis, York University, Toronto, Canada (2002).
[25] Nugent CE, Vollmann TE, Ruml J. An experimental comparison of techniques for the assignment of facilities to locations. Oper Res. 1968;16:150–73.
[26] Hauschild M, Pelikan M. An introduction and survey of estimation of distribution algorithms. Swarm Evol Comput. 2011;1:111–28.
[27] Zhao F, Shao Z, Wang J, Zhang C. A hybrid differential evolution and estimation of distribution algorithm based on neighbourhood search for job shop scheduling problems. Int J Production Res. 2016;54:1039–60.
[28] Zhou S, Li X, Chen H, Guo C. Minimizing makespan in a no-wait flowshop with two batch processing machines using estimation of distribution algorithm. Int J Production Res. 2016;54:1–19.
[29] Wang K, Choi S, Qin H. An estimation of distribution algorithm for hybrid flow shop scheduling under stochastic processing times. Int J Production Res. 2014;52:7360–7376.
[30] Valdez SI, Chávez-Conde E, Hernandez EE, Ceccarelli M. Structure-control design of a mechatronic system with parallelogram mechanism using an estimation of distribution algorithm. Mechanics Based Design of Structures and Machines. 2016;44:58–71.
[31] Mühlenbein H, Paaß G, "From recombination of genes to the estimation of distributions i. Binary parameters," *Proceedings of the 4th International Conference on Parallel Problem Solving from Nature*,Sep. 22-26, London, UK, 178–187 (1996).
[32] Mühlenbein H, Mahnig T. Convergence theory and applications of the factorized distribution algorithm. J Comput Inf Technol. 1999;7:19–32.
[33] Baluja S,"Population-based incremental learning: a method for integrating genetic search based function optimization and competitive learning," *Technical Report*,School of Computer Science, Carnegie Mellon University, 1–41 (1994).
[34] Buffa ES, Armour GC, Vollmann TE. Allocating facilities with CRAFT. Harv Bus Rev. 1964;42:136–58.
[35] Burkard RE, Karisch SE, Rendl F. QAPLIB – A quadratic assignment problem library. Journal of Global Optimization. 1997;10:391–403.
[36] Wiktionary: Frequency lists, (Cited 2015 Apr 10Available from: https://en.wiktionary.org/wiki/Wiktionary:Frequency_lists
[37] Balakrishnan J, Cheng CH. Genetic search and the dynamic layout problem. Computers and Operations Research. 2000;27:587–593.
[38] *Project Gutenberg*,(Cited 2015 Apr 10 Available at: https://www.gutenberg.org/ebooks/