The single-finger keyboard layout problem

Mauro Dell'Amico   José Carlos Díaz Díaz
Manuel Iori   Roberto Montanari

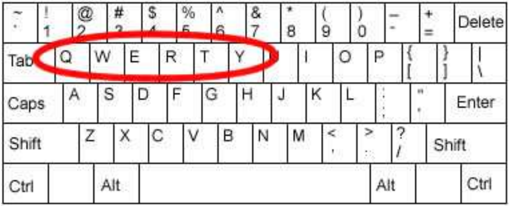
Department of Sciences and Methods for Engineering
University of Modena and Reggio Emilia

AIRO 2008
September 7-11, Ischia

Outline

The QWERTY keyboard— A major de-facto standard

- Keyboards are the most popular and effective devices to insert, edit, delete and update long chunk of information.
- Keyboards used with many fingers (*n-fingers*) were introduced more than 100 years ago.
- The first keyboard, was Q-W-E-R-T-Y: it is still the standard



The QWERTY keyboard— New challenges from new devices

- The proliferation of Portable Data Assistant (PDA), Smartphone and phones requires a strong improvement in the design of ad-hoc input devices.

- These devices are used in mobility and multitasking.
- The keyboards for portable systems are a still open design domain.

An Ergonomic point of view— *n*-fingers keyboards

Most research focused on the tentative to overtake QWERTY.

Typical measures are posture structure, discomfort produced, keying force, user acceptance etc.

Fixed layout, n simbols $\times$ n locations problem.

- M.A. Pollatschek, N. Gershoni and Y.T. Radday (1976)
  Angewandte Informatik
  – Simulation
- R.E. Burkard and J. Offermann (1977)
  Zeitschrift fr Operations Research, B
  – Simulated annealing (7-10% improvement over QWERTY)
- Eggers, Feillet, Kehl, Wagner and Yannou (2003)
  European Journal of Operational Research
  – Combined multi-objective function

  (i) distribution of the fingers load among all fingers;
  (ii) the hits number needed to compose a text;
  (iii) comfort and speed for hand changes;
  (iv) comfort and speed for finger changes;
  (v) avoidance of great steps among two different keys;
  (vi) hits direction (it should move from little finger towards thumb).

  – Global score: Weighted linear combination
  – Ant Colony Optimization
  – English and German languages.

An Ergonomic point of view— *s*-fingers keyboards

The objective is to improve the typing accurateness, as well as the time spent to edit a text.

The law introduced by Fitt (1954) model the time and difficulties required to move to a target area

$$f(\cdot) = \alpha + \beta \log_2 \left(\frac{D}{A} + 1 \right)$$

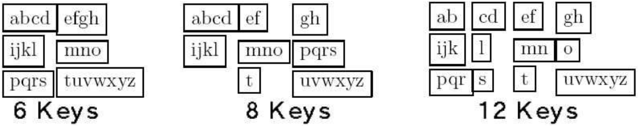
α, β	$=$	prefixed constant values
D	$=$	distance of the starting-ending keys
A	$=$	size of the target key

An Ergonomic point of view— *s*-fingers keyboards — incomplete

Incomplete keyboards: less keys than language symbols (e.g., the 12 keys ISO used in most mobile phones).

Combinatorial problem : find a 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 (2005)
Theoretical Computer Science
– NP-hard with two keys; several variants

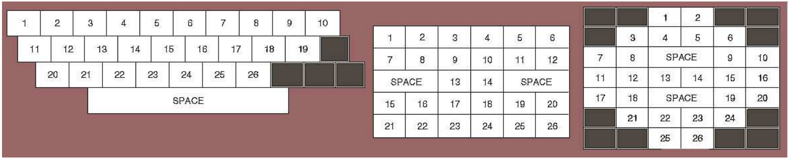


Optimal solutions by Cardinal and Lagerman

Complete keyboards: one keys for each symbol.

Many proposal: FITALY, ABC, OPTI, METROPOLIS, HOOKE, LEWIS, ... (see MacKenzie and Soukoreff, 2002)

- Li, Chen and Goonetilleke (2006)
  Industrial Ergonomics
  – Three fixed layout



  – Evaluate the movement through the Fitts' law.
  – A simple simulated annealing

The s-finger keyboard layout problem: (SK-QAP)

1. a list of different symbols
2. identical unit square keys (or locations) arranged in a grid
3. each symbol is assigned to exactly one key;
4. each key may contain exactly one symbol.

N.B. **More** keys than symbols $\Rightarrow$ the layout is NOT known

	A	B	C	
	D	E	F	
	G	H	I	

A	B	C	D	E
F	G	H	I	

	A	B		
C	D	E	F	
	G	H	I	

One instance $\longrightarrow$ many layouts

One layout $\longrightarrow$ many symbols' assignment !

The problem— Mathematical model

- $N = \{1, 2, \ldots, n\}$ set of symbols
- $M = \{1, 2, \ldots, m^2\}$ set of locations (keys)
- $\varphi : N \to M$ maps each symbol $i \in N$ to a location $\varphi(i)$ (φ is a solution)
- $\mathcal{S}$ set of all possible solutions φ

Matrices $\overline{A}$ and B to denote frequencies and "distances"

$\overline{A} = (\overline{a}_{ik})$: $\overline{a}_{ik}$ is the frequency of the symbol pair (i, k) (in an average stat.);
$B = (b_{jl})$: b_{jl} is the penalty to move from j to l (Fitts function, Eucl. dist.)

The problem can be stated as

$$z = \min_{\varphi \in \mathcal{S}} \sum_{i=1}^{n} \sum_{k=1}^{n} \overline{a}_{ik} b_{\varphi(i)\varphi(k)} \tag{1}$$

Since matrix B is symmetric we can rewrite (1) as

$$z = \min_{\varphi \in \mathcal{S}} \sum_{i=1}^{n} \sum_{k=i+1}^{n} (\overline{a}_{ik} + \overline{a}_{ki}) b_{\varphi(i)\varphi(k)} = \frac{1}{2} \min_{\varphi \in \mathcal{S}} \sum_{i=1}^{n} \sum_{k=1}^{n} a_{ik} b_{\varphi(i)\varphi(k)} \tag{2}$$

where $A = (a_{ik})$ is a symmetric matrix with $a_{ik} = \overline{a}_{ik} + \overline{a}_{ki}$.

SK-QAP is a generalization of the classic
Quadratic Assignment Problem (QAP)

– set $n = m \Rightarrow$ (i) one symbols for each location $\Rightarrow$ (ii) fixed layout

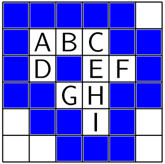
- Previous works are all devoted to fixed layout

SK-QAP is a generalization of the classic
Quadratic Assignment Problem (QAP)

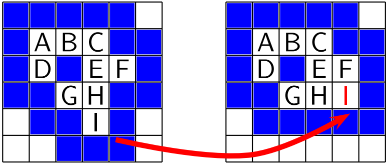
– set $n = m \Rightarrow$ (i) one symbols for each location $\Rightarrow$ (ii) fixed layout

- Previous works are all devoted to fixed layout
  - QAP is known to be strongly $\mathcal{NP}$-hard, hence SK-QAP
  - QAP is challenging: Benchmark instances nug27, nug28 and nug30 (proposed in the '60s) have been solved exactly only recently in 20 CPU days, 5 CPU months, and 2.5 CPU years, respectively (see Adams, Guignard, Hahn and Hightower '07)
  - QAP has been attacked with several heuristics and metaheuristics (see, e.g., Drezner, Hahn and Taillard (2005), James, Rego and Glover, 2007).

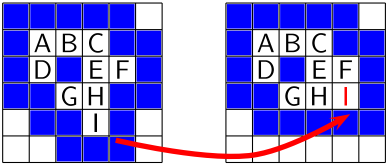
Neighborhood $\mathcal{N}_1$ (contour filling) : Set of solutions obtained by moving each symbol to an empty contour location.



Neighborhood $\mathcal{N}_1$ (contour filling) : Set of solutions obtained by moving each symbol to an empty contour location.



Neighborhood $\mathcal{N}_1$ (contour filling) : Set of solutions obtained by moving each symbol to an empty contour location.



- The neighborhood size is $O(n^2)$.
- The exploration from scratch requires $O(n^3)$ time.
- Adapting the improvement proposed by Frieze, Yadegar, El-Horbaty (1989) for QAP it reduces to $O(n^2)$.

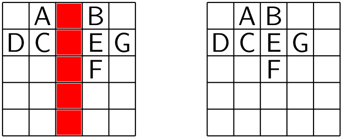
Neighborhood $\mathcal{N}_2$ (pairwise-exchange): Set of solutions obtained by swapping the assignment of two symbols r and s.

- The neighborhood size is $O(n^2)$.
- The exploration from scratch requires $O(n^3)$ time (no improvement possible)

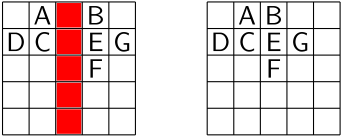
Neighborhood $\mathcal{N}_k$ (k-exchange): Set of solutions obtained by permuting in all possible ways the assignments of k symbols.

- Generalization of the pairwise-exchange (with $k = 2$).
- The size of the neighborhood is $O(n^k)$.
- The objective function is re-computed from scratch for each solution ($O(n^2)$)

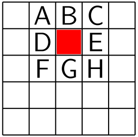
Property A solution of SK-QAP is not optimal if there is an empty line with symbols assigned on both sides.



Property A solution of SK-QAP is not optimal if there is an empty line with symbols assigned on both sides.



N.B. Special frequency matrices has optimal solutions with holes



QAP vs SK-QAP

QAP	different solutions $\Leftrightarrow$ different permutations φ.
SK-QAP	different assignments $\varphi \Leftrightarrow$ may layouts

			D	
		A	C	
		B	E	F
			G	

(a) original

	D			
A	C			
B	E	F		
	G			

(b) translation

		B	A	
	G	E	C	D
		F		

(c) 90° right rotation

			D	
			C	A
		F	E	B
			G	

(d) vertical reflectional symmetry

	A	B		
D	C	E	G	
		F		

(e) canonical form

Property We can restrict the search to canonical solutions.

We use a *long term memory* based on an hashing function

	A	B		
D	C	E	O	F
G	H	I	P	L
	Q	R	S	
	M	N		

We use a *long term memory* based on an hashing function

	A	B		
D	C	E	O	F
G	H	I	P	L
	Q	R	S	
	M	N		

	A	B		
D	C	E	O	F
G	H	I	P	L
	Q	R	S	
	M	N		

We select a witness for each border

We use a *long term memory* based on an hashing function

	A	B		
D	C	E	O	F
G	H	I	P	L
	Q	R	S	
	M	N		

	A	B		
D	C	E	O	F
G	H	I	P	L
	Q	R	S	
	M	N		

We select a witness
for each border

- Encode the symbols close to the four witness into a single integer (code)

	A	B		
D	C	E	O	F
G	H	I	P	L
	Q	R	S	
	M	N		

	A	B		
D	C	E	O	F
G	H	I	P	L
	Q	R	S	
	M	N		

	A	B		
D	C	E	O	F
G	H	I	P	L
	Q	R	S	
	M	N		

	A	B		
D	C	E	O	F
G	H	I	P	L
	Q	R	S	
	M	N		

- Two solutions are supposed identical if the two codes are equal.

Simulated Annealing (SA)

Burkard and Rendl (1984)
Wilhelm and Ward (1987)
Connolly 1990

Tabu Search (TS)

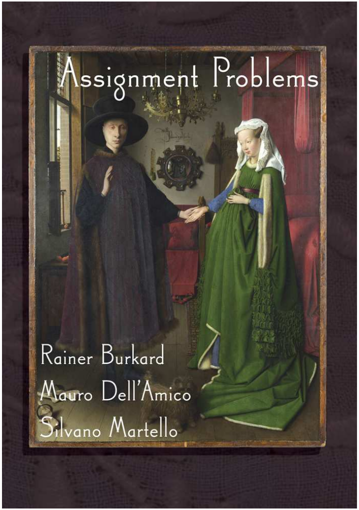
Skorin-Kapov (1990)
Taillard (1991) *Robust Tabu Search*
Battiti and Tecchiolli (1994) *Reactive Tabu Search*
Connolly 1990

Variable Neighborhood Search (VNS)

Taillard and Gambardella (1999)

Ant Colony Optimization (ACO)

Maniezzo and Colorni (1999)
Gambardella, Taillard and Dorigo (1999)
Taillard (1998) *Fast ANT* (FANT)



R. Burkard, M.Dell'Amico, S.Martello
SIAM
in print (scheduled November 2008)

(Bipartite Matching, Linear Sum, Algebraic, Bottleneck, Quadratic, Multi-index, Software Codes, etc.)

Local Search

LS_Refine

1. Find a local optimum of value $\mathbf{z_1}$ with neighborhood $\mathcal{N}_\mathbf{1}$
2. Try to improve the solution with $\mathcal{N}_\mathbf{2}$, giving $\mathbf{z_2}$
3. **if** $\mathbf{z_2} > \mathbf{z_1}$ **then goto** 1.

Two strategies

best-improvement: selects the best solution of the neighborhood

first-improvement selects the first improving solution

Metaheuristics

SA, TS, VNS and FANT implementd with

a. $\mathcal{N}_\mathbf{1}, \quad \mathcal{N}_\mathbf{2}, \quad \mathcal{N}_\mathbf{1} \cup \mathcal{N}_\mathbf{2}$

b. speed-ups for SK-QAP

c. the two strategies

English, French, Italian and Spanish languages.

List of the most frequent worlds

English and Spanish	`http://www.wiktionary.org`
French	Monde Diplomatique (1987-1997) (`http://www.up.univ-mrs.fr/~veronis`)
Italian	Scuola Normale Superiore of Pisa (`http://alphalinguistica.sns.it`)

- The first 10,000 words have been used
- The frequency of the transition between each pair of consecutive symbols was computed
- Punctuation was omitted, but for '**space**' and symbols $\boxed{\text{'}}$ and $\boxed{\text{-}}$ for English

Language symbols by decreasing frequencies

English
ø E T O A N I H S R D L U F M W C Y G P B V K X J ' Q Z -
French
ø E S A I N T R U L O D C P M é ' V Q G F B H à X è J Y ê ô Z K ç ù î û W â ï ë ä
Italian
ø E A I O N R L T S C U D P M V G H F B ' Z Q è à ù ò é ì X K Y J W
Spanish
ø E A O S N R U L T D I M C P Q Y H B V G é á í ó J F Z ú ñ X K W ü

RANDOM INSTANCES

Five random benchmarks have been generated from the statistical distribution of the frequencies of each real language.

Algorithms coded in Delphi language and run on a PC Pentium 4 at 3.0 GHz under Windows XP with a Time Limit of 120 CPU seconds

Computational results— Real languages

Algorithms coded in Delphi language and run on a PC Pentium 4 at 3.0 GHz under Windows XP with a Time Limit of 120 CPU seconds

Best solutions

	J	F	W	B	X	
Q	U	O	T	H	C	
	M	S	ø	E	R	K
	P	I	N	A	Y	-
	Z	G	D	L	W	
			’			

English
(1.199.070.166)

			ù				
		Z	J	Q	X		
	H	C	O	U	M	ê	
	V	I	N	E	D	B	ä
û	è	T	S	ø	L	’	ë
	ô	é	R	A	P	à	
	â	K	G	F	Y	W	
			î	ç	ï		

French
(13.491.323.058)

			ù			
	è	F	G	’	Z	
é	H	C	I	L	B	
ì	S	O	ø	A	M	Y
à	T	N	E	R	V	W
	Q	U	D	P	K	
		J	ò	X		

Italian
(4.025.754.990)

		Q	é	V	á	ú	
	F	U	N	S	T	J	
ü	G	D	E	ø	O	P	K
	ó	I	R	A	L	Y	W
	X	B	M	C	H	ñ	
			Z	í			

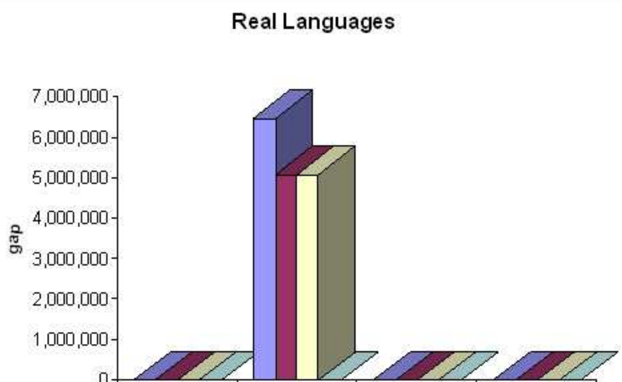
Spanish
(1.229.359.070)

Algorithms coded in Delphi language and run on a PC Pentium 4 at 3.0 GHz under Windows XP with a Time Limit of 120 CPU seconds

DELL' AMICO

			ù			
	è	F	G	'	Z	
é	H	C	I	L	B	
ì	S	O	ø	A	M	Y
à	T	N	E	R	V	W
	Q	U	D	P	K	
		J	ò	X		

Italian
(4.025.754.990)



	English	French	Italian	Spanish
SA	0	6,461,578	0	0
TS	0	5,080,177	0	0
VNS	0	5,080,177	0	0
FANT	0	0	0	0

Computational results— Random instances

N.	$\mathcal{L}$	*SA2*		*SA12*		*TS2*	*TS12*	*VNS*
		$\alpha = 0.95$	$\alpha = 0.98$	$\alpha = 0.95$	$\alpha = 0.98$			
1		0	0	0	0	0	0	0
2		0	10.730	0	115.629	0	0	0
3	English	0	0	0	1.290.638	0	0	0
4		0	0	0	0	0	0	0
5		0	0	0	0	0	0	0
1		0	142.729	1.297.299	131359	0	0	0
2		243.404	243.404	243.404	521.211	243.404	243.404	243.404
3	French	0	368.964	0	991.021	0	0	0
4		0	0	1.341.765	312.749	0	0	0
5		0	33.895	0	748.529	0	0	659.504
1		0	92.783	0	92.144	0	0	0
2		0	1.183.330	0	221.130	0	0	0
3	Italian	0	1.162.935	0	887.635	0	0	0
4		0	0	0	0	0	0	0
5		0	0	0	14.806	0	0	0
1		0	0	0	0	0	0	0
2		0	0	0	0	0	0	0
3	Spanish	0	0	0	0	0	0	0
4		0	52.141	0	0	0	0	0
5		0	0	0	0	0	0	0
	#Best	22	14	20	11	22	22	21

Conclusions—

- We discussed the single finger keyboard layout problem: an open domain
- We modeled the problem as a generalized Quadratic Assignment Problem
- We designed specialized metaheuristic algorithms
- We proposed benchmark instances from real languages

Data and results: `www.or.unimore.it`