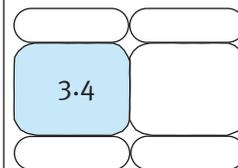


LA STRUTTURA DELLE RETI IN UN MONDO INTERCONNESSO



Andrea Bobbio

Tutti sono d'accordo nell'ammettere che ciò che caratterizza il mondo moderno è l'interdipendenza. Le strutture sociali, tecnologiche, economiche tendono sempre più ad aggregarsi e a connettersi. È possibile individuare leggi generali che regolano queste interconnessioni e ne determinano il comportamento? Questa domanda ha aperto una nuova scienza la "teoria delle reti complesse" che ha avuto uno sviluppo prepotente e impensabile negli ultimi anni.



1. INTRODUZIONE

Una caratteristica peculiare e generale dell'odierna organizzazione del vivere umano è data dalla interconnessione. Le strutture sociali, tecnologiche, economiche tendono sempre più ad aggregarsi, a connettersi. Componendo un numero sul nostro apparecchio telefonico possiamo parlare con qualunque altro utente del servizio telefonico in qualunque parte del mondo. L'energia elettrica viene capillarmente distribuita attraverso una rete, così come il gas e l'acqua. I trasporti sono basati su reti: reti stradali, ferroviarie, aeree. Esistono reti sociali di legami personali e di gruppo, reti di relazioni economiche. I sistemi biologici sono costituiti da cellule interagenti attraverso complesse reti di reazioni biochimiche. Ma forse l'evento che ha avuto di più influenza sulla nostra vita è stata l'apparizione di Internet e del *World Wide Web* (www). Da pochi anni i computer di tutto il mondo possono collegarsi fra di loro attraverso la rete fisica di Internet, mentre il www è una rete virtuale i cui nodi sono pagine o documenti che possono contenere

qualunque informazione: testi, immagini, filmati e suoni. La rete Internet sta diventando sempre più veloce e potente e il www sta crescendo a ritmi vertiginosi: centinaia o migliaia di nuove pagine vengono prodotte ogni giorno e sta diventando forse la creazione dell'uomo più complessa e multiforme. Come studiare questi sistemi complessi? È possibile trovare leggi che ne regolano la struttura, le forme di aggregazione, le proprietà? Nei secoli scorsi lo scopo primo dell'indagine scientifica era stato quello di decifrare i componenti base della natura perché si pensava che la comprensione del mondo passasse attraverso la conoscenza dei suoi costituenti primi: le particelle elementari, gli atomi, le molecole, le proteine, le cellule. Questo approccio ha portato a risultati di enorme valore, ma si è forse perso di vista l'insieme: come i mattoni elementari si combinano in complesse strutture coese. Si è perso di vista il fatto che il comportamento e le proprietà di sistemi costituiti da aggregati di parti elementari dipendono dai legami e dalle relazioni fra le parti. I neu-

rosenziati ci spiegano che la dicotomia fra cervello e mente, cioè fra l'ammasso di cellule che costituisce il cervello e il pensiero dipende dai legami fra le cellule. Scrive Rita Levi Montalcini [1] "Nessuna cellula nervosa, includendo tra queste anche quelle della corteccia cerebrale, è in grado di percepire, né essere capace di reazioni emotive, di pensare, ma tali proprietà sarebbero una funzione emergente dell'intero sistema corticale e dei sistemi neuronali del sistema stesso. Per questo motivo soltanto l'analisi di complessi cellulari e dei circuiti che li interconnettono, può portare un contributo allo studio dei processi mentali".

La consapevolezza dell'importanza dei legami fra le parti, l'impatto determinato dalla presenza, ubiquità e irrinunciabilità delle reti di calcolatori e la spinta derivante dal crescente interesse per i sistemi biologici e per le neuroscienze hanno portato all'insorgere di una nuova disciplina che viene normalmente identificata come "teoria delle reti complesse".

Gli studi per la comprensione delle strutture a rete complessa prendono le mosse da due lavori quasi contemporanei di Watts e Strogatz [2] sulle reti a "piccolo mondo" e di Barabasi e Albert [3] sulle reti a legge di potenza e vedono uno sviluppo incredibilmente fertile e rapido che coinvolge in uno sforzo congiunto fisici, matematici, informatici, biologi. Il presente lavoro intende fornire una prima lettura sugli indirizzi che hanno assunto questi recenti studi e sui principali risultati raggiunti. Per un avvicinamento più profondo alle pro-

blematiche legate alla teoria delle reti, si consiglia la lettura del libro di uno degli iniziatori di questo filone di ricerca A.L Barabasi [4]. Il libro è disponibile in una edizione italiana, è di facile comprensione per un lettore di normale cultura scientifica ed è letterariamente molto gradevole e attraente. Per chi fosse interessato ad approfondimenti più tecnici e dettagliati, riguardanti anche le metodologie di analisi e le tecniche matematiche di indagine, si rinvia a estesi lavori scientifici di rassegna sull'argomento [5, 6, 7, 8] e alle loro voluminose bibliografie. L'ultimo e più recente lavoro citato [8] rimanda a quasi mille riferimenti bibliografici!

2. LE GRANDEZZE CHE CARATTERIZZANO UNA RETE

Una rete è rappresentabile mediante un grafo $G = (V, E)$ dove V è l'insieme dei nodi (o vertici) e E è l'insieme degli archi che connettono i nodi (Figura 1 A). Gli archi possono essere non orientati e quindi possono essere percorsi nelle due direzioni (come ad esempio un collegamento fra due calcolatori sulla rete internet, o un collegamento telefonico fra due apparecchi, o i legami di amicizia fra due soggetti), oppure possono essere orientati.

Gli archi orientati possono essere percorsi in una sola direzione indicata normalmente da una freccia (Figura 1 B). Sono esempi di grafi orientati il reticolo fluviale di una regione (l'acqua scorre in una sola direzione), le reti di citazioni bibliografiche e il www. Se una pagina a del www contiene un puntatore

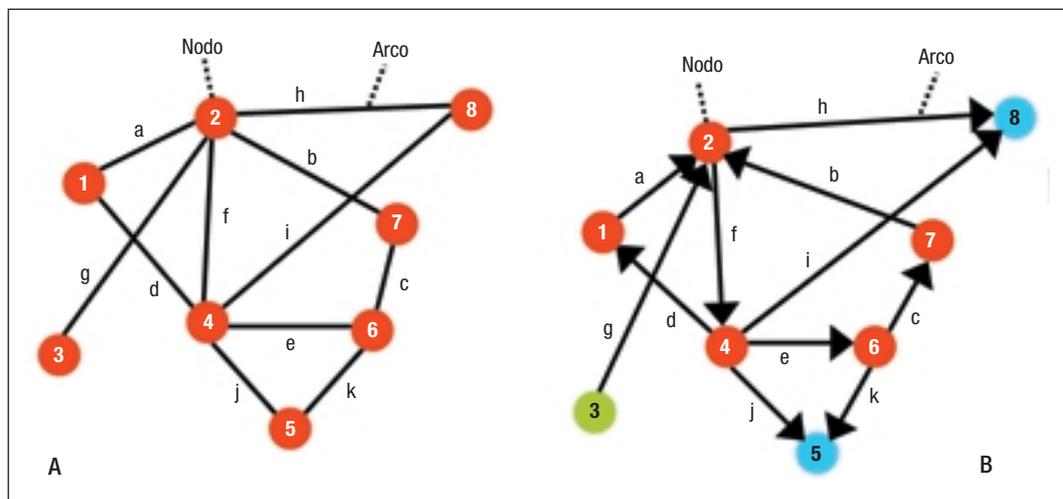


FIGURA 1
A Esempio di grafo non orientato;
B Esempio di grafo orientato

(link) ad una pagina b vuol dire che da a si può raggiungere b con un click del mouse ma non viceversa (a meno che nella pagina b non ci sia un link esplicito alla pagina a).

In una rete non orientata basta un solo arco per nodo per rendere il grafo completamente connesso, cioè per poter raggiungere qualsiasi nodo partendo da qualsiasi altro nodo. Al di sotto della soglia limite di un arco per nodo una rete si decompone in aggregati di nodi non raggiungibili reciprocamente. Le reti in natura superano normalmente il valore di soglia limite, in quanto hanno normalmente più di un arco emergente da ogni nodo: questa molteplicità di archi rende le reti molto più strettamente connesse e ridondanti. Per le reti orientate la connessione è più complessa, in quanto si possono identificare tre tipologie distinte di nodi. Nodi che formano insiemi connessi cioè in cui tutti i nodi sono raggiungibili fra loro seguendo la direzione degli archi (per esempio i nodi rossi nella Figura 1 B). I nodi sorgente che hanno solo archi in uscita e non sono raggiungibili da nessun altro nodo (per esempio il nodo verde in Figura 1 B). I nodi assorbenti che hanno archi in ingresso ma non in uscita: possono essere raggiunti ma una volta raggiunti non possono essere abbandonati (per esempio i nodi blu nella Figura 1 B).

Un grafo, quale quello di figura 1 A o figura 1 B, può avere anche una rappresentazione matematica mediante la cosiddetta matrice di incidenza. Se il grafo ha N nodi numerati da 1 a N , la matrice di incidenza è una matrice quadrata $N \times N$ il cui elemento ij è uguale a 1 se i e j sono direttamente connessi da un arco, altrimenti è uguale a 0. La i -esima riga della matrice di incidenza individua gli archi in uscita dal nodo i , mentre la i -esima colonna individua gli archi in ingresso al nodo i . Se il grafo è non orientato la matrice di incidenza è simmetrica (cioè l'elemento con indice di riga i e indice di colonna j è uguale all'elemento con indice di riga j e indice di colonna i). Se il grafo è orientato la matrice non è simmetrica: i nodi sorgente hanno la corrispondente colonna della matrice di incidenza tutta di zeri; i nodi assorbenti hanno la corrispondente riga della matrice di incidenza tutta di zeri.

Le reti molto complesse con migliaia, milio-

ni o miliardi di nodi, quali sono quelle con cui abbiamo a che fare quotidianamente, sono difficili da descrivere e impossibili da disegnare. Per definire e caratterizzare le proprietà di queste reti complesse sono state introdotte varie misure. Le misure maggiormente adottate [9], e che verranno riprese in questo lavoro (vedi riquadro a p. 6 per un esempio numerico) sono il *grado di connettività*, la *lunghezza del cammino caratteristico* e il *coefficiente di aggregazione (clustering)*.

2.1. Grado di connettività

Il *grado di connettività* di un nodo (o semplicemente il *grado* di un nodo) è il parametro più studiato perché facilmente discrimina le caratteristiche di una rete complessa. In una rete non orientata il grado di un nodo è dato dal numero di archi emergenti dal nodo (equivalente al numero di nodi direttamente raggiungibili dal nodo in esame, detti anche nodi primi vicini). In una rete orientata si distingue fra il grado in ingresso (in-degree) che conta il numero di archi in ingresso in un nodo e il grado in uscita (out-degree) che conta il numero di archi emergenti da un nodo. Limitandoci ora a reti non orientate si indichi con k_i il grado del nodo i , e con $P(k)$ la distribuzione del grado di connettività cioè la probabilità che un nodo abbia grado k (o più semplicemente la percentuale di nodi di grado k sul numero totale di nodi N). Il valore medio del grado $\langle k \rangle$ è ottenuto mediando il grado su tutti i nodi della rete. Sebbene il grado sia una proprietà locale di un nodo, la distribuzione del grado $P(k)$ è una delle principali caratteristiche che determina, come vedremo, comportamenti complessivi della rete.

2.2. Lunghezza del cammino caratteristico L

La *Lunghezza del cammino caratteristico* (o semplicemente il *cammino caratteristico*) L misura la tipica separazione fra due generici nodi del grafo ed è anche, a volte, indicata come il diametro del grafo. Dati due nodi i e j definiamo d_{ij} la loro distanza minima, cioè il numero minimo di archi che si devono attraversare per raggiungere il nodo j partendo dal nodo i (supponiamo in questo caso che il

Un esempio di misure di rete

Prendendo a riferimento l'esempio di rete non orientata riportata nel grafo di figura 1 A, vengono esplicitamente calcolate le misure caratterizzanti le reti complesse definite nel paragrafo 2 del presente lavoro.

Grado di connettività

Il grado di connettività di un nodo è dato dal numero di archi emergenti dal nodo. Nella tabella A viene riportato, con riferimento al grafo di figura 1 A, il numero di archi emergenti da ogni nodo e il suo grado: poiché ogni arco connette due nodi, il grado è dato dal No. Archi diviso due. Il calcolo della media dei valori della terza colonna della tabella A permette di ottenere il valore medio del grado $\langle k \rangle = 1.375$.

I valori della terza colonna della tabella A forniscono anche la distribuzione del grado di connettività $P(k)$.

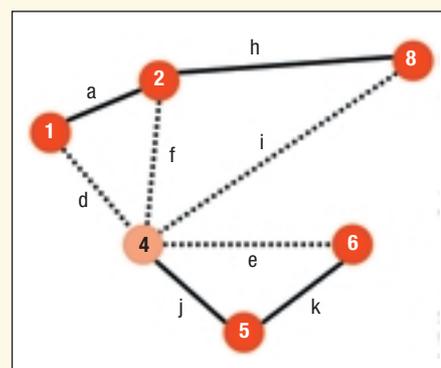
Lunghezza del cammino caratteristico L

Dato che la lunghezza del cammino caratteristico L è definita come la media delle distanze minime fra ogni coppia di nodi, occorre valutare dal grafo la distanza minima, espressa come numero minimo di archi che bisogna attraversare per connettere i due nodi.

Per esempio, prendendo come sorgente il nodo 3 del grafo di figura 1 A, le distanze minime nei confronti di tutti gli altri nodi sono riportate nella tabella B.

TABELLA A		
Nodo	No. Archi	Grado
1	2	1
2	5	2.5
3	1	0.5
4	5	2.5
5	2	1
6	3	1.5
7	2	1
8	2	1

TABELLA B		
Nodo sorgente	Nodo destinazione	Distanza minima
3	1	2
3	2	1
3	4	2
3	5	3
3	6	3
3	7	2
3	8	2



Il sottografo G_4

Calcolando le distanze minime per tutti i nodi e facendo la media si ottiene la *Lunghezza del cammino caratteristico* $L = 1.68$.

Coefficiente di aggregazione (clustering coefficient) C
Nel grafo di figura 1 A, si consideri il nodo 4 e si disegni il sottografo G_4 ottenuto dal grafo completo G prendendo i primi vicini del nodo 4 e eliminando il nodo e gli archi da esso emergenti (vedi Figura).

Poiché i primi vicini del nodo 4 sono i nodi 1, 2, 5, 6 e 8, il sottografo G_4 ha 5 nodi e al massimo (se tutti i nodi fossero connessi) $5 \cdot 4/2 = 10$ archi. Di questi 10 archi potenziali il sottografo G_4 ne contiene 4 da cui il coefficiente di aggregazione relativo al nodo 4 diventa $C_4 = 4/10 = 0.4$.

La tabella C riporta i coefficienti di aggregazione relativi ad ogni nodo da cui si calcola il coefficiente di aggregazione medio $C = 0.492$.

TABELLA C		
Nodo	Primi vicini	Coefficiente aggregazione
1	2	1
2	5	0.2
3	1	0
4	5	0.4
5	2	1
6	3	0.333
7	2	0
8	2	1

grafo sia non orientato e connesso cioè che partendo da ogni nodo sia possibile raggiungere qualunque altro nodo). Quindi, se i e j sono direttamente connessi da un arco avre-

mo $d_{ij} = 1$, altrimenti $d_{ij} > 1$. Si definisce la lunghezza del cammino caratteristico L come la media delle distanze minime fra ogni coppia di nodi. Una rete in cui tutti i nodi sono con-



nessi a tutti gli altri nodi ha una matrice di incidenza con tutti gli elementi ij fuori dalla diagonale principale ($i \neq j$) uguali a uno: in questo caso il cammino caratteristico è $L = 1$, il numero totale di archi è $N(N-1)/2$ e il grado è costante pari a $k = (N-1)/2$. Se $\langle k \rangle$ è il valor medio del grado, da ogni nodo si possono mediamente raggiungere in un solo passo $\langle k \rangle$ nodi, in due passi $\langle k \rangle^2$ nodi ecc... e in L passi si ricopriranno tutti gli N nodi. Per cui avremo $N \sim \langle k \rangle^L$ da cui $L \sim \ln N / \ln \langle k \rangle$. Il valore di L è normalmente un numero piccolo che cresce lentamente con il numero totale di nodi N . Questa proprietà, per cui la distanza media fra nodi è un numero sovente molto basso, viene normalmente riferita come proprietà di "piccolo mondo".

2.3. Coefficiente di aggregazione (clustering coefficient) C

Dato un nodo i , si definisce il sottografo G_i ottenuto dal grafo G prendendo i primi vicini di i e eliminando il nodo i e gli archi da esso emergenti. Se il nodo i ha grado k_i , allora G_i ha k_i nodi e, al massimo, $k_i(k_i-1)/2$ archi. Il coefficiente di aggregazione C_i relativo al nodo i , misura il numero di archi realmente presenti nel sottografo G_i rispetto al numero di archi teoricamente possibili; misura, cioè, quanto i primi vicini del nodo i sono connessi fra loro rimuovendo i . Se un nodo a è connesso ai nodi b e c , il coefficiente di aggregazione va a considerare se b e c sono anche connessi fra di loro (in termini di reti sociali si usa esprimere questa proprietà dicendo che il coefficiente di aggregazione misura se i miei amici sono anche amici fra di loro). Il coefficiente di aggregazione C è il valor medio di C_i calcolato su tutti i nodi del grafo.

3. LE RETI CASUALI

Le prime reti complesse ad essere state sistematicamente studiate furono le reti casuali introdotte da due matematici ungheresi Erdős e Rényi [4] attorno agli anni '60. In una rete casuale di N nodi, ogni coppia di nodi ha la stessa probabilità p di essere connessa. Quindi quando p è prossimo a zero i nodi sono prevalentemente isolati, quando p tende a 1 i nodi tendono a essere connessi a tutti gli altri nodi. Si può dimostrare che quando N è

grande, il grado medio di connettività dei nodi è $\langle k \rangle = Np$ e la distribuzione del grado $P(k)$ è approssimata da una legge di Poisson.

$$P(k) = (\langle k \rangle^k / k!) e^{-\langle k \rangle} \quad (1)$$

La legge di Poisson ha una forma a campana con un massimo centrato sul valor medio $\langle k \rangle$. Un esempio di rete con distribuzione del grado di tipo Poissoniano (1) è la rete autostradale negli USA, rappresentata nella parte sinistra di figura 4. Gli archi sono le strade e i nodi i loro incroci (prevalentemente nelle città). In una rete Poissoniana, i nodi hanno come grado più probabile il valore $\langle k \rangle$, e la probabilità del valore del grado diminuisce rapidamente (esponenzialmente) al discostarsi di k dal valore medio $\langle k \rangle$ nelle due direzioni. In una rete casuale il cammino caratteristico L tende a essere piccolo e a diminuire con l'aumentare di p ; anche il coefficiente di aggregazione tende ad essere piccolo perchè il fatto che un nodo a sia connesso a due nodi b e c non aumenta la probabilità che b sia connesso a c . In una rete casuale il coefficiente di aggregazione può essere espresso come $C = \langle k \rangle / N$. Tuttavia, le reti reali sono tutt'altro che casuali! In una rete casuale di Erdős e Rényi la probabilità che il mio computer sia connesso al router del mio ufficio sarebbe uguale a quella di essere connesso a qualunque altro computer del mondo oppure la probabilità di un individuo di conoscere, e quindi stabilire un legame con il vicino di casa sarebbe la stessa della probabilità di conoscere uno sciamano della Papuasias. Nessuno pensa che ciò sia vero. Poiché lo scopo fondamentale dell'indagine scientifica è quello di identificare modelli per descrivere la realtà, è necessario introdurre nuove ipotesi e nuovi ragionamenti per l'interpretazione e lo studio delle reti reali.

4. LE RETI A PICCOLO MONDO

Nel 1967 il sociologo americano Stanley Milgram mise in piedi un esperimento per studiare la "distanza" fra due cittadini qualunque degli Stati Uniti [10]. La domanda a cui si voleva dare risposta era: scelte due persone a caso, quanti contatti sono necessari

per connetterle? L'esperimento consisteva nel selezionare casualmente un gruppo di individui e nel consegnare a ciascuno di loro una lettera con indicato un destinatario finale. Ogni soggetto, se non conosceva direttamente il destinatario finale, doveva inoltrare il documento ad una sola persona, a lui personalmente nota, e che riteneva avesse maggiore opportunità di raggiungere il destinatario finale. Si ipotizzava che questa catena di contatti richiedesse almeno un centinaio di passaggi. Alla fine dell'esperimento Milgram registrò che su centosessanta individui iniziatori, quarantaquattro catene erano arrivate alla conclusione e che il numero medio di passaggi era sorprendentemente basso: 5.5 passaggi. È ciò che i sociologi chiamano i sei gradi di separazione. Qualunque persona nel mondo può essere raggiunta mediamente attraverso sei passaggi. Il mondo è un "piccolo mondo". Si noti che nell'esperimento di Milgram i cammini ottenuti non erano necessariamente minimi: un individuo in una catena poteva non sapere che il suo vicino di casa era il cugino del destinatario finale. Tuttavia l'esperimento mostrava un altro aspetto estremamente interessante: gli individui riescono a trovare percorsi brevi (anche se non minimi) orientandosi puramente sulla base delle conoscenze locali della rete e senza nessuna visione globale.

Le reti reali hanno normalmente un coefficiente di aggregazione più elevato di quanto previsto dalla teoria delle reti casuali, in quanto è ragionevole ipotizzare che due nodi che hanno un qualche legame con un terzo nodo possano avere anche un legame fra di loro. È facile, inoltre, immaginare che il cammino caratteristico diminuisca e il grado di aggregazione cresca quando la rete si rivolge a gruppi di persone con interessi omogenei.

Nato da un quiz televisivo americano, è stato sviluppato un curioso sito internet che misura il "numero di Kevin Bacon", cioè la distanza (minima) degli attori cinematografici di tutto il mondo da Kevin Bacon. Si riteneva (supposizione poi dimostratasi non del tutto fondata [4]) che Kevin Bacon fosse uno degli attori di Hollywood che avesse recitato in più film con il maggior numero di colleghi attori, e si intende-

va misurare quanto gli altri attori fossero "distanti" da Kevin Bacon. Indicando il numero di Kevin Bacon con il simbolo NKB , un attore che abbia recitato in un film con Kevin Bacon ha $NKB = 1$; un attore che abbia recitato in un film in cui era presente un attore che avesse recitato in un film con Kevin Bacon ha $NKB = 2$ e così via. Il sito internet che permette di misurare NKB per tutti gli attori cinematografici e che fornisce anche la catena minima di passaggi con l'indicazione degli attori e dei film intermedi è stato sviluppato, e viene costantemente aggiornato, da due studenti dell'Università della Virginia a partire dal 1996 e ha indirizzo <http://www.cs.virginia.edu/oracle/>.

Per due attori italiani sulla cresta dell'onda al momento, Giovanna Mezzogiorno e Roberto Benigni, il sito di Kevin Bacon ha prodotto i seguenti risultati:

Giovanna Mezzogiorno has a Bacon number of 3.

1. Giovanna Mezzogiorno was in *Ilaria Alpi - Il più crudele dei giorni* (2002) with Tony Lo Bianco
2. Tony Lo Bianco was in *City Heat* (1984) with Beau Starr
3. Beau Starr was in *Where the Truth Lies* (2005) with Kevin Bacon

Roberto Benigni has a Bacon number of 2.

1. Roberto Benigni was in *Pinocchio* (2002) with Queen Latifah
2. Queen Latifah was in *Beauty Shop* (2005) with Kevin Bacon

Attivando il sito con qualunque attore si osservano valori di NKB sempre di poche unità. La comunità degli attori è veramente un piccolo mondo, per cui è stata stimata una distanza media di tutti gli attori da Kevin Bacon di $L = 2,8$ e di tutti gli attori fra loro di $L = 3,6$. Le reti reali hanno la proprietà di piccolo mondo riscontrate anche nelle reti casuali di Erdős e Rényi, ma rispetto alle reti casuali hanno coefficiente di aggregazione molto più elevato. Il primo tentativo di spiegare la concomitante presenza di piccolo mondo e grande aggregazione fu proposto da Watts e Strogatz in un lavoro [2] che ebbe un grandissimo impatto sulla comunità scientifica. Per illustrare l'insorgere di una rete a piccolo mondo Watts e Strogatz hanno proposto un modello (che chiamo-

remo modello WS) in cui la variazione di un singolo parametro (p) permette di passare con continuità da una struttura a reticolo regolare ad una struttura a grafo casuale. Nella figura 2 è rappresentato un grafo su un reticolo regolare di N nodi disposti su una circonferenza e ogni nodo ha lo stesso grado k . Nella figura 2 si è rappresentato il caso $N = 16$ e $k = 2$ (ogni nodo è connesso con i suoi due primi vicini sia a destra che a sinistra, ma gli archi sono bidirezionali e vanno contati una volta sola).

Con questa struttura il coefficiente di aggregazione C risulta indipendente dalla dimensione del grafo mentre la lunghezza del cammino caratteristico L risulta proporzionale al numero di nodi N e al grado k , e cresce al crescere di N , perdendo la caratteristica di piccolo mondo. La struttura regolare abbina quindi elevato cammino caratteristico L e elevato grado di aggregazione C .

Il modello WS prevede ora di assegnare un valore di probabilità $p > 0$, chiamata probabilità di riassegnamento; con probabilità p si decide se ritracciare ogni arco, sconnettendolo dalla vecchia locazione e ridisegnandolo in una nuova locazione scelta a caso. Nascono così, in modo casuale, archi di "corto circuito" che attraversano il grafo (Figura 2 B). Con questa procedura di riassegnamento degli archi, quando p cresce la rete tende a diventare sempre più disordinata e per p che tende a 1 il modello WS produce una rete casuale (Figura 2 C). Con questo procedimento, Watts e Strogatz hanno dimostrato che con piccoli valori di p , cioè con un numero anche molto limitato di archi casuali, la rete cambia regime e il suo cammino caratteristico L decade rapidamente pur mantenendo quasi inalterato il valore del coefficiente di aggregazione C . Bastano pochi archi casuali in un reticolo regolare, anche di grandissime dimensioni, per far precipitare il valore del cammino caratteristico L e far apparire l'effetto di piccolo mondo, pur mantenendo un elevato coefficiente di aggregazione caratteristico delle strutture regolari.

Il fatto che pochi archi casuali possano modificare in modo sostanziale la connettività delle reti e ridurre drasticamente il cammino caratteristico può anche essere visto intuitivamente in modo semplice. Siano A , B e C tre reti a piccolo mondo separate fra loro. Tutti i nodi all'interno di A possono essere raggiun-

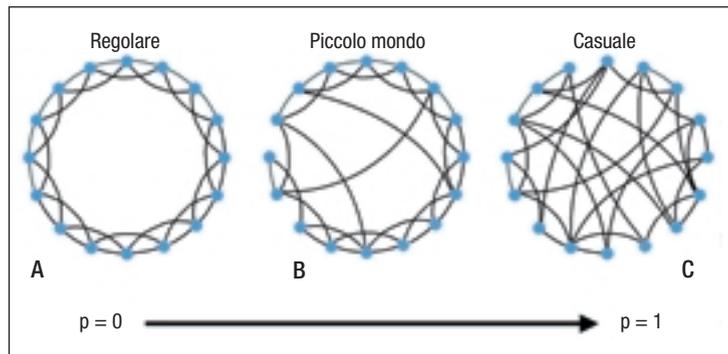


FIGURA 2

Il modello di Watts e Strogatz (WS) al crescere della probabilità di riassegnamento p . A) $p = 0$, reticolo regolare; B) p piccolo, nascita di archi casuali con l'effetto piccolo mondo; C) $p = 1$, grafo casuale

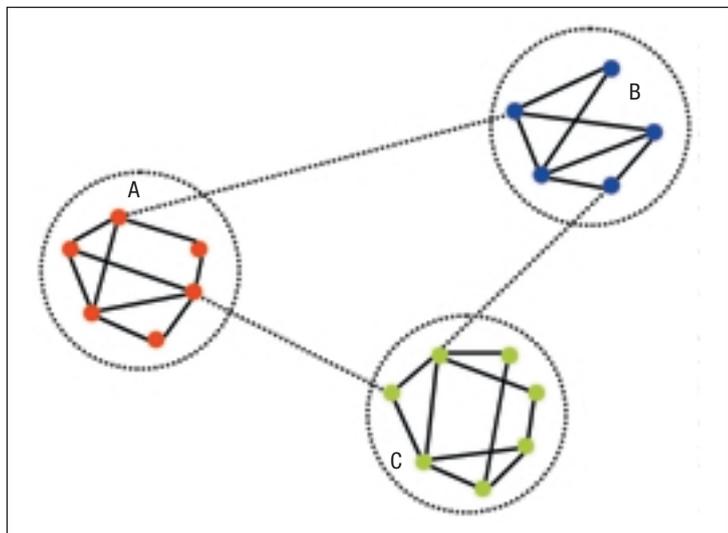


FIGURA 3

Tre reti a piccolo mondo A , B e C connesse da pochi link

ti in pochi passi, tutti i nodi all'interno di B possono essere raggiunti in pochi passi e tutti i nodi all'interno di C possono essere raggiunti in pochi passi (A , B e C possono essere reti locali in una rete di calcolatori, o comunità separate nel caso di reti sociali). È sufficiente che un solo nodo di A sia connesso con un solo nodo di B e B con un solo nodo di C per far diventare un piccolo mondo l'unione delle reti A , B e C e permettere ad ogni nodo di un colore di raggiungere in pochi passi un nodo di qualunque altro colore.

La rete Internet (vedi riquadro a p. 14) presenta una struttura parzialmente ispirata a quella di figura 3, in cui sotto-reti locali sono connesse fra loro mediante collegamenti ad alta capacità.

5. LE RETI IN EVOLUZIONE E LE LEGGI DI POTENZA

Il crescente interesse per le reti complesse ha indotto vari ricercatori ad indagare sulla struttura di varie reti reali in ambiti molto diversi: informatico, tecnologico, sociale, biologico, economico. Questi studi hanno mostrato che molte reti reali presentano una distribuzione del grado di connettività $P(k)$ ben lontana dalla forma a campana prevista dalle legge di Poisson generata dalle reti casuali. I lavori di rassegna sull'argomento [5, 6, 7, 8], già citati, riportano tabelle riepilogative di proprietà di reti da cui emerge che un significativo numero di reti reali (www, internet, citazioni scientifiche nel database ISI, reti metaboliche, reti di proteine) ha una distribuzione del grado $P(k)$ che segue una legge di potenza (almeno per alti valori di k) rappresentabile dalla seguente espressione:

$$P(k) \sim c k^{-\gamma} \quad (2)$$

Nella formula (2) c è una costante di proporzionalità il cui valore è solitamente di scarso interesse pratico (vedi riquadro a p. 11 per maggiori dettagli) e γ è il coefficiente della legge di potenza. Dagli studi citati emerge che il valore di γ è normalmente compreso fra 2 e 3. Le leggi di potenza erano già note da molti anni nella

statistica e nella letteratura tecnica dove erano, a volte, chiamate leggi di Pareto o di Zipf. Già l'economista e sociologo Vilfredo Pareto (1848-1923) aveva scoperto, infatti, che la distribuzione del reddito degli individui segue una legge di potenza. Come illustrato nel quadro esplicativo B, una legge di potenza disegnata su un grafico con scala logaritmica su entrambi gli assi, viene rappresentata da una retta la cui pendenza è legata all'esponente γ .

In estrema sintesi le leggi di potenza sono caratterizzate da una "lunga coda", cioè la funzione $P(k)$ tende a zero lentamente (secondo la potenza $-\gamma$) al crescere di k . Il lento decadimento di $P(k)$ in funzione di k fa sì che ci sia una probabilità non trascurabile di trovare valori di k grandi e molto grandi. Quindi una rete con distribuzione del grado del tipo rappresentato nella formula (2) contiene un gran numero di nodi con grado piccolo insieme a pochi nodi con grado molto alto, cioè con un numero molto elevato di archi. Questi nodi, detti connettori (o *hub*), sono gli elementi che maggiormente influenzano la topologia e le proprietà della rete. Il percorso minimo fra due nodi qualunque passa normalmente attraverso i connettori. Una rappresentazione molto intuitiva, esplicita e significativa delle reti a legge di potenza è data dalle reti delle linee aeree. Nella figura 4 B è sche-

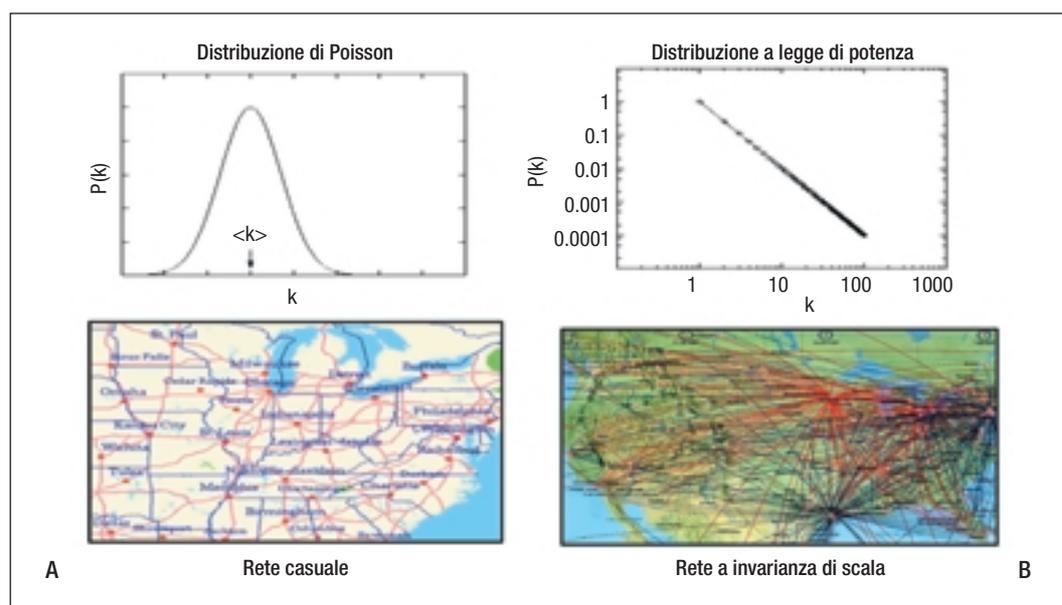


FIGURA 4

A) mappa delle principali autostrade negli USA: un esempio di rete con distribuzione del grado di tipo Poisson.
B) mappa delle principali linee aeree nazionali negli USA: un esempio di rete con distribuzione del grado a legge di potenza

Distribuzioni a legge di potenza

Le distribuzioni a legge di potenza costituiscono una famiglia di distribuzioni caratterizzate dall'aver una "lunga coda" cioè la funzione di probabilità tende a zero lentamente all'aumentare del valore della variabile. Questa caratteristica fa sì che le leggi di potenza forniscano probabilità non trascurabili di riscontrare valori della variabile anche estremamente elevati. Le leggi di potenza pur essendo note e utilizzate nella letteratura tecnica e nella statistica da parecchi decenni, hanno avuto una ripresa di interesse sia per effetto delle ricerche legate alle reti complesse, sia, più in generale, perché si è riscontrato che permettono di interpretare innumerevoli fenomeni fisici e serie di dati sperimentali di natura e origine molto diversa. Per un approfondimento, si rimanda a due recenti lavori di rassegna sull'argomento [1, 2] molto chiari ed esaurienti, e che tracciano anche una illuminante prospettiva storica sull'origine e sull'uso di queste leggi.

Per dimostrare la versatilità delle leggi di potenza, il lavoro [2] cita 12 esempi semplici e reali di grandezze di origine molto diversa fra loro che seguono una legge di potenza, almeno verso la coda della distribuzione. Gli esempi sono: ricorrenza di parole, citazioni in lavori scientifici, accessi a pagine web, copie di libri venduti, chiamate telefoniche, grandezza di terremoti, diametro dei crateri lunari, intensità delle esplosioni solari, intensità delle guerre, reddito netto dei cittadini americani, popolazione delle città negli USA.

Il primo studioso che individuò che l'ammontare dei patrimoni seguiva una legge di potenza fu l'economista e sociologo Vilfredo Pareto (1848-1923) per cui queste leggi sono a volte note come leggi di Pareto, mentre il linguista G.K. Zipf (1902,1950) introdusse metodi statistici nello studio del linguaggio, e trovò leggi di potenza per la ricorrenza delle parole. Una bibliografia aggiornata sulla legge di Zipf (e sulle leggi di potenza) è mantenuta nel sito <http://www.nslj-genetics.org/wli/zipf/index.html>: il primo lavoro citato è del 1881 e la bibliografia contiene ad oggi 555 riferimenti.

Una legge di potenza è caratterizzata dall'aver una densità di probabilità $P(x)$ (dove $P(x)$ rappresenta la probabilità di trovare la variabile compresa fra x e $x + dx$) data da:

$$P(x) = cx^{-\gamma} \quad (1)$$

Dove $\gamma > 0$ è l'esponente che caratterizza la legge di potenza e c è una costante di normalizzazione che non ha rilevanza nel determinare il comportamento della funzione e viene calcolata in modo che l'area sottesa dalla funzione di densità $P(x)$ sia uguale a 1 (essendo $P(x)$ una densità di probabilità la sua area totale deve essere uguale a 1). La legge (1) deve essere definita per un valore di x maggiore di un valore minimo x_{\min} (per $x = 0$ la funzione $P(x)$ tende all'infinito). Si suole quindi affermare che una variabile segue una legge di potenza a partire da un certo valore minimo, cioè quando è la coda della distribuzione a soddisfare la legge (1).

Il valor medio della variabile $\langle x \rangle$ è finito solo per $\gamma > 2$ (per $\gamma \leq 2$ è infinito); la varianza diverge per $\gamma \leq 3$. La distribuzione mantiene le proporzioni al variare della scala. Se in un disco di un computer i file da 2 KB sono il 25% dei file da 1 KB, la legge predice che i file da 2 MB saranno il 25% dei file da 1 MB. La mancanza di un picco attorno al valore medio e il mantenimento delle proporzioni al variare della scala hanno indotto a definire questa legge come legge a invarianza di scala (*scale free*).

Prendendo il logaritmo naturale di entrambi i membri nella formula (1) si ottiene $\ln P(x) = -\gamma \ln x + \ln c$; quindi in un diagramma log-log la legge è rappresentata con una retta a pendenza negativa e il coefficiente angolare della retta è il valore dell'esponente γ (Figura 4 B).

Quali sono alcune possibili conseguenze pratiche di una legge di potenza.

Esempio 1 - Il reddito netto dei cittadini americani segue una legge di potenza con $\gamma = 2.1$ [2]. Questo comporta che il 50% della popolazione più ricca detenga il 94% del reddito nazionale, e che 80% del reddito nazionale sia nelle mani del 20% degli individui più ricchi. Questa è la regola "80/20" attribuita a Pareto.

Esempio 2 - L'altezza degli esseri umani segue tipicamente una legge a campana con valor medio attorno a $\langle x \rangle = 1.75$ m, quindi una legge molto lontana da una legge di potenza. Se l'altezza degli umani seguisse una legge di potenza con esponente $\gamma = 2.5$ un individuo ogni 100.000 sarebbe alto 15 m, e uno ogni milione sarebbe alto 30 m.

RIFERIMENTI BIBLIOGRAFICI

- [1] Mitzenmacher M: A brief history of generative models for power law and lognormal distributions. *Internet Mathematics*, Vol. 1, 2004, p. 226-251.
[2] Newman M.E.: Power laws, Pareto distributions and Zipf's laws. *Contemporary Physics*, Vol. 46, 2005, p. 323-351.

maticamente riportata la rete dei principali collegamenti aerei nazionali negli USA. Alcuni aeroporti fungono da connettori (e sono effettivamente chiamati *hub* in inglese); per viaggiare da un aeroporto ad un altro, in mancanza di un collegamento diretto, si deve effettuare una coincidenza in un *hub*.

Il grafico a destra in alto nella figura 4 riporta la distribuzione del grado $P(k)$ per la rete delle linee aeree in scala bilogarithmica. È evidente, anche ad una prima e superficiale ispezione visiva, la differente struttura della rete autostradale (Figura 4 A) e della rete delle linee aeree (Figura 4 B).

Le reti a legge di potenza sono frequenti in natura e nella realtà. Perché si formano reti

con questa struttura e quale è il meccanismo responsabile della loro generazione? Quali sono le proprietà principali indotte da una topologia a legge di potenza e perché questa struttura organizzativa è spesso privilegiata rispetto ad altre? La risposta a queste domande venne affrontata per la prima volta in maniera semplice e brillante da Barabasi e Albert in [3]. La loro spiegazione parte da una semplice constatazione: le reti complesse sono sistemi aperti che evolvono nel tempo crescendo di dimensione in quanto nuovi nodi si aggiungono man mano a quelli preesistenti. Sia i modelli casuali di Erdős e Rényi che quelli a piccolo mondo di Watts a Strogatz sono modelli statici, pen-

sati per un numero fisso di nodi per i quali si cercano opportune regole per il tracciamento degli archi. Nelle reti in evoluzione nuovi nodi si generano continuamente e i nuovi nodi devono connettersi ai nodi preesistenti sulla base di opportune regole. Gli autori fanno l'ipotesi che il meccanismo di connessione dei nuovi nodi al resto della rete sia ispirato al principio del "aggancio preferenziale (*preferential attachment*)", cioè i nuovi nodi tendono preferenzialmente a connettersi ai nodi con grado più alto. Gli esempi che possono ispirare un meccanismo di aggancio preferenziale sono molteplici. Se si apre un nuovo aeroporto è ragionevole pensare che i primi collegamenti aerei che verranno resi operativi saranno verso gli aeroporti principali della regione (gli *hub*). In un articolo scientifico è facile che venga citato un lavoro che ha avuto notevole impatto sulla comunità scientifica e che ha già avuto molte citazioni. I link che verranno indicati su una pagina *www* di nuova generazione, saranno con grande probabilità rivolti verso i siti di maggiore frequentazione.

Partendo da questi due semplici presupposti (reti in evoluzione e aggancio preferenziale), Barabasi e Albert simularono la crescita di una rete virtuale, e, con ragionevole e legittima soddisfazione, trovarono conforto alle loro supposizioni, in quanto al crescere del numero di nodi immessi nella rete la distribuzione del grado $P(k)$ assumeva la forma di una legge di potenza con esponente $\gamma = 3$.

Le reti generate dal modello di Barabasi e Albert (modello BA) vennero chiamate dagli autori reti a *invarianza di scala* (*scale free*). Nelle reti casuali, in cui la distribuzione del grado segue una curva a campana, è definito un grado medio $\langle k \rangle$ che determina la *scala* della rete, cioè determina il numero di link normalmente presenti nei nodi essendo rare le forti deviazioni da questo valore. Nelle reti a legge di potenza, la curva di distribuzione non presenta un picco, la maggioranza dei nodi ha pochi archi emergenti, e questi innumerevoli piccoli nodi convivono con i grandi *hub*. Nelle distribuzioni a legge di potenza il valore medio (e la varianza) possono anche risultare infiniti per piccoli valori dell'esponente (vedi riquadro esplicativo di p. 11). Sia le reti di Erdős e Rényi sia quelle a piccolo mondo di Watts a Strogatz presu-

ponevano una struttura democratica della rete nel senso che il legame era ripartito in modo sostanzialmente uniforme sui nodi; le reti a invarianza di scala presuppongono un comportamento dei nodi fortemente disomogeneo, i pochi nodi connettori tendono ad aumentare i loro link durante l'evoluzione della rete mentre per gli altri nodi il numero di link rimane basso e sostanzialmente costante. Il sito *www* creato da Barabasi alla Notre Dame University riporta numerosi e suggestivi esempi di reti complesse <http://www.nd.edu/~networks/> con struttura a invarianza di scala.

Una rete BA a invarianza di scala può essere facilmente costruita attraverso il seguente algoritmo di generazione:

1. aggiunta di un nodo - partendo da un numero predefinito (piccolo) di nodi m_0 , ad ogni intervallo di tempo si aggiunge un nuovo nodo con $m \leq m_0$ archi che devono essere connessi ai nodi preesistenti della rete;

2. aggancio preferenziale - la probabilità che un arco del nuovo nodo sia connesso ad un preesistente nodo i dipende dal grado k_i secondo la legge $q_i = k_i / \sum_j k_j$ (vengono favoriti gli agganci ai nodi con grado k_i elevato).

Dopo t passi l'algoritmo ha generato una rete di $N = m_0 + t$ nodi e mt archi. È stato dimostrato che la presenza di entrambi gli ingredienti (crescita e aggancio preferenziale) è necessaria e sufficiente: la mancanza di uno dei due fa perdere la proprietà della legge di potenza. Le reti a invarianza di scala secondo il modello BA hanno piccola lunghezza caratteristica (il passaggio attraverso gli *hub* connette rapidamente tutti i nodi) ma hanno anche piccolo coefficiente di aggregazione normalmente minore di quello riscontrato nelle reti reali: i nodi periferici sono connessi agli *hub* ma non sono connessi fra di loro.

Sebbene il modello BA sia piuttosto semplicistico e rozzo (i nodi possono solo nascere e non morire, e i nodi più vecchi sono favoriti nel diventare gli *hub* della rete), tuttavia ha avuto una grandissima risonanza, in quanto mostrava che era possibile ipotizzare delle leggi di *auto-organizzazione* nei sistemi fisici complessi e queste leggi di auto-organizzazione portavano naturalmente a generare leggi di potenza. Sulla base di questi primi significativi risultati, vari ricercatori e vari gruppi di ricerca si misero a studiare leggi auto-

organizzative più rispondenti al mondo fisico, secondo le seguenti direttrici:

- i nodi e gli archi possono essere ridiretti e possono anche scomparire (si modifica o si elimina un link, si chiude una pagina web);
- i nodi possono invecchiare modificando la loro capacità di attrazione (sono state proposte varie leggi di aggancio preferenziale più complesse di quella BA originaria);
- i nodi competono fra di loro per attrarre i link dei nodi di nuova generazione. I nodi hanno un coefficiente di competitività o di buona salute (*fitness*) che ne esalta il potere di attrazione. Il modello a *fitness* introduce un meccanismo per cui non sempre i nodi più vecchi sono i più favoriti nella formazione degli hub, e si è reso necessario per spiegare fenomeni tipo Microsoft o Google, cioè di attori che entrati sulla scena per ultimi riescono a

imporre molto rapidamente a scapito di nodi più vecchi;

- la rete cresce con un meccanismo di aggancio preferenziale, ma i nodi possono essere attivi o non-attivi e passano da una condizione all'altra attraverso opportune leggi di probabilità; utilizzando questo accorgimento Klemm e Eguíluz [11] hanno ottenuto reti a invarianza di scala con alto un coefficiente di aggregazione;
- il modello BA genera reti non orientate, mentre varie reti del mondo reale sono orientate (ad esempio il www per cui il modello BA era stato inizialmente pensato).

Un esempio tipico di rete con crescita auto-evolutiva, e che anzi è stata quella che più di altre ha spinto gli studiosi ad affrontare lo studio delle reti complesse è il www. La struttura del www come rete complessa è richiamato nel riquadro. La rete Internet è pure una

La struttura della rete World Wide Web

La rapida diffusione del World Wide Web ha già fortemente inciso sull'organizzazione della società moderna e il suo impatto è certamente destinato a crescere. L'immagazzinamento, l'acquisizione e la trasmissione dell'informazione tende sempre più ad essere delegata al www soppiantando tutte le forme precedenti, in particolare quelle cartacee. I motori di ricerca hanno accesso ad un numero di documenti enorme e impensabile fino a pochi anni fa; i criteri di selezione dei documenti sono ancora rozzi (la corrispondenza di sequenze di lettere o parole) ma sono comunque incomparabilmente più potenti di qualunque strumento precedente.

Il www ha un'altra particolarità: è una rete totalmente auto-organizzata e con un tasso di crescita straordinariamente veloce: non c'è nessun supervisore centrale che ne controlli lo sviluppo e ne gestisca l'evoluzione. Il www può essere considerata una struttura paradigmatica per reti auto-evolutive e la comprensione della sua struttura ha originato e fortemente influenzato lo studio delle reti complesse. Quale è quindi la struttura della rete che chiamiamo www?

Innanzitutto il www è una rete orientata, in quanto i link inclusi in una pagina sono dei puntatori direzionali: permettono con un click del mouse di passare dalla pagina corrente a quella puntata dal link. Non è quindi definibile un unico grado di connettività ma si dovrà distinguere fra grado di ingresso (in-degree) e grado di uscita (out-degree) caratterizzati da due distinte distribuzioni $P_{in}(k)$ e $P_{out}(k)$. $P_{in}(k)$ fornisce la probabilità che k link puntino ad una certa pagina, mentre $P_{out}(k)$ fornisce la probabilità che una pagina abbia k link in uscita. È stato verificato sperimentalmente che entrambe le distribuzioni appartengono alla classe delle distribuzioni a legge di potenza con un esponente γ_{in} leggermente inferiore a γ_{out} [6]. La discrepanza fra i due valori può essere ragionevolmente spiegata dal diverso processo evolutivo dei gradi di ingresso e uscita. Il grado in ingresso non ha nessun estremo superiore, dipende da quante altre pagine decidono di indirizzare la pagina data. Pagine molto popolari (per esempio i più noti motori di ricerca) sono indirizzati da un numero enorme e crescente di pagine. Il grado in uscita è invece determinato dal numero di link esplicitamente posizionati su una pagina, che sono definiti dal proprietario della pagina stessa; in questo caso c'è anche un limite fisico al numero di link scrivibili in uno spazio limitato.

Non si conosce il numero esatto di pagine del www, ed è praticamente impossibile contarle tutte in modo rigoroso. Attualmente dovrebbe superare abbondantemente parecchi miliardi (una stima del 2000 parlava di 800 milioni di pagine). Ciò nonostante, il www è un piccolo (abbastanza piccolo) mondo: misure di cammino caratteristico fatte sperimentalmente su porzioni ristrette del www e estrapolate all'intera rete portano a stime di 19 passi di separazione. Il numero medio di link per pagina (cioè il grado medio) di uscita è circa 7. Con questi valori, in diciannove passi si potrebbero raggiungere circa $7^{19} \approx 10^{16}$ pagine, un numero sicuramente molto superiore al numero di pagine realmente esistenti.

Riprendendo la classificazione dei nodi per grafi orientati di figura 1 B, si intuisce che anche il www può essere suddiviso in tre aree principali. Un'area centrale che contiene pagine tutte connesse fra loro. In questa zona si possono raggiungere tutte le pagine saltando da una pagina all'altra attraverso un click del mouse. È la zona centrale del web e quella più facilmente raggiungibile dai motori di ricerca e quindi accessibile a tutti i navigatori.

Esiste poi una zona che chiameremo *Zona-in* in cui le pagine sono, in parte connesse fra di loro, e connesse alla zona centrale ma non viceversa. Sono pagine normalmente inaccessibili ai motori di ricerca. Quando un individuo costruisce una nuova pagina web, può indicare tutti i link che vuole sulla sua pagina, ma se nessun link di nessun'altra pagina punta a lui, rimane nella *Zona-in*, invisibile al resto della rete: non basta creare una pagina per entrare nel gran gioco del www. Alcuni motori di ricerca forniscono anche la possibilità di conoscere quanti link puntano ad una data pagina con un dato indirizzo. Per conoscere il numero di pagine che puntano ad un ipotetico indirizzo www.xxx.yyy, bisogna lanciare il seguente comando link: <http://www.xxx.yyy>.

La *Zona-out* è una zona costituita da siti raggiungibili dalla zona centrale, (e quindi anche dalla *Zona-in*) ma una volta entrati non permettono di essere abbandonati. Molti siti commerciali hanno queste caratteristiche: invogliano i navigatori ad entrare ma non li lasciano più scappare (si ripete nel cyberspazio il mito di Ulisse e della Maga Circe).

I motori di ricerca perlustrano continuamente il www alla ricerca di nuove pagine per catalogare e indicizzare nuove informazioni. Si stima, tuttavia, che anche i motori più affermati abbiano una visibilità del www non superiore al 50%.

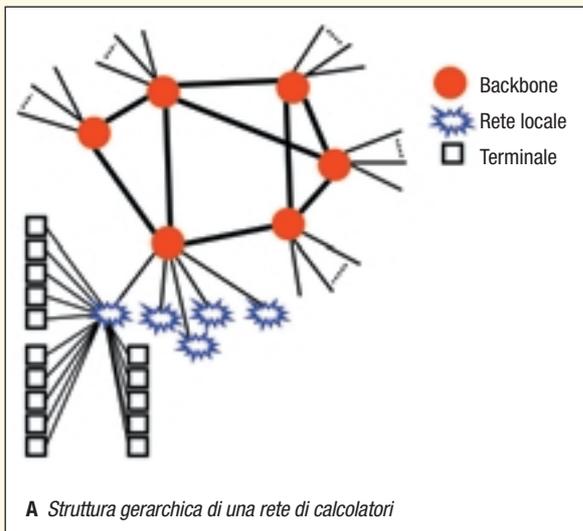
La struttura della rete Internet

La rete Internet è costituita da una collezione di migliaia di reti più piccole, ciascuna sotto il controllo di amministratori locali, e non è possibile avere un quadro esaustivo e completo della sua topologia. Tuttavia, nell'euforia e nell'entusiasmo per la scoperta di fenomeni di diversissima natura che manifestavano un andamento a legge di potenza, nel 1999 venne presentato un articolo nell'importante conferenza SIGCOMM (*Special Interest Group on Data Communication*) dell'ACM (*Association for Computing Machinery*) in cui si affermava di aver sperimentalmente verificato che la rete Internet presentava un andamento ad invarianza di scala con un esponente $\gamma = 2.5$ [1]. Tale risultato fece subito pensare che la topologia di Internet potesse essere spiegata con un modello a crescita del tipo BA, e che di questo modello ereditasse le proprietà.

Successive indagini più accurate [2] hanno messo in luce che la presenza di una legge di potenza nella distribuzione del grado non implica necessariamente la presenza di connettori o *hub* come nelle reti BA, ma sono anche possibili topologie diverse, in cui i nodi a grado più elevato sono localizzati alla periferia della rete. Nel caso della rete Internet i collegamenti fisici fra nodi possono avere ampiezza di banda (capacità di trasmissione) molto disuniforme e quindi non è corretto assegnare a tutti gli archi lo stesso peso (vedi paragrafo 5). È necessario considerare grafi pesati, in cui ad ogni arco viene assegnato un valore proporzionale alla sua capacità trasmissiva. Nei grafi pesati la domanda che è giusto porsi non è quale sia la probabilità che due nodi della rete siano connessi (affidabilità) ma quale sia la probabilità che fra due nodi della rete sia presente una assegnata ampiezza di banda (qualità del servizio).

Nella rete internet, inoltre, il grado di connettività di ogni nodo è limitato da considerazioni ingegneristiche e da vincoli tecnologici, dovuti alla capacità massima del nodo stesso di trasmettere flussi di bit. In sostanza un nodo può avere pochi collegamenti a larga banda o un maggior numero di collegamenti a banda stretta, in modo da rispettare il vincolo sul traffico massimo consentito dalla tecnologia del nodo [2]. Queste considerazioni hanno favorito la crescita e lo sviluppo della rete internet con una topologia gerarchica (schematizzata nella Figura A), in cui i nodi in un primo livello (backbone) sono connessi fra loro con linee ad alta capacità; da questi nodi centrali si dipartono reti locali che a loro volta gestiscono i computer terminali. Quindi il grado di connettività dei nodi tende a crescere andando verso la periferia della rete.

Esistono nodi con un numero molto elevato di link, che determinano l'osservata legge di potenza, ma che non hanno la posizione e la funzione



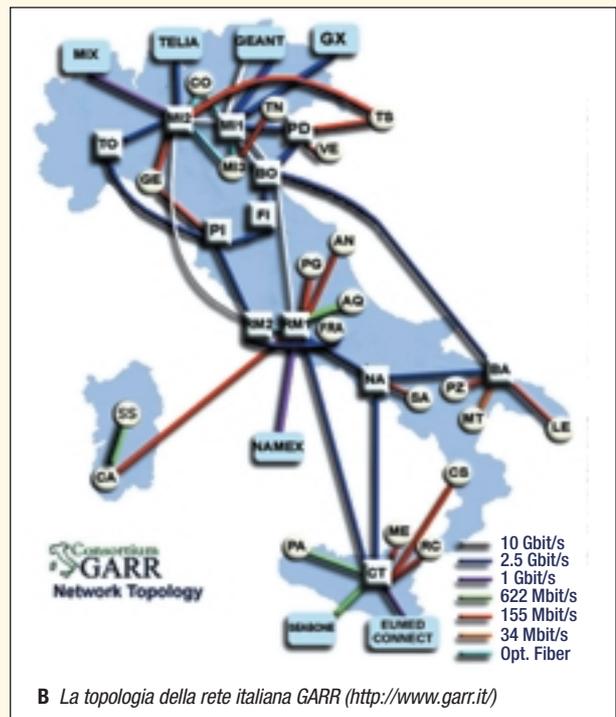
di connettori; questi nodi di alto grado gestiscono reti locali periferiche e il loro non funzionamento ha ripercussioni solo locali. La parte centrale della rete (backbone), quella che sostiene la maggior parte del traffico e che garantisce la qualità del servizio anche fra reti locali dislocate geograficamente in zone molto distanti, ha una struttura più vicina ad una rete poissoniana, e come tale possiede una elevata ridondanza dovuta a possibili percorsi alternativi.

Per esempio, la rete italiana GARR (*Gestione Ampliamento Rete Ricerca*) a cui aderiscono tutti gli enti che rappresentano la comunità accademica della ricerca scientifica in Italia ha la topologia rappresentata nella figura B. Nella figura B sono riportati i nodi da cui si dipartono le reti locali gestite dagli enti aderenti al sistema nella zona geografica di pertinenza, e sono anche evidenziate le larghezze di banda dei canali fisici di connessione, che vanno da 10 Gbit/s (Giga bit per secondo) a 34 Mbit/s (Mega bit per secondo).

RIFERIMENTI BIBLIOGRAFICI

[1] Faloutsos M., Faloutsos P., Faloutsos C.: On the Power-law Relationships of the Internet Topology. (*ACM-SIGCOMM99*) *Computer Communication Review*, Vol. 29, 1999, p. 251-262.

[2] Lun Li, Alderson D., Willinger W., Doyle J.: A First-Principle Approach to Understanding Internet's Router-level Topology. *ACM-SIGCOMM2004*.



rete in evoluzione, ma vincoli ingegneristici sulla capacità trasmissiva dei collegamenti ne condizionano e guidano la crescita deter-

minando una struttura, pure a legge di potenza, ma che si discosta dal semplice modello BA (vedi riquadro).

Le regole di evoluzione dei modelli auto-organizzativi dipendono da comportamenti locali (regole di aggancio di un nodo di nuova generazione) ma determinano comportamenti globali del sistema. Avendo scoperto che l'emergere in natura di reti complesse con distribuzione del grado a legge di potenza può derivare dal meccanismo di crescita della rete accompagnato da semplici principi auto-organizzativi, viene naturale la domanda, perché la natura ha scelto in molti casi questi processi di evoluzione o comunque processi di evoluzione che portano alle leggi di potenza? La risposta a questa domanda non è univoca e inequivocabile, ma alcuni ricercatori la fanno risalire al modo con cui le reti si comportano in presenza di fenomeni di degradazione, invecchiamento o attacco.

6. TOLLERANZA AI GUASTI E AGLI ATTACCHI

I sistemi fisici possono manifestare nel tempo vari meccanismi di degradazione, usura, malfunzionamento o guasto. Fenomeni cioè che tendono in vario modo a ridurre l'efficienza e la connettività del sistema. Si è già osservato che, in una rete non orientata, un arco per nodo è sufficiente per connettere tutti i nodi fra di loro cioè a permettere di raggiungere qualunque nodo partendo da qualunque altro nodo. Le reti reali sono normalmente molto più connesse di questa soglia minima. Le stime riportate da vari autori non sono molto uniformi: seguendo i quadri riepilogativi riportati in [6] si trovano i seguenti valori per il grado medio $\langle k \rangle$: per la rete internet un grado medio di circa 4, per il www un grado medio attorno a 7, per la rete degli attori cinematografici un grado medio di 61. Questo valore di grado medio maggiore di uno, comporta una intrinseca ridondanza nella rete, cioè il fatto che si possano raggiungere due nodi qualunque attraverso una molteplicità di cammini distinti. Nel grafo non-orientato di figura 1 A, partendo dal nodo 1 si può raggiungere il nodo 6 attraverso numerosi cammini: i più brevi sono, per esempio, il cammino *de*, il cammino *abc*, il cammino *afe*, *djk* ecc¹.. Se per qualche motivo il cam-

mino *abc* fosse interrotto, non si pregiudica la possibilità di un collegamento fra 1 e 6 in quanto sono percorribili strade alternative. Ovviamente in questo ragionamento si suppone che qualunque cammino connesso abbia lo stesso valore. Nella pratica i cammini hanno associato un peso (lunghezza, capacità, costo ecc.) che rende preferibili alcuni percorsi (in generale i più brevi o i più veloci) rispetto ad altri. Dovendo andare in treno da Torino a Reggio Calabria, se tutte le altre linee fossero interrotte, si potrebbe trovare un collegamento passante da Bolzano: il collegamento sarebbe garantito ma gli utenti avrebbero comunque qualcosa da ridire! Nelle reti informatiche la larghezza di banda (o capacità trasmissiva) del collegamento determina la velocità della comunicazione e rende quindi preferibili alcuni cammini (non necessariamente i più corti) rispetto ad altri.

La forza e la robustezza delle reti risiede nella loro intrinseca ridondanza. Per indagare questa problematica più da vicino alcuni studiosi hanno cominciato a porsi il problema di come valutare la ridondanza di una rete complessa, e di come le proprietà di robustezza fossero influenzate dalla struttura topologica della rete. Un malfunzionamento in un arco o in un nodo perturba il funzionamento del sistema localmente, ma può avere scarse ripercussioni sulle proprietà globali del sistema; solo all'aumentare dei nodi o degli archi interrotti cominciano a manifestarsi ripercussioni globali che incidono sui valori dei parametri caratteristici della rete. Per reti di piccole dimensioni è possibile calcolare in modo accurato l'affidabilità della rete, cioè la probabilità che due nodi siano connessi date le probabilità di fallimento di archi e nodi, attraverso algoritmi esaustivi che cercano tutti i possibili cammini fra due nodi [12]. Per reti di grandi o grandissime dimensioni è necessario ricorrere ad argomenti statistici.

Il guasto di un elemento della rete, sia esso nodo o arco, comporta la perdita di operatività dell'elemento stesso, e può essere emulato rimuovendo l'elemento guasto dalla rete. La rimozione di un arco ha influenza solo sul legame fra i due nodi connessi e su tutti i

¹ Si noti, incidentalmente, che la direzionalità degli archi modifica sostanzialmente la connettività della rete. Nel grafo orientato di figura 1 B, l'unico cammino che connette 1 con 6 è il cammino *afe*.

cammini che avrebbero attraversato quell'arco. La rimozione di un nodo implica, invece, la disconnessione di tutti gli archi in ingresso e in uscita dal nodo.

Gli studi in questa direzione hanno cercato di valutare la reazione di una rete alla rimozione progressiva di suoi elementi, cioè la tolleranza della rete ai guasti in funzione della topologia della rete stessa. Prendendo esempio dalle reti informatiche, sono state individuate e studiate due principali categorie di guasti:

□ *guasti casuali*: ogni elemento della rete ha la stessa probabilità di guastarsi indipendentemente dalla sua posizione e dalle sue proprietà (un nodo della rete internet può andare fuori servizio per innumerevoli motivi, un fulmine può cadere su una linea ferroviaria in qualunque punto della rete nazionale, un black-out può causare la sconnessione della rete elettrica). Per emulare questo tipo di guasti, i nodi della rete vengono via via rimossi con probabilità uniforme;

□ *guasti maliziosi o attacchi preferenziali*: sono guasti provocati intenzionalmente allo scopo di indurre il maggior danno sulla connettività della rete. Per i sistemi informatici sono queste le minacce più gravi alla sicurezza del sistema. Emulare gli attacchi di questo tipo, comporta iniziare a rimuovere i nodi partendo da quelli di grado più alto.

Le prime ricerche furono indirizzate a esplo-

rare le proprietà di connettività della rete in funzione della frazione di nodi rimossi, e in funzione della topologia della rete. Indicando con f la frazione di nodi rimossi, in [13] si è indagato sull'esistenza di una frazione critica f_c per cui la rete perde la sua coesione.

6.1. Reti casuali

Nel caso in cui i guasti indotti siano di tipo casuale, all'aumentare della frazione f di nodi rimossi, il cammino caratteristico L cresce lentamente (Figura 5, tratta da [9]). All'aumentare di f si può identificare una frazione critica f_c per cui, quando $f > f_c$, la rete si disgrega in piccole sottoreti fra loro non connesse. Tale effetto si verifica per valori di f_c dell'ordine del 3% [13]. La rete manifesta un comportamento molto simile anche in presenza di attacchi mirati in cui i nodi sono rimossi a partire da quelli con grado più alto. La figura 5 mostra che nel caso di reti casuali (EXP) la crescita di L dovuta a guasti casuali (*failure*) o attacchi (*attack*) è molto simile.

6.2. Reti a invarianza di scala

In questo caso l'effetto di guasti casuali o attacchi mirati è molto diverso. Nel caso di guasti casuali, il cammino caratteristico L cresce molto lentamente (Figura 5): nodi con grado piccolo sono in numero enormemente alto e una loro rimozione ha scarso effetto sulla con-

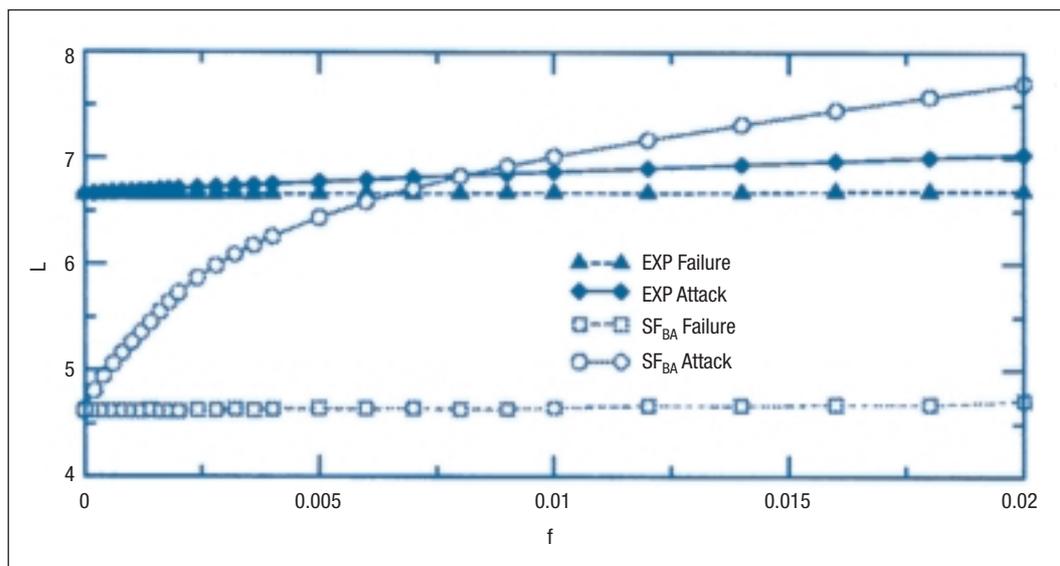


FIGURA 5

Variatione del cammino caratteristico L in funzione della frazione di nodi rimossi f per reti casuali (Exp) e a invarianza di scala con il modello BA (SF) – Figura tratta da [9]

nettività globale della rete. Si è dimostrato che, nel limite di reti che crescono all'infinito, non è possibile identificare una frazione critica f_c , in quanto la rete rimane comunque connessa attraverso gli hub. Ben diverso è il comportamento in presenza di attacchi mirati. Iniziando a colpire gli hub di grado più elevato, la topologia della rete viene drasticamente alterata: il cammino caratteristico L cresce rapidamente (Figura 5) e la rete perde ben presto le sue caratteristiche di connettività.

La conclusione può essere sommariamente formulata con queste parole: le reti casuali hanno un comportamento sostanzialmente simile sia rispetto ai guasti casuali che agli attacchi mirati e si disaggregano al superamento di una soglia critica. Le reti a invarianza di scala sono praticamente immuni rispetto ai guasti casuali ma sono fragilissime rispetto ad attacchi mirati. La fortissima tolleranza rispetto ai guasti casuali delle reti a invarianza di scala viene considerata uno dei fattori principali per giustificare la presenza di questa topologia in reti che evolvono naturalmente.

7. PROPAGAZIONE DELLE EPIDEMIE

La propagazione di malattie o virus (anche informatici) può essere affrontata mediante la teoria delle reti. La rete consiste di nodi (soggetti) sani e nodi (soggetti) infetti. Per modellare la propagazione dell'infezione si può supporre che un soggetto infetto abbia una probabilità v di propagare la malattia ad un soggetto sano a lui direttamente connesso, e che un soggetto sano abbia una probabilità δ di guarire. Indicando con $\lambda = v/\delta$ il tasso di diffusione dell'infezione si sono ottenuti i seguenti risultati principali [5]. In reti casuali l'infezione si propaga a tutti i nodi soltanto se il tasso di diffusione supera una soglia epidemica critica, altrimenti, l'epidemia rimane confinata in zone più o meno ristrette della rete e non riesce ad auto alimentarsi. In reti ad invarianza di scala, quando le dimensioni tendono all'infinito, non esiste soglia epidemica, ma l'epidemia assume le forme di una pandemia e si propaga a tutta la rete, per qualunque valore del tasso di propagazione. Nelle reti finite (anche a invarianza di scala), tuttavia, la soglia epidemica deve esistere altrimenti tutte le reti

biologiche a invarianza di scala sarebbero sparite da tempo, eliminate dalla inarrestabile propagazione di una infezione distruttiva. È vero, comunque, che le reti a invarianza di scala sono più vulnerabili rispetto alla propagazione di virus e malattie rispetto alle reti casuali. Ma esse sono anche più curabili (almeno in teoria). Curando i nodi a partire dagli hub, si può arrestare l'epidemia e ripristinare le condizioni di salute più rapidamente che in una rete casuale. Se invece, in una rete ad invarianza di scala, si applica una strategia di cura che agisce su nodi scelti a caso, la probabilità di arrestare la propagazione dell'infezione diventa molto piccola. Questi studi hanno, come ovvio, un enorme potenziale interesse per arginare e controbattere la distruttività di attacchi maliziosi a sistemi informatici e la propagazione di virus informatici di varia natura, oltre che per debellare epidemie in sistemi biologici (per esempio per sviluppare strategie di confinamento della diffusione dell'AIDS [4]). Per una più approfondita discussione su questi temi si rimanda al lavoro [14].

8. CONCLUSIONI

La teoria delle reti complesse ha avuto negli ultimi anni uno sviluppo travolgente. Ma nonostante centinaia di lavori scientifici siano stati pubblicati sulle riviste di tutto il mondo, essa può ancora considerarsi una scienza nella fase della prima infanzia da cui è lecito aspettarsi altri e più significativi avanzamenti. La speranza è quella di gettare luce su come i sistemi complessi evolvono e si auto-organizzano e su come la topologia della rete (indipendentemente dalle sue caratteristiche sociali, fisiche o tecnologiche) ne determina le proprietà e il comportamento. La comprensione di questi meccanismi guiderà verso la costruzione di reti sempre più efficienti, affidabili e sicure. La sfida alla complessità sarà sicuramente l'obiettivo più impegnativo ma anche il più stimolante per gli scienziati delle prossime generazioni.

Ringraziamenti

Nella compilazione del presente lavoro mi sono avvalso della preziosa collaborazione di Roberta Terruggia.

Bibliografia

- [1] Rita Levi Montalcini: *L'asso nella manica a brandelli*. Baldini & Castoldi 1998, p. 45.
- [2] Watts D.J., Strogatz S.H.: Collective dynamics of small world networks. *Nature*, Vol. 393, 1998, p. 493-592.
- [3] Barabasi A.L., Albert R.: Emergence of scaling in random networks. *Science*, Vol. 286, 1999, p. 509-512.
- [4] Barabasi A.L.: *Linked: the new science of networks*. Perseus, 2002 (traduzione italiana: A. Barabasi, "Link: la scienza delle reti", Einaudi, 2004).
- [5] Dorogovtsev S.N., Mendes J.F.F.: Evolution of networks. *Advances in Physics*, Vol. 51, 2002, p. 1079-1187.
- [6] Albert R., Barabasi A.L.: Statistical mechanics of complex networks. *Review Modern Physics*, Vol. 74, 2002, p. 47-97.
- [7] Newman M.E.: The structure and function of complex networks. *SIAM Review*, Vol. 45, 2003, p. 167-256.
- [8] Boccaletti S., Latora V., Chavez M., Hwang D.: Complex networks: structure and dynamics. *Physics Reports*, Vol. 424, 2006, p 175-308.
- [9] Crucitti P., Latora V., Marchiori M., Rapisarda A.: Efficiency of scale free networks: error and attack tolerance. *Physica A*, Vol. 320, 2003, p. 622-642.
- [10] Milgram S.: The small-world problem. *Physiology Today*, Vol. 2, 1967, p. 60-67.
- [11] Klemm K., Eguíluz V.M.: Highly clustered scale free network. *Physical Review*, Vol. 65, 2002, art. 036123.
- [12] Bobbio A., Ferraris C., Terruggia R.: *New challenges in network reliability analysis*. In: Proceedings of International Workshop on Complex Network and Infrastructure Protection - CNIP06, 2006, p. 554-564.
- [13] Albert R., Jeong H., Barabasi A.L.: Error and attack tolerance of complex networks. *Nature*, Vol. 406, 2002, p. 378-82.
- [14] Lloyd A., May R.M.: How viruses spread among computers and people. *Science*, Vol. 292, 2001, p. 1316.

ANDREA BOBBIO è professore di informatica presso l'Università del Piemonte Orientale A. Avogadro con sede ad Alessandria. Egli si è prevalentemente occupato di metodologie di analisi per sistemi stocastici e di teoria e metodi dell'affidabilità di sistemi. In quest'area di ricerca ha pubblicato numerosi lavori su riviste e conferenze internazionali, ed è stato responsabile di vari progetti di ricerca con enti pubblici e privati. È stato ospite ricercatore in importanti università in Europa, USA e India.

E-mail: bobbio@mfn.unipmn.it

URL: <http://www.mfn.unipmn.it/~bobbio/>

COMPETENZE INFORMATICHE E PRODUTTIVITÀ NEL SETTORE BANCARIO



Pier Franco Camussone

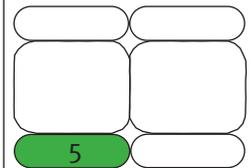
In due precedenti ricerche AICA e SDA Bocconi hanno cercato di valutare il “costo della ignoranza informatica” per l'intero sistema produttivo italiano e poi, nello specifico, per il settore sanitario. Ora l'attenzione viene rivolta al settore bancario, certamente più avanzato nell'impiego dell'ICT. In tal caso il costo dell'ignoranza si rivela modesto ma gli incrementi di produttività derivanti dalla formazione informatica sono molto interessanti. La ricerca, che si è basata anche su verifiche empiriche, fornisce una quantificazione al riguardo.

1. IL COSTO DELL'IGNORANZA INFORMATICA NELLA SOCIETÀ DELL'INFORMAZIONE

In questi ultimi anni l'uso delle tecnologie informatiche e telecomunicative di tipo digitale (ICT) si è largamente diffuso sia nel contesto in cui lavoriamo, sia in quello in cui trascorriamo il nostro tempo libero. I lavoratori dell'era digitale devono essere quasi tutti alfabetizzati nell'uso del PC e delle applicazioni di uso individuale più comune quali Internet, la posta elettronica, il word processing e il foglio di lavoro ecc..

La preparazione adeguata della forza lavoro per la “società dell'informazione” che si desidera costruire in Europa può rappresentare però il tallone d'Achille di questa strategia di sviluppo [11]. La formazione in ICT non può fermarsi alla fine dei corsi scolastici. Le aziende devono favorire ed incoraggiare l'aggiornamento delle conoscenze informatiche. Per i cosiddetti *generic user* l'azienda deve procedere con un intervento di formazione più estensivo, che intensivo, senza curarsi se ciò che viene insegnato possa tor-

nare utile subito nella mansione attualmente svolta dal lavoratore. Insegnando i rudimenti dell'ICT si creano potenziali utenti, che possono più facilmente adattarsi a cambi di mansione e possono intravedere essi stessi miglioramenti organizzativi nelle modalità di svolgimento dei propri compiti [10]. E qui si introduce una nuova riflessione. La formazione tecnica sugli strumenti informatici dovrebbe essere affiancata da corsi con un taglio più organizzativo, in cui si illustrino le nuove modalità di lavoro che l'ICT consente [12]. Come osservato da Danziger gli aumenti di produttività dipendono da come le persone sanno usare gli strumenti informatici, da come rivedono e semplificano il proprio lavoro grazie a questi strumenti, per ricavare spazi di tempo da dedicare ad attività più ricche e che possano determinare un maggior valore aggiunto [9]. In altri casi non è il risparmio di tempo di lavoro che ci si deve aspettare dall'uso dell'ICT, quanto piuttosto un miglioramento dell'output del lavoro medesimo che risulta più completo e più ricco di contenuti. Le misurazioni



dei vantaggi tangibili hanno rilevato che entrambe queste aspettative trovano riscontro nella verifica empirica [6].

AICA e la *Scuola di Direzione Aziendale* (SDA) della Bocconi hanno lanciato tre anni or sono un piano di ricerca, mirato alla valutazione del “costo della ignoranza informatica” nel sistema produttivo del nostro Paese. Due fasi di tale campagna di ricerca si sono concluse negli anni scorsi ed i risultati sono già stati presentati in articoli apparsi su questa rivista¹.

La prima indagine [7], che ha preso in esame l'intero contesto produttivo italiano, ha evidenziato un costo di circa 15 miliardi di euro l'anno per il sistema economico del nostro Paese derivante dalla scarsa competenza informatica degli utenti di applicazioni informatiche (circa 6.870.000 lavoratori). La seconda ricerca [4], che si è focalizzata sul settore della sanità (1.280.000 addetti), ha mostrato un significativo gap tra le conoscenze informatiche necessarie in questo contesto e gli skill posseduti dagli appartenenti a questo settore. Il costo di tale lacuna

è stato valutato in circa 860 milioni di euro l'anno, una cifra superiore addirittura alla spesa informatica annuale dell'intero settore sanitario. A questo punto, avendo misurato gli effetti della ignoranza informatica in generale, e poi in uno specifico settore non ritenuto avanzato sotto il profilo informatico, si è voluto conoscere la situazione in una realtà ad alta intensità d'uso dell'ICT quale è il settore bancario.

2. IL SETTORE BANCARIO

Il settore della intermediazione finanziaria è un comparto a forte intensità di lavoro intellettuale (*brain intensive*). Secondo l'ultimo censimento ISTAT sulle attività produttive in Italia il settore della intermediazione finanziaria era composto nel 2001 da 363.000 dipendenti bancari e 29.000 dipendenti di società par bancarie (Tabella 1). Questi valori sono sostanzialmente il linea con quanto riportato nella relazione annuale della Banca d'Italia del 2005 secondo cui i dipendenti delle banche italiane (con l'esclusione della

TABELLA 1
I dipendenti
del settore
finanziario in Italia
(Fonte: Censimento
ISTAT 2001)

Categorie ISTAT	
65110 - Attività delle banche centrali	9.099
65121 - Attività delle banche commerciali	275.055
65123 - Attività delle banche cooperative e mutue	79.729
Totale dipendenti settore bancario	363.883
65210 - Leasing finanziario	5.405
65221 - Servizi di credito al consumo	6.601
65222 - Istituti e sezioni per il credito speciale	507
65231 - Società fiduciarie e di investimenti mobiliari	3.567
65232 - Società per la gestione dei fondi comuni di investimento	3.905
65233 - Altre società finanziarie	9.407
Totale dipendenti settore par bancario	29.392
Totale dipendenti settore bancario e par bancario	393.275

¹ Si veda: Camussone: Il costo dell'ignoranza informatica. *Mondo Digitale*, giugno 2003. Camussone: Formazione informatica e produttività. *Mondo Digitale*, settembre 2003. Camussone: Il costo dell'ignoranza informatica nella sanità. *Mondo Digitale*, settembre 2004. Camussone: Formazione informatica e produttività nel sistema sanitario italiano. *Mondo Digitale*, dicembre 2004.

Banca Centrale) assommano a 337.000, un valore inferiore che tiene conto degli effetti delle fusioni avvenute tra molti operatori bancari nel frattempo. I dipendenti resi disponibili dal settore bancario sono stati quasi sempre assorbiti dal settore parabancario, che nel frattempo si è espanso. Per tali ragioni possiamo ritenere che nel settore bancario e parabancario italiano lavorino attualmente circa 390.000 addetti complessivamente, di cui 340.000 nelle banche vere e proprie, 8.200 nell'istituto centrale (Banca d'Italia), e i rimanenti nelle aziende par bancarie.

Una delle caratteristiche distintive del personale bancario è rappresentata dal livello di scolarizzazione e professionalità degli addetti. La percentuale dei laureati è compresa tra il 30 e il 40% e in alcune aree particolari della banca si supera anche il 50%, come nel caso delle funzioni centrali. Per tali ragioni la retribuzione media degli addetti risulta elevata e, secondo i valori forniti dalla Banca d'Italia, il costo aziendale medio di un dipendente bancario in Italia ammonta a 65.600 € l'anno². Il tempo di lavoro del personale è quindi "prezioso", forse più di quello di altri lavoratori appartenenti a settori differenti. Questa è una delle ragioni per le quali le banche guardano con interesse all'impiego dell'ICT nelle loro aziende: secondo una diffusa convinzione, rilevata presso le direzioni delle banche italiane, queste tecnologie sono ritenute degli strumenti per la razionalizzazione delle attività operative, cioè per rendere più efficiente il lavoro interno del proprio personale e per spostare sui clienti una parte di tale lavoro (*self service banking* e *virtual banking*). L'impiego dell'informatica è visto principalmente come un mezzo di "cost saving" piuttosto che come una leva per il miglioramento del prodotto e del servizio bancario, vale a dire una arma per competere sui mercati [15, 17, 20].

Il personale bancario è costituito per due

terzi da impiegati e per quasi un altro terzo da quadri operativi. Sono queste due categorie di addetti che svolgono la quasi totalità del lavoro, trattandosi di attività di concetto che consistono nel trattamento delle informazioni che precedono, documentano e permettono di controllare le operazioni finanziarie. I dirigenti sono una categoria molto ristretta (Figura 1).

Nello svolgimento del lavoro bancario si riscontra un forte utilizzo delle infrastrutture informatiche. Secondo quanto rilevato dalla nostra indagine³ il tempo medio di utilizzo del computer per un dipendente bancario si aggira nell'intorno del 72% del suo tempo lavorativo settimanale. Si tratta naturalmente di un valore medio ponderato che è la risultante di una media tra il tempo speso al computer dai quadri (circa 50% del proprio tempo lavorativo) e il tempo speso dagli impiegati che è ben superiore e raggiunge addirittura l'80%. Se osserviamo i risul-

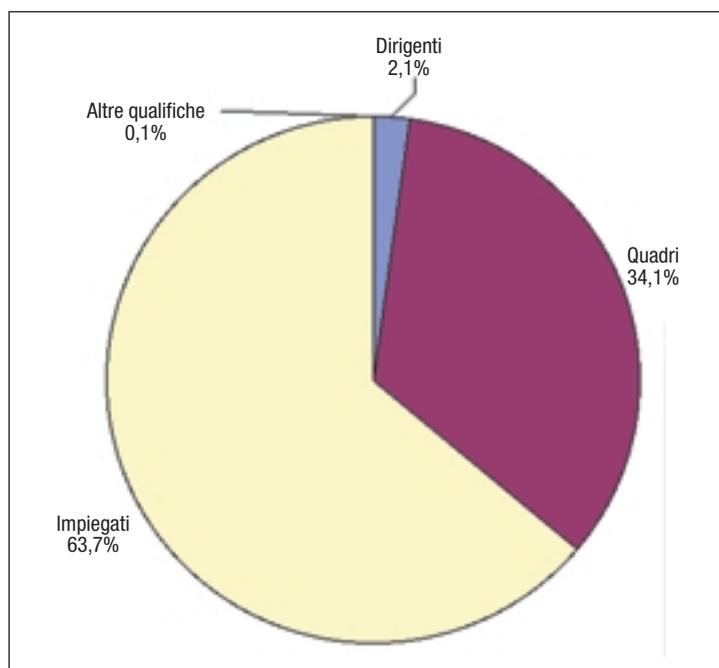


FIGURA 1
La composizione del personale bancario

² Relazione annuale del Governatore della Bdl 2005.

³ Sono stati sentiti tramite un questionario 169 utenti appartenenti ad una sessantina di banche differenti. Per l'illustrazione del metodo della ricerca e dei questionari utilizzati si rimanda alla parte terza del rapporto finale: Camussone P.F., Occhini G., Santececca D.: Competenze digitali e produttività nel settore bancario italiano, McGrawHill 2006.

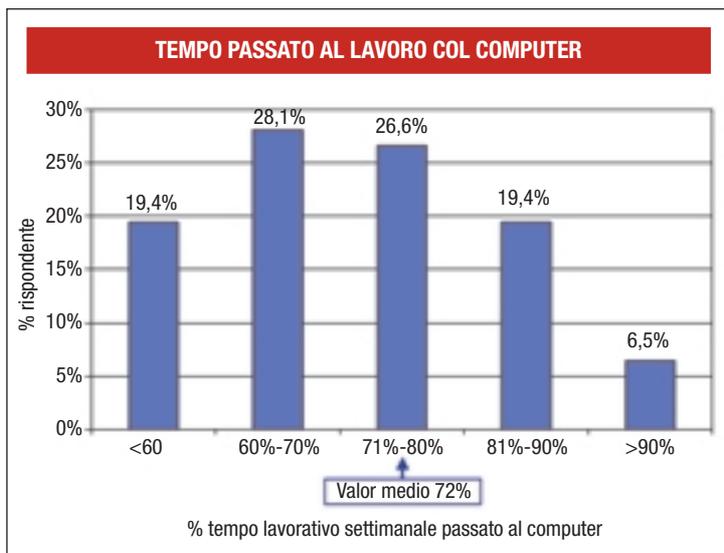


FIGURA 2

Percentuale del tempo di lavoro in cui si utilizza il computer

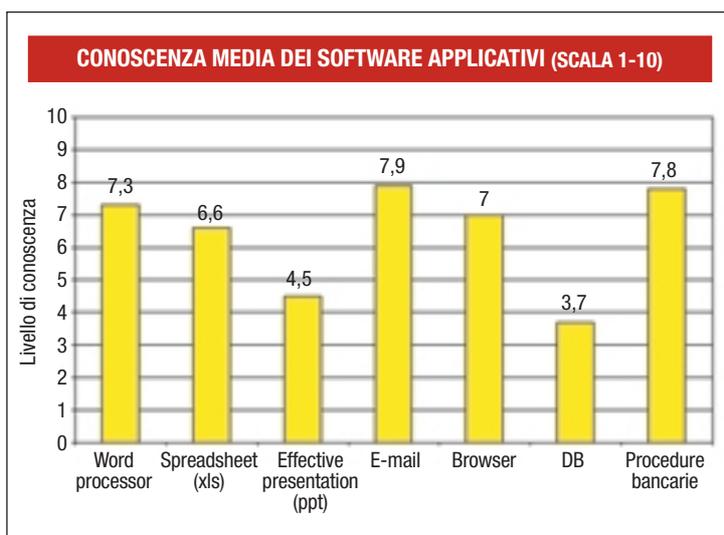


FIGURA 3

Conoscenza degli strumenti informatici da parte dei dipendenti bancari

tati complessivi della rilevazione svolta durante la nostra ricerca (Figura 2) si può constatare che almeno un quarto dei dipendenti bancari dichiara di passare al computer più dell'80% del proprio tempo lavorativo. Un livello di utilizzo del computer così elevato e così diffuso non è riscontrabile in altri settori economici, non appartenenti all'ambito della finanza. È legittimo pertanto domandarsi quale sia il livello delle conoscenze degli addetti che lavorano in questo comparto nei riguardi degli strumenti applicativi disponibili sul computer. La ricerca ha eviden-

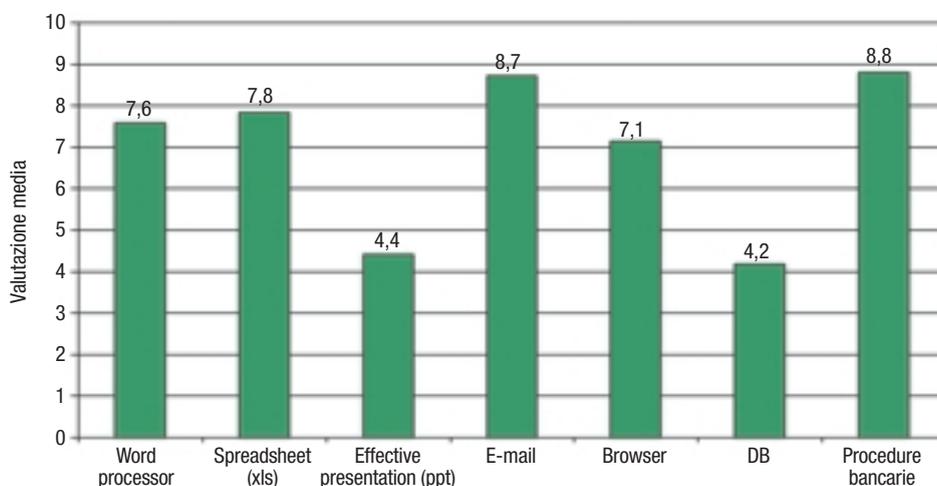
ziato un livello generalmente molto elevato di conoscenza degli strumenti (Figura 3): in particolare sono ben conosciute le procedure informatiche bancarie (i *legacy system* messi a disposizione dalla azienda per lo svolgimento delle operazioni bancarie). Subito dopo come livello di conoscenza sono citati la posta elettronica, i *word processor* e i *browser* (per la navigazione in Internet). Tra gli strumenti meno conosciuti sono annoverati gli strumenti di *effective presentation*, quali –per esempio- MS/Power Point, e il software di organizzazione e accesso ai Data Base. I livelli di conoscenza corrispondono al grado di utilizzo dei rispettivi software: ciò che si conosce meglio è ciò che usa più frequentemente, o ciò che si rivela più utile per la propria attività (Figura 4).

3. L'IMPRODUTTIVITÀ DERIVANTE DA PROBLEMI INFORMATICI

A tutti gli utenti che hanno partecipato alla ricerca è stato chiesto di indicare quanto tempo settimanale essi "perdevano" per problemi con i computer o le reti. Come si può osservare dalla figura 5 il tempo "perso" varia da poco meno di 10 min alla settimana ad oltre 40. Il valore medio pesato è di circa 37 min. Si tratta di un valore sensibilmente differente rispetto a quello misurato prendendo in esame gli utenti di aziende rappresentative di tutti i comparti economici in contesti informaticamente evoluti come la Scandinavia e gli USA (171 min alla settimana), valore che è stato sostanzialmente confermato anche nel nostro Paese [7].

Nel caso del settore bancario italiano dobbiamo però distinguere l'entità del tempo improduttivo in relazione alla tipologia degli utenti. I dirigenti denunciano una improduttività settimanale di 10 min, i quadri di 25 e gli impiegati di 45.

Il tempo improduttivo deriva da tre diverse cause: motivazioni tecniche (come guasti dell'hardware e malfunzionamenti del software) non conoscenza degli strumenti e aiuto prestato a colleghi in difficoltà. Mediamente il 30% del tempo "perso" al computer, come si può vedere dalla figura 6, potrebbe essere recuperato migliorando le conoscenze degli utenti: è questa la parte della "improduttività"

UTILITÀ MEDIA DEGLI APPLICATIVI SOFTWARE PER PROPRIA ATTIVITÀ (MEDIE; SCALA 1-10)

FIGURA 4

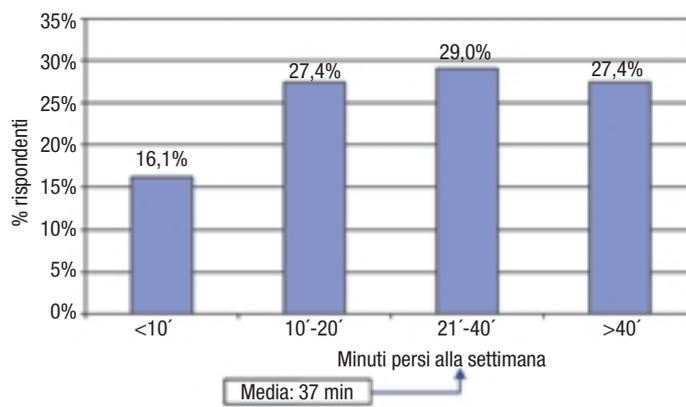
Gli strumenti informatici più utilizzati dai dipendenti bancari

che possiamo indicare con l'espressione "costo della ignoranza informatica".

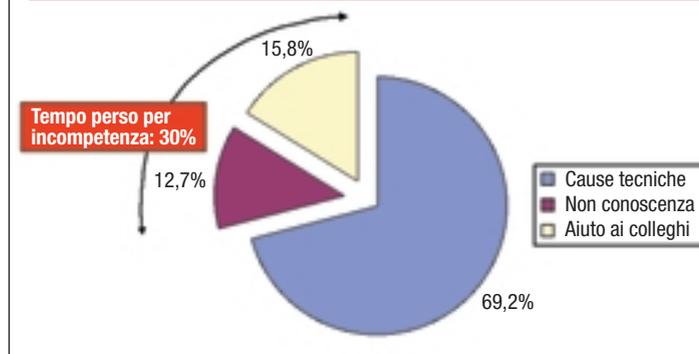
Possiamo ora procedere, con gli elementi raccolti durante la ricerca, a valorizzare l'improduttività causata dagli strumenti informatici nel settore bancario. Come si può vedere dalla tabella 2 il valore dipende sia dalla differente durata del tempo perso, sia dal costo aziendale del dipendente.

Il valore del tempo perso durante l'arco di un anno per problemi con i computer e le reti è di 494 € per i dirigenti, 883 per i quadri e 1.124 per gli impiegati. Conoscendo il numero degli addetti si può anche calcolare il costo complessivo della "improduttività" per l'intero settore bancario. Come si può vedere dalla tabella 3 si tratta di un valore di circa 352 milioni di euro all'anno. Una quota parte di tale valore (all'incirca il 30%) è imputabile alla "ignoranza informatica" in senso stretto. Nello specifico il costo dell'ignoranza ammonta, per ogni dirigente, a 148 € l'anno, per ogni quadro a 265 € e per ogni impiegato a 337 €.

L'impreparazione degli utenti costa al sistema bancario italiano circa 105 milioni di euro ogni anno, un importo non trascurabile che vale la pena di provare a ridurre con interventi di formazione, soprattutto considerando il fatto che sostenendo un onere di formazione una tantum, si può incidere su un costo ricorrente come quello derivante dalla ignoranza informatica.

MINUTI PERSI ALLA SETTIMANA: DISTRIBUZIONE DI FREQUENZA

FIGURA 5

Tempo perso alla settimana per problemi con il computer

CAUSE DELLA PERDITA DI TEMPO SETTIMANALE

FIGURA 6

Il tempo perso settimanalmente per problemi con il computer

Personale bancario	Costo medio aziendale (€)	Costo orario (€)	Tempo di lavoro settimanale (ore)	Tempo di uso del computer	Tempo di lavoro settimanale al computer (ore)	Tempo perso settimanalmente al computer (minuti)	% del tempo lavorativo perso settimanalmente al computer	Valore del tempo perso annualmente al computer per addetto (€)
Dirigenti	118.650	67,41	40,0	25%	10,00	10	0,4%	494
Quadri	84.750	48,15	40,0	50%	20,00	25	1,0%	883
Impiegati	56.500	34,24	37,7	80%	30,16	45	2,0%	1.124
Altre qualifiche	39.550	23,97	37,5	0%	-			

TABELLA 2

Il valore del tempo improduttivo ("perso") nell'utilizzo della tecnologia

Qualifiche	N° addetti settore bancario	Valore del tempo perso annualmente al computer per addetto (€)	Costo per il settore dei problemi con il computer (€)	% tempo perso per incompetenza	Costo dell'ignoranza informatica per addetto (€)	Costo dell'ignoranza informatica per il settore (€)
Dirigenti	7.140	494	3.529.838	30%	148	1.058.951
Quadri	115.940	883	102.353.281	30%	265	30.705.984
Impiegati	216.580	1124	243.437.069	30%	337	73.031.121
Altre qualifiche	340	0	-		-	-
Totale	340.000		349.320.188			104.796.056

TABELLA 3

Il "costo" dell'ignoranza informatica nel settore bancario

4. FORMAZIONE INFORMATICA E PRODUTTIVITÀ NELLE BANCHE

Il "costo della ignoranza informatica" non è dunque particolarmente elevato in questo settore. I ricercatori, quindi, si sono posti un ulteriore quesito: una migliore preparazione nell'impiego degli strumenti informatici potrebbe determinare significativi incrementi di efficienza nello svolgimento del lavoro degli addetti di questo settore, e per conseguenza un miglioramento dell'efficienza complessiva dell'intero sistema bancario nel suo insieme?

L'ipotesi che i ricercatori hanno formulato riguarda la sussistenza di una relazione positiva di causa-effetto tra preparazione informatica e capacità di usare meglio gli strumenti informatici. Per conseguenza ad una maggiore conoscenza degli strumenti informatici dovrebbe corrispondere una più elevata produttività individuale e aziendale. Le ipotesi alla base della ricerca potrebbero es-

sere formulate in modo più esplicito come segue:

a. la formazione informatica può aumentare il livello delle conoscenze informatiche e contribuire pertanto alla riduzione del cosiddetto "costo dell'ignoranza informatica" nel settore bancario;

b. l'aumento delle conoscenze informatiche, determinato dalla formazione, può aumentare l'efficienza delle persone che lavorano al computer (riduzione del tempo di svolgimento del lavoro) aumentando la produttività aziendale e -per conseguenza- l'efficienza del settore nel suo complesso.

I ricercatori della SDA Bocconi e di AICA hanno già verificato in due precedenti cicli di ricerca (il primo riguardante un campione di impiegati e manager appartenenti ad un insieme di aziende italiane generiche [7] e il secondo un campione del personale del sistema sanitario italiano [4]) che sussiste effettivamente una relazione di causa effetto tra la



formazione informatica e l'aumento di produttività di chi lavora al computer e che l'incremento delle conoscenze informatiche riduce anche il tempo perso al computer a causa della ignoranza informatica. In entrambi i casi si è riscontrato un duplice effetto:

1. la frequenza ad un corso di formazione ECDL ha prodotto un incremento delle conoscenze iniziali di circa 20 punti su una scala da 0 a 100 (dove 100 indica la conoscenza perfetta di uno strumento informatico);
2. a seguito della formazione si è verificata una riduzione dei tempi di esecuzione del lavoro, compresa tra il 10 e il 15% del tempo inizialmente impiegato.

5. GLI EFFETTI DELLA FORMAZIONE INFORMATICA SUL SETTORE BANCARIO ITALIANO

Questa volta si è cercato di ripetere la verifica delle ipotesi della ricerca con riferimento al personale bancario. Due sono state le attività di indagine attivate al riguardo. In primo luogo è stato fatto un esperimento di tipo empirico tendente a misurare i "ritorni" della formazione informatica. A tal fine è stato condotto un esperimento di formazione sul personale di una banca italiana di piccola dimensione (Banca di Credito Cooperativo di Lesmo). È stata formata una classe costituita da una decina di impiegati e quadri, rappresentativa su piccola scala dell'universo degli addetti del settore bancario, ed è stato erogato a questi soggetti un corso preparatorio al superamento della prova per la patente europea di guida del computer (ECDL).

L'analisi degli effetti della formazione sull'uso degli strumenti informatici è stata effettuata misurando la capacità dei soggetti nella risoluzione di problemi applicativi prima e dopo la frequenza al corso e il conseguimento della certificazione ECDL. Come è noto la patente ECDL riguarda la capacità di utilizzare in modo appropriato il personal computer come strumento di lavoro individuale. Si è trattato quindi di misurare i progressi dei partecipanti ai corsi nella capacità di saper lavorare con i seguenti strumenti:

- il sistema operativo e le *utilities di gestione degli archivi*;
- il *word processor*;

- il foglio di lavoro elettronico (*spreadsheet*);
- gli strumenti di *effective presentation*;
- internet* e la posta elettronica (*e-mail*).

La scelta degli strumenti per i quali sono stati svolti i corsi di formazione corrisponde alle esigenze di conoscenza degli strumenti rilevate presso le istituzioni bancarie.

I corsi ECDL prevedono dei test per verificare il livello di conoscenza di uno strumento software. Tale livello è misurato da un punteggio su una scala che va da 0 (nessuna conoscenza) a 100 (conoscenza completa dello strumento). Nel nostro caso un medesimo test, costituito da 37 domande riguardanti la soluzione di problemi pratici nei 5 ambienti precedentemente citati è stato affrontato dai partecipanti prima e dopo il corso. Per ogni soggetto e per ogni ambiente sono state misurate due parametri:

1. il punteggio conseguito nel test, che misura il livello di padronanza della tematica;
2. il tempo impiegato nella soluzione del problema prospettato dal test.

Una seconda attività di ricerca ha riguardato la raccolta di dati presso la direzione delle banche sulla formazione nelle banche medesime e sulle sue motivazioni⁴. Infine si è chiesto a chi aveva partecipato a iniziative di formazione di indicare gli effetti prodotti dalla formazione secondo loro⁵.

Per quanto riguarda l'esperimento empirico (formazione con i corsi ECDL), come era prevedibile i partecipanti a questo test hanno fatto registrare un livello di conoscenza iniziale degli strumenti molto elevato. Prima del corso solo il 14% dei partecipanti al test non raggiungeva, come livello di conoscenza, il 50% del valore massimo. Tutti gli altri potevano considerarsi già "molto alfabetizzati", anzi un 57% ha conseguito nel test iniziale un

⁴ Sono stati raccolti tramite intervista telefonica o contatti diretti 88 questionari compilati dai responsabili dei sistemi informativi e dai responsabili del personale di 86 banche italiane con 103.000 dipendenti complessivamente.

⁵ Tramite appositi questionari compilati da 169 utenti, di oltre una sessantina di banche, si è cercato di rilevare il fabbisogno di formazione informatica avvertito nel settore bancario, la formazione effettuata effettivamente e le modalità con cui essa è avvenuta.

punteggio superiore ai 90 punti su un massimo di 100 (Figura 7); mentre il 28% ha mostrato un livello di conoscenza buono, cioè compreso tra il 50 e il 90% del massimo delle conoscenze raggiungibili. Questo risultato è coerente con quanto rilevato precedentemente. Non è pensabile che persone che trascorrono circa il 72% del proprio tempo lavorativo davanti ad un computer (Figura 2) possano essere impreparate nell'uso del software di utilizzo personale. A tal proposito si osserva che la conoscenza media degli strumenti per l'intero gruppo è risultata pari a 77/100. Alla fine del corso sono stati ripetuti i test (naturalmente non i medesimi, ma prove analoghe e comparabili) e il livello di conoscenza media è risultato pari a 93/100 con un

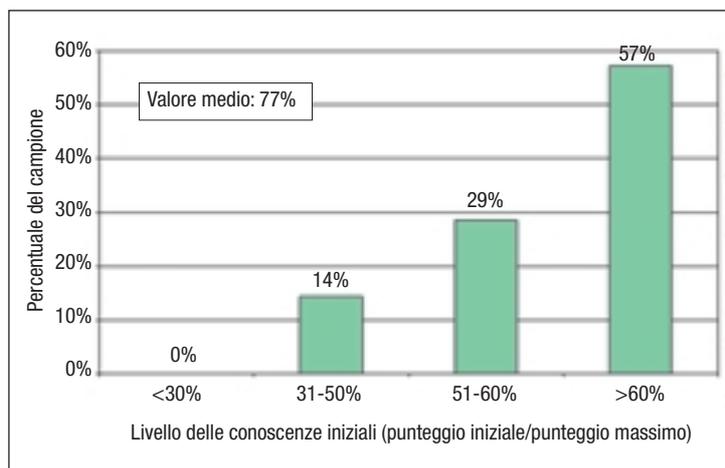


FIGURA 7
Articolazione del campione per livelli di conoscenza iniziale (scala delle conoscenze: 0-100)

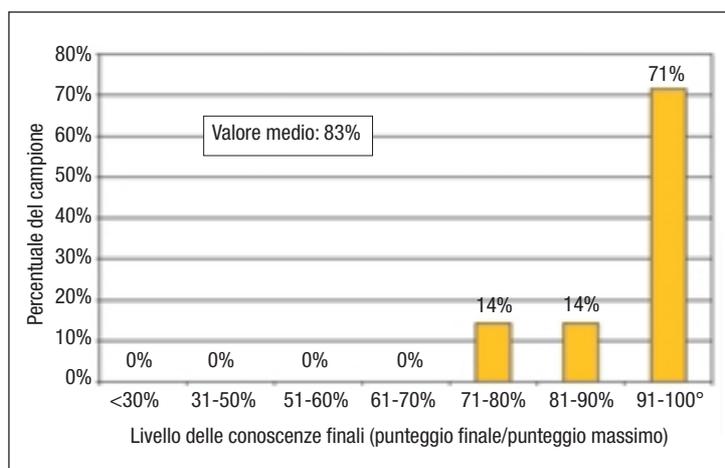


FIGURA 8
Livello delle conoscenze finali del campione (scala delle conoscenze: 0-100)

incremento di ben 27 punti su una scala da 0 a 100 (Figura 8).

Si tratta di un risultato molto interessante soprattutto se confrontato con quanto ottenuto nelle due precedenti edizioni della ricerca sul costo della ignoranza informatica (Tabella 4). Mentre nelle precedenti ricerche il campione del test era rappresentato da soggetti con preparazione media inferiore al 50% delle conoscenze massime, in questo caso l'esperimento ha coinvolto un insieme di soggetti già acculturati dal punto di vista informatico. Nonostante ciò si sono riscontrati significativi incrementi del livello delle conoscenze iniziali a conferma del fatto che la formazione produce comunque un risultato positivo. L'incremento di conoscenze generato è ovviamente dipendente dal livello di partenza. In realtà, se analizziamo l'incremento di conoscenza che si è verificato come conseguenza del corso frequentato, notiamo che la maggior parte dei partecipanti al test ha conseguito un miglioramento compreso tra 1 e 10 punti percentuali, mentre una porzione significativa del campione ha superato i 20 e addirittura i 30 punti di miglioramento (Figura 9). Ciò significa che l'effetto dei corsi di formazione non è omogeneo ma, come è naturale, dipende dal livello iniziale delle conoscenze, dalla predisposizione dei soggetti all'apprendimento, dalle capacità individuali, e così via. Comunque, pur con tutti i limiti di un esperimento condotto su un campione ristretto di soggetti, rimane confermato quanto già riscontrato nelle precedenti fasi della ricerca: la formazione informatica produce immediatamente un miglioramento sensibile nella capacità di risolvere

Settori	Livello conoscenze iniziali	Livello conoscenze finali	Variazione
Aziende Italiane	40	60	+20
Settore Sanitario	25	43	+18
Settore Bancario	77	93	+16

TABELLA 4
Gli effetti della formazione sul livello delle conoscenze informatiche

i problemi che si presentano nell'utilizzo del computer.

6. GLI EFFETTI DELLA FORMAZIONE SUL TEMPO DI SVOLGIMENTO DEL LAVORO CON IL COMPUTER

L'aumento di conoscenze informatiche, determinato dall'aver frequentato il corso ECDL, può produrre positive conseguenze sulle modalità di svolgimento del lavoro con il computer. Come osservato da Camussone [7] il personale "formato":

1. fa meglio le attività che precedentemente gli creavano difficoltà;
 2. svolge lavori che prima non sapeva svolgere;
 3. impiega meno tempo (grazie alle nozioni apprese) a svolgere il proprio lavoro al computer;
 4. perde meno tempo davanti al computer per problemi derivanti dalla sua ignoranza informatica.
- Anche nel caso dell'esperimento effettuato con la BCC di Lesmo sorge la curiosità di sapere se i ritorni della formazione siano -almeno in parte- misurabili e a quanto ammontino. Pur riconoscendo che alcuni dei vantaggi sopra citati sono di natura qualitativa, e quindi difficilmente quantificabili in termini monetari, seguiremo la strada tracciata da coloro che si sono già cimentati in questa impresa [9]. Nel nostro caso è stato rilevato il tempo che i partecipanti hanno impiegato per risol-

vere i problemi previsti dai test prima e dopo aver frequentato il corso ECDL (Figura 10). Come si può constatare per quattro moduli su cinque si è effettivamente riscontrata una diminuzione del tempo, mentre nel caso del modulo su Internet (Browser ed e-mail) i tempi si sono allungati. Quest'ultimo fatto può essere spiegato con un miglioramento nella capacità di usare meglio la rete e quindi con un corrispondente aumento del tempo dedicato alla ricerca di informazioni, che risulta più efficace e soddisfacente rispetto a prima (fenomeno noto come "compiacimento nell'uso dello strumento"). Anche per la posta

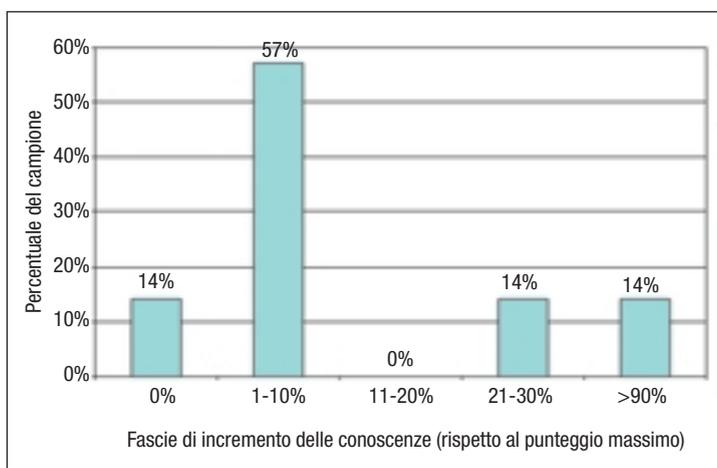


FIGURA 9
Miglioramento delle conoscenze dopo il corso ECDL

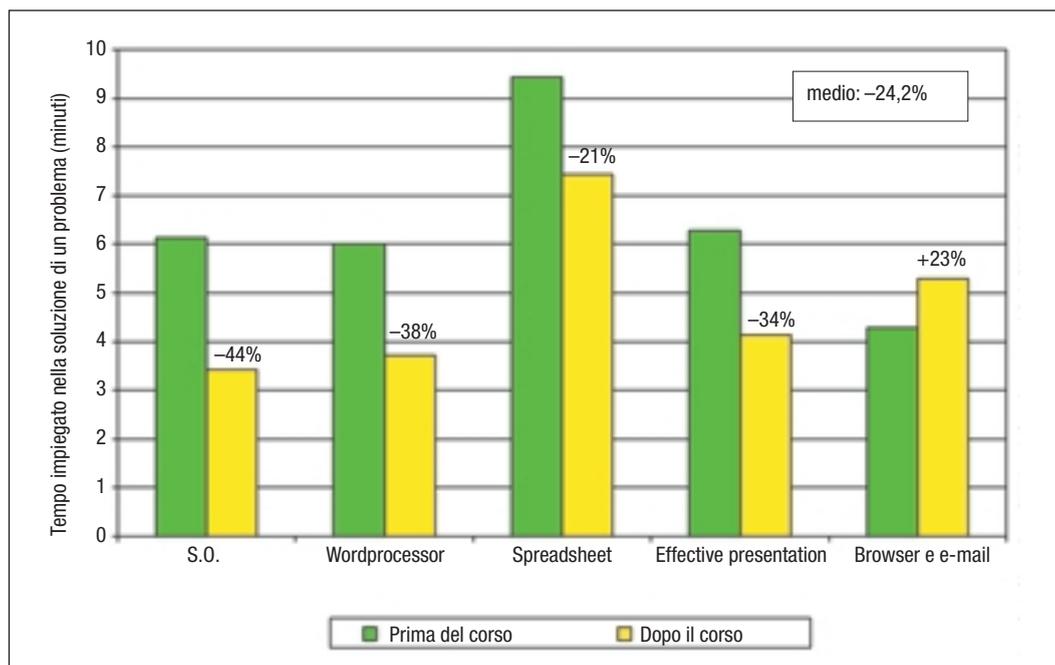


FIGURA 10
Tempo impiegato nella soluzione di problemi prima e dopo il corso

elettronica può verificarsi un fenomeno di aumento dell'uso, se se ne apprezzano le possibilità e le potenzialità (liste personalizzate di distribuzione di messaggi, classificazione della posta in entrata e uscita, uso di filtri anti-spamming ecc.).

Nel complesso, comunque, risulta una riduzione nei tempi di svolgimento dei test di circa il 24%. Ma ciò che sorprende è che la riduzione dei tempi si sia verificata negli ambienti in cui maggiore era la conoscenza iniziale, come per esempio l'ambiente del sistema operativo e quello della elaborazione dei testi.

Questo fatto può essere spiegato in questi termini: quando il livello di conoscenza è già elevato, probabilmente il corso di formazione rimuove piccole sacche di ignoranza e ciò determina la percezione di nuovi modi di utilizzo degli strumenti (scorciatoie, o prassi prima non note) che abbreviano i tempi di lavoro in modo significativo. Nel caso di uso di strumenti più complessi, come i fogli di lavoro elettronici, un corso può determinare certamente un miglioramento immediato nel modo di lavorare, ma i risparmi di tempo sono minori perché probabilmente è necessario un periodo di pratica più lungo, per metabolizzare procedure di uso dello strumento più avanzate.

Comunque, se dalle misure empiriche sul campione di individui che ha affrontato i test

si è rilevata una riduzione del 24,2% nel tempo di svolgimento delle varie attività sul computer per effetto del corso di formazione frequentato, possiamo domandarci quali effetti si determinerebbero se questo risultato si potesse estrapolare alla realtà lavorativa di tutti gli addetti bancari; in questo modo potremmo valutare gli effetti economici della formazione, ovvero il cosiddetto "ritorno della formazione".

Come si può vedere dalla tabella 5 gli addetti del settore bancario denunciano una percentuale di tempo lavorativo speso sul computer assai diverso. Pertanto un aumento della produttività del lavoro con il computer del 24,2% incide in modo differente sulla produttività complessiva delle varie categorie del personale. Per esempio a livello dei quadri si verificherebbe un aumento di produttività settimanale di 4,84 h, mentre per gli impiegati tale valore salirebbe a 7,26 h. Si può poi risalire da questi valori all'aumento di produttività complessivo nell'arco di un anno e al valore economico corrispondente che deve essere calcolato tenendo conto del differente livello di costo aziendale delle varie categorie degli addetti del settore bancario. L'aumento di produttività individuale risulta pertanto compresa tra i 7.178 €, nel caso di un dirigente, e i quasi 11.000 € nel caso di un impiegato (Tabella 5).

Categorie del personale	N° addetti	Tempo di lavoro settimanale (ore)	Tempo di uso settimanale del computer (in ore e in percentuale del tempo lavorativo)	Aumento settimanale della produttività (+24,2%) in ore	Aumento annuale di produttività (giorni)	Costo annuale pro capite (€)	Costo giornaliero (€)	Valore annuale dell'aumento di produttività pro capite (€)	Valore annuale dell'aumento di produttività della categoria (€)
Dirigenti	4.760	40	10 25%	2,42	13,3	118.650	539	7.178	34.168.827
Quadri	103.360	40	20 50%	4,84	26,6	84.750	835	10.255	1.059.930.960
Impiegati	229.840	37,5	30 80%	7,26	42,6	56.500	257	10.938	2.514.081.856
Altre qualifiche	2.040	37,5	- 0%	-	-	39.550	180	-	-

TABELLA 5

Calcolo del miglioramento della produttività nel settore della intermediazione finanziaria dopo i corsi ECDL

Si tratta di valori certamente significativi che moltiplicati per tutti gli addetti comporterebbero un aumento di produttività complessiva, per l'intero settore bancario, di circa 3.6 miliardi di euro l'anno. I benefici tangibili della formazione informatica, oltre a consistere in un aumento di produttività nel lavoro svolto con il computer, riguardano anche la riduzione del tempo perso al computer per ignoranza sull'uso dello strumento, fenomeno questo che in letteratura [7] è indicato con l'espressione "costo dell'ignoranza informatica". Nel caso del settore della intermediazione finanziaria tale perdita di tempo è stata quantificata in valori che variano tra i 148 € l'anno per ogni dirigente e i 337 € per ogni impiegato (vedi Tabella 3).

Tali valori possono ridursi per effetto della formazione informatica, anche se non si può pensare che tutti i dipendenti raggiungano una perfetta conoscenza degli strumenti informatici, tale da evitare qualsiasi effetto derivante dalla "ignoranza informatica". Dal momento che i test effettuati sul campione dei dipendenti bancari della BCC di Lesmo hanno mostrato un incremento medio del livello delle conoscenze informatiche del 16% si può ragionevolmente ritenere che di una percentuale equivalente si riducano le occasioni di perdita di tempo per effetto della ignoranza informatica. Si può quindi calcolare a che cosa equivale, in termini economici, una riduzione del 16% del tempo perso per ignoranza informatica dalle diverse categorie di addetti del settore bancario e parabancario (Tabella 6).

Il ritorno economico della formazione informatica, per quanto concerne la riduzione del tempo perso per ignoranza, non è comparabile con quanto si ottiene come miglioramento nella produttività del lavoro sul computer. Nel caso dei dirigenti si otterrebbe un beneficio di circa 24 € l'anno e nel caso degli impiegati, che passano più tempo sul computer, si avrebbe un beneficio maggiore, anche se non si arriverebbe comunque ai 60 € l'anno. La ricerca empirica sui ritorni della formazione informatica basata sul corso ECDL porta quindi alla seguente conclusione: *nel caso del settore bancario gli addetti sono già abbastanza "alfabetizzati" dal punto di vista informatico, e quindi i benefici maggiori della formazione non derivano dalla riduzione della "ignoranza informatica", quanto piuttosto dall'aumento di produttività nell'uso del computer, che consegue ad un maggior livello di conoscenza degli strumenti software.*

7. LA FORMAZIONE INFORMATICA NELLE BANCHE ITALIANE

Le banche italiane sono coscienti dell'importanza della formazione informatica, e investono in questo strumento di crescita delle competenze del proprio personale? Per rispondere a questo quesito i ricercatori di AICA e della SDA Bocconi, mediante questionari strutturati, hanno raccolto le opinioni al riguardo di molti responsabili della formazione nelle banche e di alcuni responsabili di sistemi informativi. Inoltre si è cercato di sentire, sempre tra-

Categorie del personale	N° addetti	Valore annuale del tempo perso per impreparazione informatica pro capite (€)	Percentuale del tempo perso recuperabile con la formazione	Valore annuale del tempo perso per impreparazione informatica pro capite recuperabile con la formazione (€)	Valore annuale del tempo perso per impreparazione informatica recuperabile con la formazione nell'intero settore (€)
Dirigenti	4.760	148	16%	24	112.717
Quadri	103.360	265	16%	42	4.382.464
Impiegati	229.840	337	16%	54	12.392.973
Altre qualifiche	2.040	-		-	-
Totale	340.000				16.888.154

TABELLA 6

Calcolo della riduzione del costo della impreparazione informatica nel settore bancario dopo i corsi ECDL

FABBISOGNO DI FORMAZIONE AVVERTITO PER APPLICATIVO (SCALA 1-10)

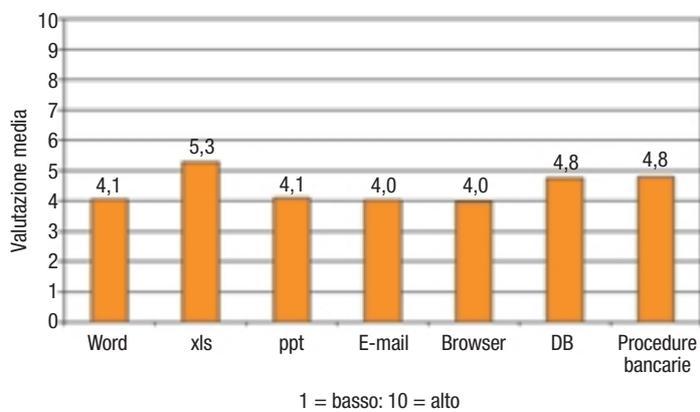


FIGURA 11

Le esigenze formative avvertite dall'utenza

mite un altro questionario strutturato, cosa ne pensano a proposito della formazione informatica i soggetti più interessati, cioè gli utilizzatori delle infrastrutture informatiche predisposte dalle aziende⁶.

Ne è emerso un quadro interessante. Gli utenti hanno segnalato un fabbisogno di formazione non particolarmente elevato, posizionandosi come valore tra il 4 e il 5 su una scala da 0 a 10 (Figura 11). Ciò può essere giustificato da un livello di conoscenza già acquisita piuttosto elevato, dal momento che l'uso del personal computer è abituale e generalizzato a tutti i livelli della banca.

Il contratto di lavoro dei bancari prevede annualmente diverse giornate di formazione per ciascun addetto ed in effetti le banche tengono fede a questi accordi. Mediamente ogni dipendente partecipa a 7,06 giorni di formazione ogni anno, anche se la maggior parte dei dipendenti non arriva a 5 giorni di istruzione l'anno. Sono comunque pochissimi coloro che non fanno formazione per nulla (solo il 4%) mentre il 13% partecipa a iniziative di formazione per più di 11 giorni l'anno. Tuttavia, se si indaga sui contenuti della formazione, si scopre che il tempo destinato ai

⁶ Si veda quanto illustrato nelle note precedenti. Per la descrizione completa del metodo della ricerca, del campo di rilevazione e dei questionari utilizzati si rimanda alla parte terza del rapporto di ricerca: Camusso P.F., Occhini G., Santececca D.: Competenze digitali e produttività nel settore bancario italiano, McGrawHill 2006.

Azienda	78%
Dipendente	4%
Entrambi	17%
Non precisato	1%

TABELLA 7

Chi paga la formazione ai dipendenti bancari

corsi di argomento informatico è marginale. Circa due terzi del personale bancario non fa neppure un giorno di formazione informatica all'anno e il 18% si limita a frequentare un giorno di formazione all'anno: in media per ogni addetto si è rilevato un valore di 0,78 giorni l'anno di formazione informatica.

Si tratta di dati preoccupanti soprattutto dopo quanto si è appena visto sul ritorno della formazione informatica nel settore bancario. Evidentemente le banche non sono edotte circa i potenziali ritorni di una formazione di questo tipo erogata al proprio personale. Oppure vi sono lacune conoscitive più preoccupanti da colmare su cui le banche dirigono le risorse disponibili.

Molto interessante è l'analisi delle risposte al quesito: *chi sostiene i costi della formazione informatica degli addetti del settore bancario?* La sorpresa deriva dal fatto che i dipendenti per il 21% dei casi sostengono, se non tutto, almeno una parte dei costi della propria formazione (Tabella 7). Questo fatto lascia intendere che molti dipendenti frequentano corsi di loro iniziativa e al di fuori delle strutture bancarie.

8. GLI EFFETTI DELLA FORMAZIONE SECONDO GLI UTENTI

Infine, la ricerca ha cercato di rilevare anche il punto di vista degli utenti sugli effetti della formazione. Tramite il questionario già citato, gli utenti sono stati invitati a indicare gli effetti percepiti dopo aver frequentato un corso di formazione sugli strumenti informatici.

Le risposte raccolte hanno messo in evidenza la prevalenza di due differenti tipologie di conseguenze (Figura 12). La prima riguarda il miglioramento della modalità di svolgimento del proprio lavoro. A tal proposito so-

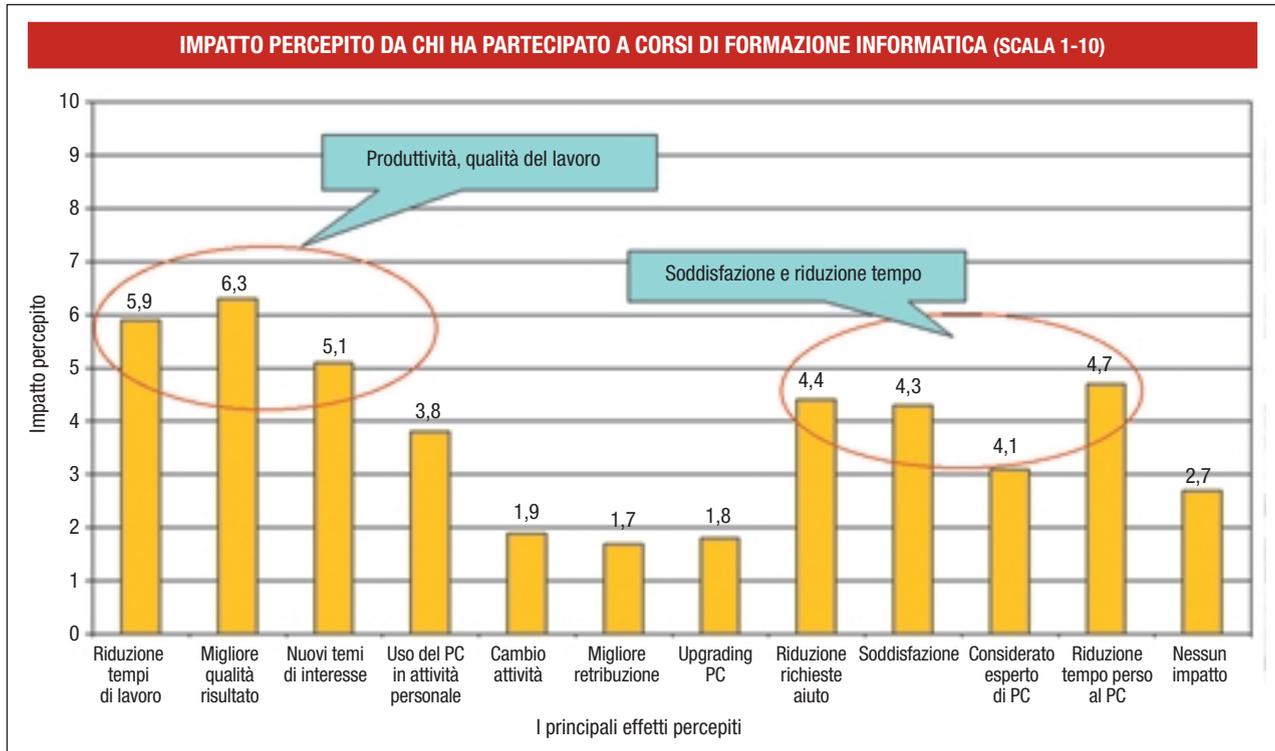


FIGURA 12

Gli effetti della formazione secondo gli utenti

no stati indicati i seguenti effetti percepiti:

- un miglioramento della qualità dei risultati del proprio lavoro;

- la riduzione del tempo di lavoro;

- la possibilità di svolgere compiti nuovi e più interessanti grazie alla padronanza dell'ICT.

Il secondo insieme di conseguenze percepite si riferisce alla riduzione del senso di frustrazione che precedentemente provavano gli intervistati quando dovevano lavorare con il computer. A tal proposito sono stati indicati i seguenti effetti derivanti dalla formazione informatica:

- una riduzione del tempo improduttivo, ovvero "perso" al computer,

- la maggior soddisfazione nello svolgimento del proprio lavoro con il computer,

- la riduzione della dipendenza da colleghi più esperti in caso di difficoltà.

Sulla base dei test empirici effettuati e delle opinioni raccolte ci sembra di poter concludere che la formazione informatica determina in prima istanza positivi effetti sulla produttività dei dipendenti bancari. Ma non è solo uno strumento per accrescere l'efficienza aziendale. Anche gli utenti ne sentono la necessità e ne apprezzano, a livello persona-

le, gli effetti motivazionali sulle modalità di svolgimento del proprio lavoro.

Bibliografia

- [1] ABI - CIPA: *Costi e benefici degli investimenti informatici in banca: metodi di analisi e di valutazione per l'ottimizzazione della spesa informatica*, Bancaria. 1996.
- [2] Banca d'Italia: *Relazione Annuale per il 2004*. Roma, 31 maggio 2005.
- [3] Beaujean M., Reiche D., Roxburgh C.: *How Europe's banks can win in tougher times*. In The McKinsey Quarterly, June, 2005.
- [4] Borgonovi, Camussone, Occhini: *Il costo dell'ignoranza nella sanità*. ETAS 2004
- [5] Brynjolfsson E., Hitt L.: *Productivity, profit and consumer welfare: the different measure of IT's value*. MIS Quarterly, 1997.
- [6] Brynjolfsson E., Hitt L.M.: *Computing productivity: firm-level evidence*. MIT Sloan working paper June 2003.
- [7] Camussone, Occhini: *Il costo dell'ignoranza nella società dell'informazione*. ETAS 2003.
- [8] CIPA-ABI: *Rilevazione dello stato dell'automazione del sistema creditizio*. 2004-2005.

- [9] Danziger J.N., Jennings J.A., Park S.C.: *ICT Training; Center for research on ICT and Organizations*. University of California Irvine, 1999.
- [10] Danziger J.N., Wang Y.C.: *Enhancing end users' ICT skills in the new economy; Center for research on ICT and Organizations*. University of California Irvine, 2000.
- [11] EU Commission: *ICTs and Social Capital in the Knowledge Society*. Report on a Joint DG JRC/DG EMPL Workshop (IPTS, Sevilla, 3-4/11/2003).
- [12] EU Commission: *Information Society jobs - quality for change - Exploiting the Information Society's contribution to managing change and enhancing quality in employment*. Commission Staff Working Document; SEC, 2002, p. 372.
- [13] EU Commission: eEurope Action Plan. *An Information Society for all, COM*, 2002.
- [14] EU Commission: Strategies for jobs. In *the Information Society, COM*, 2000, p. 48.
- [15] Huizenga E.: *Innovation Management in the Ict Sector: How Frontrunners Stay Ahead*. Edward Elgar Publishing, 2004.
- [16] Johnson H.: *Technical support cost for dual-platform desktops: managed diversity*. Research Note, Gartner, 1995.
- [17] Khosrowpour M. (edited by): *Issues and Trends of Information Technology Management in Contemporary Organizations*. Idea Group Inc., 2002.
- [18] Lucas H.C.,JR: *Information Technology and the Productivity Paradox: Assessing the value of investing in IT*. Oxford University Press, 1999.
- [19] Mitchell W.J., Inouye A.S., Blumenthal M.S. (editors): *Beyond productivity*. Information Technology, Innovation and Creativity, The National Academic Press, 2003.
- [20] Mata F. J., Fuers W.L.t, Barney J.B.: Information Technology and Sustained Competitive Advantage. *A Resource-Based Analysis, MIS Quarterly*, Vol. 19, n. 4, December, 1995.
- [21] Nielsen B. (Cap Gemini-Ernst Young): *Documentation of hidden cost in Norwegian workforce*. ECDL-CEO meeting Cyprus, 2002.
- [22] Strassmann P.A.: *Information Productivity*. Assessing the Information Management Costs of US Industrial Corporations, Strassmann Inc., 1999.

PIER FRANCO CAMUSSONE è professore ordinario di "Organizzazione e sistemi informativi" presso l'Università di Trento. Direttore del corso di laurea magistrale in "Net Economy" della facoltà di Economia dell'Università di Trento. Docente di "Information Technology e nuovi modelli di business" presso l'Università Bocconi. Direttore dell'Area Sistemi Informativi della SDA Bocconi dal 1983 al 2003. Membro di comitati scientifici di diverse riviste (tra cui Economia e Management, Mondo digitale). Autore di numerosi libri e pubblicazioni riguardanti i sistemi informativi e l'impiego delle nuove tecnologie informatiche in chiave strategica ed organizzativa. E-mail: pfcamussone@unibocconi.it



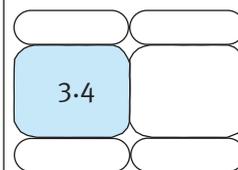
FRONTIERE DELLA RICERCA

DNA COMPUTING

IL CALCOLATORE IN PROVETTA

Il *DNA Computing* è un modello di calcolo proposto dodici anni fa da Leonard Adleman. In esso le molecole di DNA codificano sequenze di simboli e tipici processi biotecnologici realizzano algoritmi che elaborano tali sequenze. In questo articolo si presentano le idee fondamentali e i presupposti dei calcoli effettuati tramite il DNA. Quindi, si descrivono i principi dell'esperimento di Adleman e si indicano gli sviluppi, le tendenze e le attuali frontiere della ricerca nel settore.

Vincenzo Manca



1. INTRODUZIONE

L'idea iniziale del *DNA Computing* è molto semplice: "calcolare" significa passare da dati iniziali a risultati finali che soddisfino certe condizioni risolutive, ma dati e risultati sono sempre esprimibili con "parole" in un qualche linguaggio di rappresentazione dell'universo in cui si opera. Le molecole di DNA si possono assimilare a "parole doppie", costruite a partire da quattro simboli: A, T, C, G che stanno per le iniziali delle basi azotate: *Adenina, Timina, Citosina e Guanina*. Quindi, calcolare con il DNA significa sviluppare calcoli su "parole" di DNA, ovvero "stringhe" costruite sulle quattro lettere A, T, C, G. Questa metafora è stata tradotta in realtà da un esperimento ideato da Leonard Adleman [1] in cui, manipolando opportunamente una popolazione iniziale di molecole DNA che codificavano un grafo, si otteneva una popolazione finale composta da molecole di un solo tipo, che codificava la soluzione di un particolare problema definito sul grafo di partenza. Su una molecola di DNA si può scrivere l'informazione desiderata, non appena si definisca un

criterio di codifica. Per esempio, usiamo due lettere consecutive per indicare una cifra decimale secondo lo schema: 0 = AA, 1 = AT, 2 = TT, 3 = CC, 4 = CG, 5 = GG, 6 = AC, 7 = AG, 8 = TC, 9 = TG. In questo modo il numero 1270 viene codificato dalla parola ATTTAGAA. Per sommare tramite DNA i numeri 1270 e 27 basta quindi partire da una provetta in cui vi siano inizialmente molecole ATTTAGAA e TTAG, in una qualche concentrazione (diciamo una pico-mole di ciascuna in soluzione acquosa), e applicare dei procedimenti di manipolazione biomolecolare, di cui diremo più avanti, in modo che alla fine vi sia solo un tipo di molecola lunga otto bp (coppie di basi) del tipo ATTTTGAG, che corrisponde, secondo la nostra codifica, proprio alla somma 1297 dei due numeri 1270 e 27. Ovviamente tale metodo è di scarso interesse pratico, ma si presta bene a spiegare l'essenza di un qualsiasi algoritmo DNA: codificare i dati con molecole DNA poste in una provetta iniziale, applicare delle operazioni che trasformano ad ogni passo il contenuto della provetta e alla fine "leggere" il risultato in qualche tipo di molecola selezionata secondo un opportuno criterio di decodifica.

Un tale modello di calcolo è profondamente diverso dal classico modello della macchina di Turing. Infatti questa, secondo una delle sue formulazioni più comuni, elabora una struttura lineare, assimilabile ad un nastro suddiviso in caselle, in cui è posto un simbolo per casella. L'organo di controllo della macchina può alterare il contenuto di una casella e spostarsi a leggere nella casella contigua di destra o di sinistra. L'azione svolta dalla macchina, ad ogni passo, è determinata da un programma, in base allo stato della macchina e al simbolo letto. Un tale tipo di calcolo è essenzialmente sequenziale e "monogenico" per due aspetti cruciali. La struttura lineare che si elabora è unica e su di essa si interviene leggendo ed eventualmente alterando un simbolo alla volta. Inoltre, tutto il processo segue un programma centralizzato. Il paradigma sottostante al calcolo DNA è invece essenzialmente parallelo e "poligenico": la stessa sequenza è presente in centinaia di miliardi di copie e vari agenti di calcolo (enzimi) agiscono su di esse con operazioni globali (concatenazioni, tagli, appaiamenti, riconoscimento di porzioni), in gran parte indipendentemente l'uno dall'altro.

In una tale prospettiva, la classica differenza *software/hardware* risulta difficilmente riconducibile allo schema tipico della macchina a programma di von Neumann. In altre parole, per dare solo una prima idea iniziale, una "programmazione DNA" può assumere la forma dello pseudo-codice dato nel riquadro di p. 30, ove, in relazione ad un problema combinatorio, è descritta una procedura per manipolare delle molecole iniziali di DNA, eseguendo certe operazioni su provette. Il funzionamento di un tale paradigma prevede quindi di agire piuttosto che su una singola sequenza, su "popolazioni" di sequenze, elaborate in parallelo, e portando avanti miliardi di calcoli singoli. Ciascun calcolo esplora una possibilità risolutiva. Alla fine, se una soluzione viene determinata, questa viene selezionata secondo opportuni criteri e quindi "letta". Volendo fare un esempio, è come se, per realizzare un certo compito si costrissero tanti piccoli automi, di diversi tipi, ciascuno con una specifica funzionalità, e poi si mettessero in un ambiente dove questi incontrano varie occorrenze del problema da risolvere. Gli automi, in base alle loro caratteristiche, provano varie for-

me di assemblaggio e di coordinazione reciproca per aggredire "il nemico" da sconfiggere. Chi prima riesce nell'impresa "suona un campanello" e quindi esibisce la strategia vincente. In tal senso non si programma la soluzione, ma piuttosto dei comportamenti da cui la soluzione può emergere.

Negli attuali algoritmi DNA l'esecuzione di certe manipolazioni di base è per lo più devoluta all'operatore umano. Tuttavia, un aspetto sempre più emergente nella ricerca nel settore tende a sviluppare metodi in cui opportune strutture molecolari codificano programmi eseguiti da opportuni nano-agenti. In tal caso, si ha una sorta di distinzione tra *hardware* e *software*, anche se entrambi i ruoli sono giocati da opportune molecole. Del resto, semplificando alquanto, una tale differenza è quella che in natura distingue i due tipi di biopolimeri che realizzano i processi vitali fondamentali: i geni (opportune sequenze di DNA) codificano la costruzione delle proteine (opportune sequenze di amminoacidi) e queste, a loro volta, realizzano le funzioni primarie degli organismi. Capire l'algorithmica di vari processi naturali, ed elaborare nuovi algoritmi che possano "girare" su questa sorta di "bio-ware", è una sfida completamente nuova in cui l'informatica può giuocare, a vari livelli, un ruolo insostituibile, che impone ripensamenti e nuove elaborazioni concettuali: la scoperta di algoritmi su strutture di dati non convenzionali, l'analisi di codifiche, la simulazione e la gestione computazionale di esperimenti per la realizzazione *in vitro* di calcoli molecolari.

Nel resto dell'articolo presenteremo i principi e gli strumenti sottostanti all'esperimento di Adleman, indicando gli antefatti e le prospettive che questo esperimento ha dischiuso. Quindi, concluderemo con un breve cenno alle attuali tendenze e alle linee di frontiera del *DNA Computing*.

2. RICHIAMI SUL DNA

2.1. La struttura bilineare del DNA

Il DNA (*Acido Deossiribonucleico*) è, come noto, il componente costitutivo del materiale genetico degli organismi viventi. Le molecole di DNA sono sequenze di coppie di **nucleotidi**. Ogni nucleotide, a sua volta, risulta dalla sintesi di tre tipi di molecole: uno zucchero **Z**, un

gruppo fosforico **P** (un fosfato PO_4), e una base azotata **B**, che può essere di 4 tipi diversi. Lo zucchero chiamato **deossiribosio** ha 5 atomi di Carbonio usualmente numerati con $1'$, $2'$, $3'$, $4'$, $5'$, ed è ottenuto da un "pentosio" C_5 (H_2O) $_5$, detto Ribosio, eliminando un Ossigeno legato al Carbonio della posizione $2'$. Il Ribosio a sua volta è presente nelle molecole di RNA, che forniscono "copie di lavoro" di quelle di DNA. Le molecole Z del DNA sono usualmente allineate in due filamenti paralleli formando una sorta di "binario". In ciascun filamento ogni molecola Z è legata alle altre molecole Z dello stesso filamento per mezzo del fosfato PO_4 con un legame chimico forte detto "fosfodiesterico". Il complesso Z + B dicesi **nucleoside**. Il nucleotide è quindi ottenuto aggiungendo al nucleoside il gruppo fosforico, ottenendo P + Z + B, ove l'ossidrile OH in $5'$ è sostituito da un ossigeno di PO_4 , (Figura 1). In tale binario il ruolo di "traversine" è giuocato dalle coppie di basi appaiate con legami "deboli" a Idrogeno: A - T, T - A, C - G, G - C (tra A e T si stabiliscono due legami a Idrogeno, mentre tra C e G ve ne sono tre). Infatti, vale la regola di appiamento di Chargaff secondo cui una base A si può appaiare solo con una T e viceversa, mentre una base C si può appaiare solo con una G e viceversa. Nei filamenti vi è un orientamento intrinseco dovuto alla natura orientata degli zuccheri Z in quanto pentosi (un pentagono è una forma naturalmente orientata). Il legame di un nucleotide con il successivo avviene lungo la direzione $5' - 3'$ in un filamento mentre avviene nella direzione $3' - 5'$ nel filamento appaiato. La disposizione che segue indica lo schema di un "binario DNA" a quattro "traversine", in cui si evidenziano gli orientamenti opposti dei due filamenti appaiati.

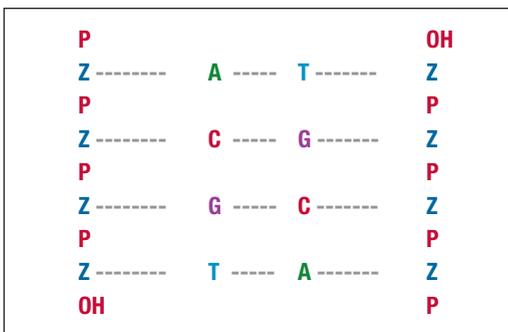


FIGURA 1

Un "binario DNA" a quattro "traversine

Dato che una lettera determina univocamente la sua appaiata, ciascuna "traversina del binario" è del tutto individuata da una sola lettera e quindi, da un punto di vista informativo, la struttura di sopra è completamente individuata dalla stringa **ACGT**.

2.2. La doppia elica DNA

Il doppio filamento di cui abbiamo detto ha uno svolgimento nello spazio secondo la tipica forma a doppia elica. Tale forma è intrinsecamente legata alla natura bilineare del DNA. Infatti, consideriamo in termini del tutto generali come si possa organizzare una doppia struttura di elementi che si susseguono su due file appaiate. Per visualizzare la cosa pensiamo ad un ballo a "pariglia" di dame e cavalieri, in cui ogni cavaliere si congiunge per mano ai danzatori della sua fila, mantenendo il viso sempre di fronte ad una stessa dama della fila opposta (e viceversa). I tre punti che determinano la disposizione di danza sono le posizioni della mano destra, della sinistra e del viso. Siccome le basi azotate che sono purine si appaiano a pirimidine (riquadro), per avere una completa analogia con l'appaiamento delle basi azotate, imponiamo che ogni cavaliere (Purina) si appaia ad una dama (Pirimidina), e che danzatori "alti" (a 3 legami H) e "bassi" (a 2 legami H) si appaiano con danzatori di altezza simile. In modo analogo, per ciascun nucleotide si determinano tre punti fondamentali nella realizzazione della forma bilineare: i due punti lungo la linea

Elementi strutturali del DNA

Basi azotate: Timina, Adenina, Citosina, Guanina: T, A, C, G (A, G Purine; T, C Pirimidine).

Deossiribosio = Z = $\text{C}_5\text{H}_{10}\text{O}_4$ (Pentosio con 5 atomi di Carbonio numerati: $1'$, $2'$, $3'$, $4'$, $5'$).

Molecola DNA = Sequenza di coppie di Nucleotidi a basi appaiate (A - T, C - -G).

Nucleoside = Z(B) = Deossiribosio Z + Base Azotata B (legata al Carbonio di Z in $1'$)

Z(B) = OH - ($5'$) - Z(B) - ($3'$) - OH (sono evidenziati i gruppi OH legati al Carbonio di Z in $3'$ e $5'$).

Nucleotide = Nucleoside + Fosfato = Z(B) + PO_4 = $\text{PO}_4 - (5') - Z(B) - (3') - \text{OH}$.

Concatenazione di nucleotidi ove -- indica il legame fosfodiesterico: $\text{PO}_4 - (5') - Z(B) - (3') - \text{OH} + \text{PO}_4 - (5') - Z(B) - (3') - \text{OH} = \text{PO}_4 - (5') - Z(B) - (3') - \text{PO}_4 - (5') - Z(B) - (3') - \text{OH}$.

Bilinearità = Appaiamento di due filamenti.

Complementarietà = T e A si appaiano con 2 legami H; C e G si appaiano con 3 legami H.

Antiparallelismo = Due filamenti appaiati sono orientati in senso opposto ($5' - 3'$) e ($3' - 5'$).

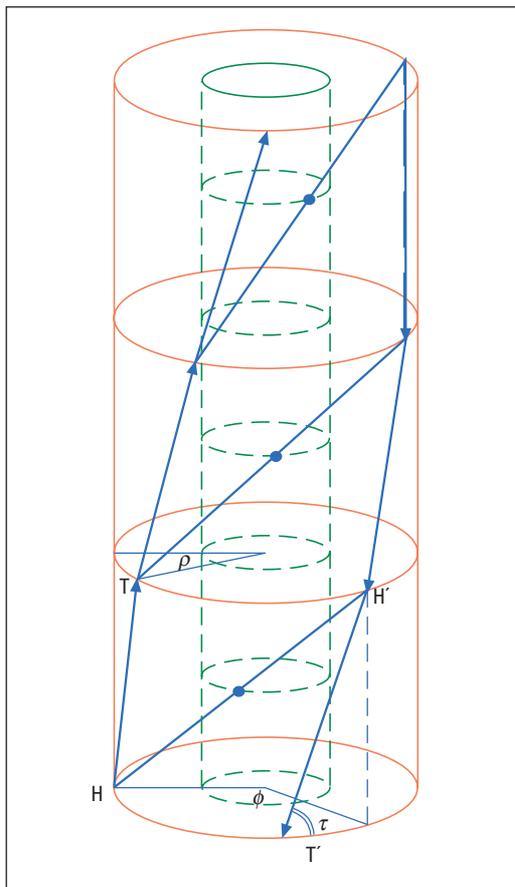


FIGURA 2
Lo sviluppo
a doppia elica
di una struttura
lineare doppia

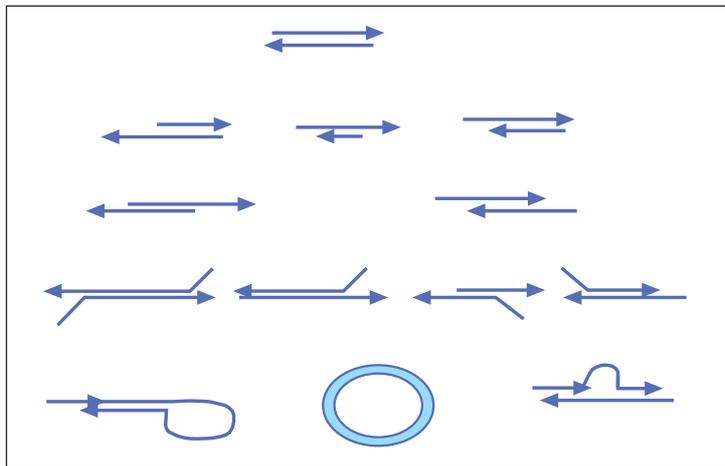


FIGURA 3
Tipi basilari
di forme bilineari

di concatenazione fosfodiesterica, diciamoli Testa e Coda, e il punto intermedio tra la propria testa e quella del nucleotide ad esso appaiato. In definitiva, il modulo compositivo che rende completamente conto della doppia struttura lineare si riduce in termini semplificati ad un triangolo, e le coppie allineate di triangoli si possono pensare come sviluppate entro il cilindro di figura 2, ove, per semplificare ulterior-

mente, si assume che le dimensioni dei triangoli siano sempre le stesse. Nella figura 2 i simboli H e T indicano la testa e la coda di ciascun elemento compositivo, mentre H' e T' sono la testa e la coda dell'elemento appaiato. Si verifica che, se l'angolo formato dalla direzione di concatenazione con quella di appaiamento è acuto (come richiesto da una maggiore ottimizzazione spaziale), allora i due filamenti devono necessariamente essere orientati in modo opposto [14]. Inoltre, per evitare che i legami di concatenazione e di appaiamento di entrambi i nucleotidi siano sullo stesso piano (cosa impossibile in un ambiente fluido) deve esservi una rotazione tra i piani dei due triangoli relativi, e quindi la doppia struttura deve svolgersi nello spazio. I tre angoli ρ , τ , ϕ , indicati in figura 2, insieme al raggio del cilindro, identificano completamente lo schema della doppia elica sviluppata intorno al cilindro [14, 19]. La struttura polimerica (etimolog. "di molte parti") doppia ha un profondo significato computazionale. Infatti, sull'appaiamento di porzioni complementari sono basate le principali operazioni tra filamenti di DNA, ed in particolare la duplicazione, che è un'operazione fondamentale per i meccanismi ereditari della vita, e che, come vedremo, consente la realizzazione di procedimenti efficienti per produrre un numero esponenziale di copie di una data molecola (per mezzo di un algoritmo su doppie sequenze che produce, in n passi, 2^n copie di molecole uguali ad una singola molecola data). La *bilinearità*, la *complementarietà* e l'*antiparallelismo* determinano una ricchezza di possibili forme DNA del tutto inimmaginabile. Un recente campo in rapida evoluzione del *DNA computing* si basa sullo studio di combinazioni inedite di molecole che, sfruttando tali principi costitutivi, possano produrre molecole autoaggreganti composte da "pezzi" elementari e con caratteristiche spaziali di interesse per applicazioni in vari settori delle nanotecnologie. Tale campo di indagine che va sotto il nome di *DNA Self-Assembly* ha a sua volta aspetti di interesse computazionale, in quanto si dimostra che la stessa nozione di calcolo può essere sviluppata in termini di generazione di forme secondo moduli di assemblaggio predefiniti [22, 24]. Per dare solo una semplice idea della ricchezza combinatoria delle strutture bilineari, riportiamo nella figura 3 le varie forme bilinea-

ri di base che si possono venire a creare (il parallelismo indica complementarità e il non parallelismo non complementarità). Le ultime tre in fondo sono le cosiddette forme, *hairpin*, circolari e *heteroduplex* (negli *heteroduplex* si formano delle lacune di appaiamento all'interno di due filamenti appaiati, mentre gli *hairpin* sono realizzati da uno stesso filamento in cui una parte ibridizza con un'altra sua parte). Tali forme possono combinarsi producendo una varietà enorme di possibilità, inoltre esistono almeno tre forme diverse di avvolgimenti elicoidali, vi sono casi di intrecci a più di due filamenti, vi sono avvolgimenti esterni, intorno ad assi trasversali a quello dell'elica e meccanismi di compattamento per ottimizzare l'occupazione di spazio quando il DNA non è in fase di elaborazione.

Le molecole di DNA hanno un intrinseco carattere *informazionale*, ed in effetti il loro ruolo biologico è quello di memorizzare i "programmi" che dirigono il funzionamento degli organismi viventi, dai più semplici ai più complessi. Quindi non stupisce che il DNA possa essere considerato come il supporto naturale per processi di calcolo. Ma in questo caso come manipolarlo per svolgere calcoli matematici? E soprattutto, quale è il vantaggio di un tale uso? Da un punto di vista tecnico, le stringhe non sono altro che sequenze di simboli tra le quali è definita un'operazione di *concatenazione*, la stessa che a partire dalle parole "capi" e "tombolo" produce la parola "capitombolo". Anche i calcoli che si svolgono all'interno di un computer sono in ultima analisi riconducibili a manipolazioni di stringhe (dei due simboli 0, 1). Tuttavia, la realtà fisica di tali stringhe è completamente diversa e le operazioni che su di esse si eseguono sono di tutt'altra natura. In definitiva, calcolare con il DNA richiede un ripensamento della stessa nozione di calcolo, ricostruendo in termini nuovi i metodi risolutivi degli algoritmi tradizionali. Questo è di per sé il primo grande interesse del *DNA Computing* nel quadro della ricerca di nuovi modelli di calcolo.

3. CALCOLI NATURALI

Un calcolo si svolge sempre con il supporto di un qualche sistema fisico. Tale sistema viene configurato in modo da codificare i dati da elab-

borare, partendo da un suo stato "iniziale", quindi lo si fa evolvere con una serie di passi, o agendo sullo stato di ogni passo con un "comando" che ne induce una transizione, oppure secondo un "programma" interno al sistema che induce una sequenza di transizioni a partire da uno stato iniziale. Quando il sistema giunge ad uno stato finale (secondo un certo criterio di terminazione), allora si decodifica tale stato estraendo da esso i risultati del calcolo. Se il sistema di supporto del calcolo è un sistema naturale, o ispirato a qualche sistema della natura, si parla di "calcolo naturale".

La ricerca di modelli di calcolo diversi da quelli sviluppati a partire dagli anni '30 del Novecento, secondo il paradigma di Turing-von Neumann, ha seguito vari percorsi ed è stata sollecitata da vari problemi. Già negli anni '40 e '50 varie metafore biologiche avevano ispirato strutture che si rivelarono di importanza centrale in vari contesti informatici (Reti neurali di McCulloch e Pitts nel 1943, Automi a stati finiti di Kleene nel 1956).

Il successivo sviluppo tecnologico del modello di von Neumann, per quanto poderoso, ha indicato dei limiti fisici intrinseci a cui la tecnologia odierna si sta sempre più avvicinando. Inoltre, la teoria matematica della complessità ha dimostrato che una grande quantità di problemi interessanti rimangono al di là delle possibilità del calcolo tradizionale.

Nei procedimenti generativi delle grammatiche di Chomsky, introdotte negli anni '60, la produzione di certe forme avveniva secondo meccanismi di controllo decentralizzato ove, tra tutte le possibili parole generate, quelle in cui le regole venivano applicate in modo "corretto" riuscivano a "terminalizzare", mentre le altre producevano forme "immature" che venivano scartate quando si "raccolgevano" i risultati [21]. In genere, nei processi di elaborazione di *stringhe* (sequenze a struttura concatenativa) si evidenziavano metodi in cui l'informazione fluiva secondo strategie "evolutive" che non erano quelle tipiche dei "calcoli a programma".

Alla fine degli anni '60, nel quadro della teoria dei linguaggi formali Lindenmayer sviluppò dei sistemi, poi detti L sistemi, che usando metodi di manipolazione di stringhe, tipici della teoria dei linguaggi formali, permisero di derivare gli stadi di sviluppo di organismi bio-

logici quali alge (l'alga rossa fu il primo organismo modellato con tali sistemi) [21].

Gli algoritmi genetici sviluppati negli anni '70 [12], gli automi cellulari sviluppati negli anni '80 [25], a partire da studi iniziati da von Neumann, e le reti booleane, introdotte per modellare le reti genetiche, furono altri modelli a forte "ispirazione naturale".

Nel 1987 Tom Head introdusse un'operazione su stringhe, detta di *splicing*, che formalizzava il meccanismo di ricombinazione genetica e furono dimostrate interessanti relazioni con importanti classi di linguaggi formali [11, 16]. In particolare lo *splicing* individua un meccanismo di trasformazione combinatoria di stringhe completamente diverso dal rimpiazzamento tipico delle grammatiche di Chomsky e degli automi di riconoscimento (rappresentati come strutture di manipolazione di stringhe). Due stringhe, diciamo α , β , si ricombinano per *splicing* $u_1\#u_2\$u_3\#u_4$ (è il modo standard di indicare le regole di *splicing*, e purtroppo in biologia molecolare il termine *splicing* indica un fenomeno diverso) se, possono essere fattorizzate, per opportune stringhe x, y, w, z , rispettivamente in $\alpha = xu_1u_2y$, $\beta = wu_3u_4z$. In tal caso la regola $u_1\#u_2\$u_3\#u_4$ produce le nuove stringhe xu_1u_4z, wu_3u_2y . Tale operazione ha un evidente rilevanza biologica e per visualizzarla in modo semplice basti pensare alle chimere tipiche della fantasia mitologica (sirene, centauri, minotauri, grifoni, capricorni) in cui partendo da due animali, ricombinazione le parti si ottengono nuove forme animali. Le coppie stringhe (u_1, u_2) e (u_3, u_4) sono i cosiddetti "siti di riconoscimento" e individuano i punti in cui tagliare le stringhe da ricombinare. Un risultato notevole stabilisce che a partire da un insieme finito di stringhe e usando un insieme finito di regole di *splicing* si possono ottenere linguaggi formali in una classe strettamente inclusa nella classe dei linguaggi regolari (lemma di subregolarità [20, 16]). Tuttavia, aggiungendo delle estensioni del tutto naturali, tale meccanismo diventa *computazionalmente universale*, ovvero permette di ottenere sistemi con la stessa potenza di calcolo delle macchine di Turing [20].

In tale linea di ricerca, le strutture discrete di rappresentazione dell'informazione nei modelli di calcolo mostrano sorprendenti analogie

con la natura discreta dell'informazione genetica. E anche da un punto di vista cronologico, si nota uno svolgimento parallelo di "modelli digitali" che da una parte produce le prime macchine di calcolo a programma e dall'altro conduce, nel 1953, alla scoperta del modello di DNA, prima descritto, che portò al Nobel di Watson e Crick. In questa prospettiva, l'esperimento di Adleman del 1994, al di là del suo significato specifico e degli sviluppi che potrà avere, sembra l'epigono di una convergenza naturale tra discipline in cui l'informazione è un concetto fondante. Ed in un certo senso, il *DNA Computing* mostra un verso di interazione tra Informatica e Biologia che è complementare all'interazione InfoBio propria della Biologia computazionale, alla base dei successi straordinari dei sequenziamenti genomici su grande scala ottenuti negli anni recenti.

Per concludere, è opportuno ricordare che nel 1998 Gheorghe Paun, ispirandosi ai meccanismi di compartimentazione di popolazioni di molecole nelle membrane biologiche, ha introdotto i *P* sistemi [21]. In essi sono espresse, in termini simbolici, sia regole che trasformano oggetti, sia regole che li spostano da una membrana ad un'altra. Con opportuni adattamenti, tali modelli risultano adeguati nella rappresentazione di dinamiche biologiche, fornendo alternative promettenti ai classici modelli basati su equazioni differenziali [5, 15].

4. OPERAZIONI DI BASE SU DNA

Diamo nel seguito una descrizione sintetica delle operazioni su molecole di DNA che risultano fondamentali nel *DNA Computing* [20, 8] (riquadro di p. 25). Sorvoleremo sui dettagli biochimici e di laboratorio che talvolta sono molto complessi, soffermandoci solo sugli aspetti informativi di elaborazione di sequenze e doppie sequenze.

Elementi fondamentali per realizzare le operazioni DNA sono: nucleotidi e molecole DNA, *contenitori (provette, piastre, ...), sorgenti di energia ed enzimi, oltre a reagenti, supporti e strumenti* per la misurazione di unità chimico-fisiche, per la rilevazione e visualizzazione di molecole e per la attivazione e la regolazione dei processi biochimici. Gli enzimi dal punto di vista biochimico sono dei *catalizzatori* di pro-

Operazioni di base nel DNA Computing

1. **Denaturazione** = Disaccoppiamento di sequenze nucleotidiche appaiate.
2. **Rinaturazione** = Riaccoppiamento di sequenze nucleotidiche disaccoppiate.
3. **Unione** = Versamento del contenuto DNA di una provetta in un'altra contenente altro DNA.
4. **Divisione** = Distribuzione del contenuto di una provetta in due provette.
5. **Ibridizzazione** = Appaiamento, per complementarità, di due singoli filamenti DNA.
6. **Amplificazione** = Generazione di molte copie uguali di una data porzione di DNA (doppio).
7. **Sequenziamento** = Lettura della sequenza di basi che caratterizza una data molecola di DNA.
8. **Sintesi** = Produzione sintetica di molecole di DNA aventi una specificata sequenza di basi.
9. **Gel-Elettroforesi** = Discriminazione di molecole di DNA, in bande di uguale lunghezza.
10. **Separazione** = Estrazione, dopo elettroforesi, di una banda di molecole di data lunghezza.
11. **Selezione per affinità** = Estrazione di molecole di DNA che includono una data sottosequenza.
12. **Ligasi** = Concatenazione fosfodiesterica tra due filamenti contigui appaiati ad uno stesso filamento.
13. **Estensione** = Allungamento nel verso 5'-3' "copiando" dal filamento antiparallelo appaiato.
14. **Blocco** = Preclusione di concatenazione nell'estremo OH o PO₄ (con possibile sblocco successivo).
15. **Ancoraggio** = Adesione di un frammento singolo di DNA ad un supporto solido.
16. **Restrizione** = Taglio, tramite enzima, di doppi filamenti DNA che contengono una porzione specifica.

cessi biomolecolari, ma dal punto di vista informazionale fungono da veri e propri "automi di calcolo biomolecolare", specializzati in operazioni elementari di riconoscimento, attivazione e regolazione di specifici processi di elaborazione di biopolimeri. Essi sono particolari proteine e quindi codificati da opportuni geni. Oltre alla *ligasi*, alla *polimerasi*, sono importanti gli enzimi di *restrizione*, che tagliano sequenze contenenti certe specifiche sottosequenze (suddividendole opportunamente tra le sequenze prodotte dal taglio). Tale caratteristica si presta a vari usi, di cui alcuni molto sofisticati, ma soprattutto consente di frammentare porzioni molto lunghe di DNA in modo da potere elaborare più facilmente i frammenti prodotti, cosa che risulta fondamentale nel sequenziamento di interi genomi. Altre classi di enzimi realizzano operazioni più complesse di cui non parleremo affatto, citiamo solamente alcuni nomi: *trascrittasi inversa*, *terminal-trasferasi*, *topoisomerasi*, *integrasi*, *elicasi*. Da un punto di vista matematico, gran parte delle operazioni DNA che consideriamo non agiscono su singole molecole, ma piuttosto su popolazioni di molecole (a doppio o singolo filamento) e anche gli enzimi occorrono in popolazioni, in cui ciascuna copia dello stesso enzima svolge lo stesso compito dei suoi simili, ed in parallelo ad essi, sulle molecole appropriate alla sua funzione. Ciascuna molecola DNA costituisce una occorrenza individuale di un tipo espresso da una sequenza (singola o doppia) di basi (una

parola). Tale nozione di popolazione corrisponde al concetto matematico di *multinsieme* in cui ciascun elemento (filamento) non solo appartiene o non appartiene ad un multinsieme, ma vi occorre in un certo numero di copie (l'inclusione, l'unione, la differenza ed altre operazioni insiemistiche si estendono in modo standard ai multinsiemi). Ad ogni popolazione P di molecole DNA è associato quindi un insieme di stringhe (singole o doppie), diciamo $Type(P)$, ovvero un *linguaggio formale*, costituito da tutti i tipi delle molecole di P . L'effetto di molte operazioni consiste nel passaggio da una popolazione con un certo tipo ad un'altra con un altro tipo. In questo senso, spesso non è importante sapere esattamente come è fatta una certa popolazione, ma solamente conoscerne il tipo ed essere in grado di manipolarla in modo da farle assumere il nuovo tipo desiderato.

Quando le molecole di DNA si trovano in un ambiente in cui la temperatura supera valori intorno agli 80° Celsius, esse si **denaturano**, ovvero si scindono nei due filamenti singoli. La temperatura a cui questo avviene dipende da parametri chimico-fisici determinati dalle sequenze di basi. Abbassando la temperatura i due filamenti si riappaiano in modo tale che ciascun nucleotide si lega con il suo complementare. Il processo inverso alla denaturazione si dice **rinaturazione**. La figura 4 illustra la denaturazione e la rinaturazione di un doppio filamento. Si consideri una situazione in cui si hanno filamenti singoli, per esempio

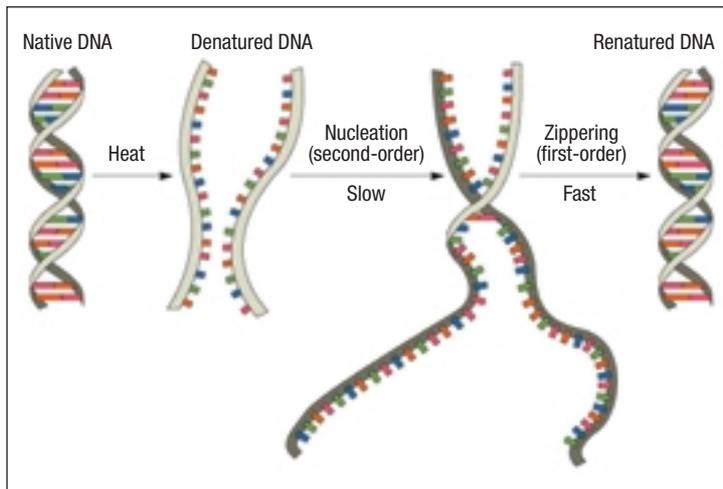


FIGURA 4
Denaturazione e Rinaturazione (da [10])

di tipo NNNNNCACTTGNNNNN, in cui N indica un generico nucleotide (la scrittura si sottintende nel verso 5'-3'). Allora, ponendo frammenti a singolo filamento di tipo CAAGTG, che è antiparallelo e complementare a CACTTG (cioè complementare alla sua sequenza invertita), sotto appropriate condizioni termiche, si ottiene l'appaiamento tra CAAGTG e NNNNNCACTTGNNNNN secondo una struttura che possiamo rappresentare come segue (nella sequenza inferiore, si sottintende il verso 3'-5'):



Tale fenomeno si chiama *ibridizzazione* o *annealing*. Le operazioni di appaiamento e disappaiamento di nucleotidi sono possibili perché il legame chimico che lega le basi appaiate è un legame a idrogeno notevolmente più debole rispetto agli altri legami interni ai nucleotidi e ai legami fosfodiesterici che legano gli zuccheri. Quando, dopo una ibridizzazione, due porzioni di DNA si trovano ad essere contigue in uno stesso filamento, un enzima *ligasi* svolge il compito di creare un legame fosfodiesterico tra i nucleotidi che si trovano accanto. Tale operazione può essere considerata una forma particolare di concatenazione di sequenze, nel contesto dei doppi filamenti.

L'operazione di *gel-elettroforesi* consiste nel depositare il DNA di una provetta in una piastra ai cui bordi è applicata una differenza di

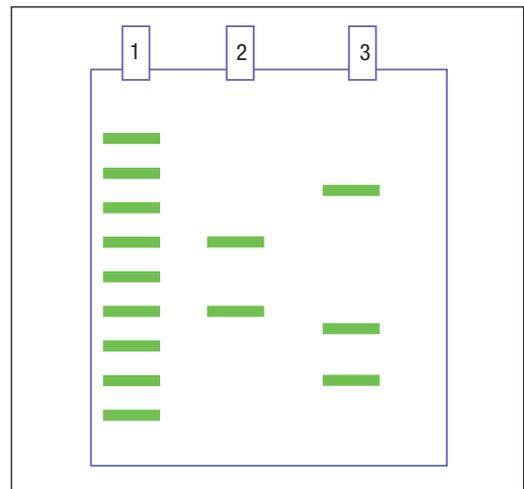


FIGURA 5
Forma stilizzata di una lastra radiografica di rilevazione dopo gel-elettroforesi. In colonna 1 è posto il ladder di marcatori. Le lunghezze (in bp) delle bande si valutano confrontandone la posizione rispetto ai marcatori.

potenziale elettrico e sulla superficie è stato depositato un gel opportuno (di Agarosio o Poliaccrilammide). In tal modo, in base al fatto che le molecole di DNA hanno una leggera carica negativa (nel gruppo fosforico), si ha una migrazione dal polo negativo a quello positivo la cui velocità è inversamente proporzionale alla lunghezza, in quanto la maggiore lunghezza determina un maggiore attrito sulla superficie del gel. Dalla posizione in cui le molecole si trovano, dopo una "corsa" che dura un tempo assegnato, si risale alla lunghezza delle molecole. Infatti, dopo la corsa, le molecole si raggruppano in bande di uguale lunghezza che si evidenziano attraverso rilevatori di radiazioni che impressionano delle lastre radiografiche (Figura 5). L'altezza delle bande rilevate è posta in relazione all'altezza di bande di riferimento, di lunghezze note, e quindi si riesce a risalire con notevole precisione alla lunghezza delle bande ottenute. In tal modo, oltre a suddividere le molecole per lunghezza, si può separare una banda di una certa lunghezza. Basta incidere la parte di gel in cui si trova una certa banda e quindi, dopo aver rimosso il gel, liberare il DNA in esso contenuto riportandolo in provetta.

L'operazione di *sequenziamento* corrisponde alla lettura di una sequenza e si realizza con un metodo (dovuto a Sanger, che per questa scoperta ha ottenuto il Nobel per la seconda

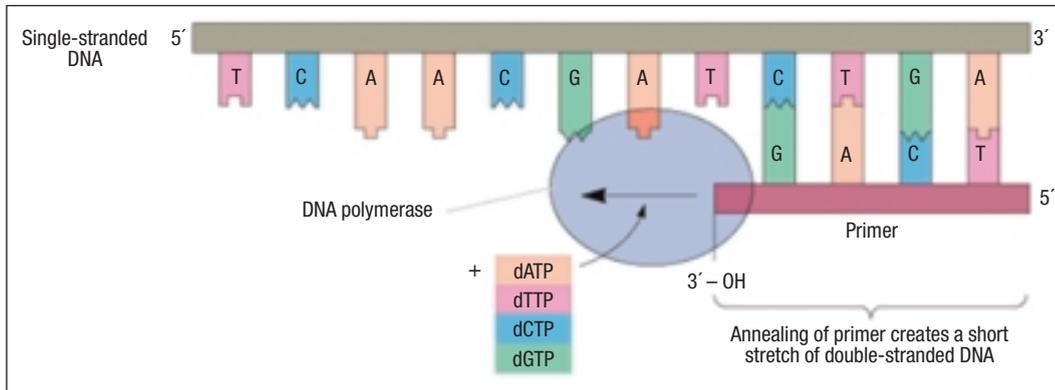


FIGURA 6
L'enzima Polimerasi
in azione (da [10])

volta) che descriveremo solo in modo metaforico. A prima vista potrebbe sembrare che la lettura di una sequenza si possa fare in qualche modo non troppo complicato, magari con qualche ultrapotente microscopio. Ma non è così. In tutti i modelli nanotecnologici, leggere è complicato e dispendioso, molto più che scrivere! Dapprima si producono molte copie della sequenza da leggere, quindi si dividono tali copie in 4 provette distinte che diremo P_T , P_A , P_C , P_G in ciascuna delle quali si usa un qualche meccanismo che ha l'effetto di tagliare le sequenze in un solo punto: in P_T sempre dopo una base **T**, in P_A sempre dopo una base **A**, in P_C sempre dopo una base **C**, e in P_G sempre dopo una base **G**. Quindi utilizzando il processo di gel-elettroforesi si determinano le lunghezze di tutti i frammenti presenti in P_T e analogamente le lunghezze di quelli in P_A , P_C , P_G . Dalle misure di tali lunghezze si scoprono quindi le posizioni delle varie basi nella sequenza originale. In definitiva, è come se, volendo leggere una parola se ne facessero miliardi di copie e successivamente si distruggessero, risalendo dai resti di tale distruzione alla parola data. Ovviamente i dettagli biochimici di tale processo sono piuttosto complessi, e soprattutto è cruciale modulare il meccanismo dei tagli per essere sicuri di realizzarne un numero adeguato, in tutte le posizioni in cui essi sono possibili. Comunque, si può notare che, nell'essenza, il metodo di sequenziamento è un algoritmo su sequenze, o meglio su multinsiemi di sequenze.

Se esiste un'operazione per "leggere" una sequenza, deve esserci anche un'operazione per potere "scrivere" una sequenza data. Tale operazione corrisponde alla *sintesi* di sequenze DNA, ovvero al procedimento secondo cui, da

una data sequenza simbolica si produce un *clone* di una molecola del tipo specificato dalla sequenza. Il termine clone indica semplicemente che si tratta di una certa quantità di molecole (per esempio una pico-mole), tutte dello stesso tipo. Un tale processo ormai avviene in modo molto efficiente usando iterativamente tre meccanismi di base:

- i) l'*ancoraggio* di molecole ad un supporto solido,
- ii) il *blocco* di concatenazione in posizione 5' dei nucleotidi, suddivisi secondo i loro quattro tipi,
- iii) il successivo *sblocco*, non appena l'estremo in 3' di un nucleotide si è concatenato alla sequenza ancorata. Senza entrare in ulteriori dettagli, possiamo dire che di fatto basta ordinare (ormai *online*) ad una ditta specializzata la sequenza desiderata per ottenere, ad un prezzo inferiore ad un dollaro per base, una certa quantità molare di molecole del tipo richiesto (provate a fare esperimenti con 20 sequenze da 100 basi, reagenti, enzimi, ...!).

L'operazione *extract* è fondamentale per il DNA computing. Essa consiste in una *selezione per affinità*, che permette di estrarre da una provetta molecole che contengano una data porzione di DNA. Un modo per realizzare tale selezione è quello di usare sequenze complementari (e antiparallele) alla porzione da "pescare", ancorandole a opportuni supporti solidi. In tal modo, dopo aver favorito l'ibridazione, si estraggono dalla soluzione i supporti solidi e quindi con essi anche le molecole ibridate ai frammenti ancorati.

L'*estensione* è un'operazione chiave svolta dall'enzima *polimerasi*. Tale enzima estende un filamento, nel verso 5'-3', usando l'altro filamento come stampo (template). La figura 6 visualizza tale meccanismo. Una volta che una

piccola sequenza è appaiata ad un singolo filamento (*primer*, ovvero innesco), la parte mancante del filamento viene ricostruita dall'enzima polimerasi se nella soluzione sono presenti dei nucleotidi singoli (in forma trifosfata: dATP sta per "deossiadenuclotridifosfato", dTTP sta per "deossitimidinotridifosfato", e via dicendo) che vengono correttamente allineati secondo la regola di appaiamento di Chargaff. Più precisamente, in soluzione si pongono deossinucleosidi 5'-trifosfato, in cui rispetto ai nucleotidi, in 5' vi è $P_3 O_{10}$ (un trifosfato al posto di un fosfato PO_4 , vedi riquadro a p. 21). Quindi il deossinucleoside 5'-trifosfato entra nella catena trattenendo un PO_4 che realizza il legame fosfodiesterico con il nucleotide a cui si concatena e da questo elimina l'ossidrile OH, rilasciando quindi un pirofosfato $P_2 O_6 OH$ (l'ossidrile in 3' del nucleotide a cui si concatena viene sostituito da un ossigeno del PO_4).

Operazioni semplici da realizzare, ma estremamente utili nella costruzione di algoritmi DNA sono *merge* e *split*, ovvero *unione* e *divisione*. Con la prima si mescolano in un'unica provetta i contenuti di due provette distinte, con l'altra si divide in due provette il contenuto di una (in quantità pressochè uguali).

Concludiamo la rassegna di operazioni DNA con un cenno al metodo principale di *amplificazione*, che permette di produrre molte copie di una sequenza di DNA. La sua realizzazione attraverso il procedimento della

PCR (*Polymerase Chain Reaction*) fu scoperta nel 1983 da Kary Mullis (laureato Nobel per questa scoperta nel 1993). Lo sviluppo di tutta la biologia molecolare è impensabile senza la PCR. Altri metodi di amplificazione, che rimangono tuttora attuali, sono basati sulla *clonazione* tramite vettori di clonazione (batteri e virus), ma richiedono procedimenti molto complessi su cui non possiamo soffermarci. L'idea geniale di Mullis fu quella di utilizzare l'estensione della polimerasi per "fotocopiare" una porzione voluta di un doppio filamento di DNA, delimitata da due primer. Il metodo usa un'idea astuta di regolazione della temperatura. Riferendoci alla figura 7 (relativa ad un caso semplice), si parte con una molecola perfettamente bilineare, diciamo *bersaglio*, e da due primer (usualmente di circa 20 bp) che indichiamo γ e $rev(\delta^c)$, ove δ^c indica la sequenza complementare δ e *rev* indica l'inversione di sequenze. Assumiamo che il primer γ indicato a sinistra si appai, per ibridizzazione, al filamento inferiore del bersaglio, e che quello indicato a destra $rev(\delta^c)$ si appai al filamento superiore del bersaglio. Regolando opportunamente le temperature, si ottengono i passi indicati nella figura 7:

i) il doppio filamento bersaglio si scompone nei suoi due filamenti,

ii) i due primer ibridizzano con le porzioni ad essi complementari della molecola bersaglio ii) la polimerasi estende i primer ibridati ciascuno nel verso 5' - 3' in modo da produrre due molecole uguali. Riapplicando tali tre passi (denaturazione, ibridizzazione dei primer, estensione dei primer ibridati) per altre n volte, alla fine del processo si ottengono 2^n molecole che iniziano con γ e che terminano con δ includendo la porzione che, nella molecola ottenuta al primo passo, era compresa tra γ e δ . Questo processo assume che nella provetta si trovino una quantità adeguata di:

i) nucleotidi singoli per alimentare l'estensione della polimerasi;

ii) unità di enzima polimerasi in modo che al generico passo k ogni singolo enzima possa, in parallelo con gli altri, condurre il processo di estensione sulle molecole bersaglio presenti in quell'istante.

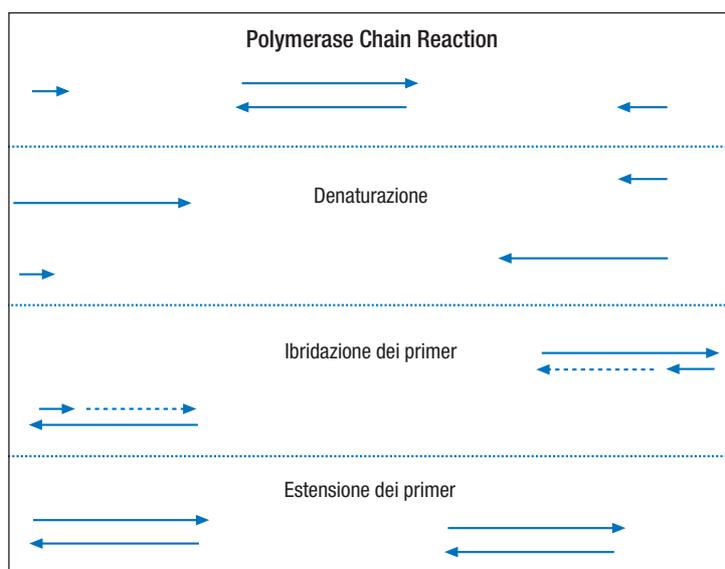


FIGURA 7
Fasi di un processo di PCR

5. L'ESPERIMENTO DI ADLEMAN

L'esperimento di Adleman prende le mosse da un tipico problema combinatorio detto del *cammino hamiltoniano* (dal matematico Hamilton che lo ha studiato). Dato un grafo orientato con *vertici* tra i quali vi sono degli *archi* di collegamento (orientati dal primo al secondo) si chiede di determinare, se esistono, dei percorsi che, partendo da un vertice iniziale, raggiungono un vertice finale passando per ciascun vertice una ed una sola volta (e attraversando gli archi secondo il loro orientamento). Si dimostra che al crescere del numero dei vertici la complessità di tale problema diventa proibitiva. In termini più precisi, possiamo dire che verificare se un dato cammino è hamiltoniano può essere fatto in un numero di passi linearmente proporzionale al numero di vertici del grafo, ma i possibili cammini da esaminare sono in numero talmente grande che se si dovessero considerare uno alla volta si impiegherebbero tempi superiori ad ogni reale possibilità. Quello del cammino hamiltoniano è un esempio della classe di problemi cosiddetti NP-completi (NP = *Nondeterministico Polinomiale*).

La soluzione di Adleman è molto intuitiva. Identifichiamo i vertici del grafo con i numeri 1, 2, ..., n e codifichiamoli con frammenti di DNA ad un solo filamento costituiti da due parti (tranne il vertice iniziale 1 e quello finale n). Siano $B_1, A_2 B_2, A_3 B_3, \dots, A_n$ le codifiche dei vertici con parole di DNA a singolo filamento. Un arco che collega il vertice i con quello j è allora codificato con una parola DNA, a singolo filamento, di due parti $A_j^c B_i^c$ ove A_j^c e B_i^c sono le sequenze DNA complementari ai pezzi A_j e B_i delle codifiche dei nodi i e j (in verso 3' - 5'). A questo punto, ponendo in provetta le codifiche DNA di vertici e di archi, queste, sfruttando il fenomeno di ibridizzazione, si compongono in strutture doppie (Figura 8) che utilizzando l'enzima ligasi si trasformano in doppi filamenti DNA. Ciascuno di questi rappresenta ovviamente un cammino nel grafo da 1 a n . Se vi sono moltissime copie di ciascuna codifica (diciamo 10^{14}) è auspicabile che tutti i possibili cammini possano essere prodotti.

L'algoritmo risolutivo di Adleman procede ad estrarre i cammini hamiltoniani con i passi seguenti:

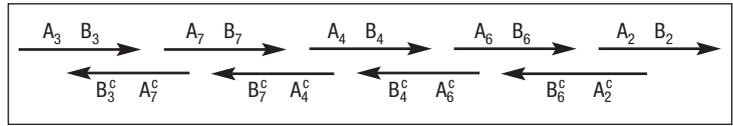


FIGURA 8

La formazione, per ibridizzazione, di un cammino di Adleman: le frecce superiori codificano i nodi, mentre le inferiori codificano gli archi. Gli spazi indicano la mancanza dei legami fosfodiesterici, poi realizzati dall'enzima ligasi

1. Si pongono in provetta le codifiche dei nodi e degli archi, favorendo la formazione di cammini del tipo riportato nella figura 8;
2. Si esegue un'amplificazione tramite PCR con primer B_1 e $rev(A_n^c)$, in modo da incrementare il numero di filamenti che codificano cammini che iniziano e finiscono come richiesto dal problema;
3. Si separano con gel-elettroforesi i filamenti di lunghezza corrispondente a quella dei cammini hamiltoniani (in generale, $(2kn - 2k) = 2k(n - 1)$ ove k è la lunghezza dei pezzi A e B);
4. Si selezionano per affinità i filamenti in cui occorrono le codifiche di tutti i vertici;
5. I filamenti rimasti in provetta codificano i cammini hamiltoniani cercati.

La separazione del terzo passo consente di escludere cammini che sicuramente non possono rappresentare soluzioni, in quanto non avendo una lunghezza data non possono passare una ed una sola volta da tutti i vertici del grafo. La selezione del quarto passo invece serve ad accertarsi che tutti i nodi siano presenti nei cammini. Mettendo insieme le condizioni soddisfatte al terzo e al quarto passo, si garantisce che sicuramente un vertice appare una ed una sola volta in un cammino. Inoltre la amplificazione eseguita al secondo passo consente di ritenere trascurabile la possibilità di selezionare un cammino che non cominci e che non finisca nel modo richiesto. Nel caso che vi sia un solo cammino hamiltoniano (come nell'esperimento originale di Adleman) tale filamento viene sequenziato per "leggere" la sua sequenza di basi. Nel caso che vi possano essere più cammini hamiltoniani il procedimento si complica, ma si possono comunque determinare tutti i diversi cammini prodotti come risultato¹.

¹ Vi sono molte ottime e dettagliate presentazioni dell'esperimento di Adleman, facilmente reperibili in rete ricercando *Adleman experiment o DNA computing*.

6. OLTRE IL MODELLO DI ADLEMAN

È stato calcolato che un Joule è sufficiente per realizzare 2.10^{19} operazioni di ligasi, laddove un potente PC attuale arriva a circa 10^9 operazioni elementari per Joule, e che si possono svolgere circa 2.10^{18} operazioni DNA al secondo contro le 2.10^{12} di un attuale processore. Infine in pochi grammi di DNA si potrebbero codificare tutte le informazioni attualmente registrate sui supporti elettronici sparsi nel mondo. Tali valutazioni numeriche, sebbene formulate in termini del tutto astratti e sotto ipotesi particolari, indicano la “potenza informazionale” del DNA. Del resto, parlando sempre in termini generali, se si considerano i genomi degli organismi superiori ci si rende conto che la quantità di informazione digitale di un genoma (circa 3.10^9 basi nell'uomo) risulta del tutto paragonabile a quella di un sistema operativo moderno (parecchie decine di milioni di linee di codice). Quindi, ogni cellula si porta dietro il sistema operativo dell'organismo di cui fa parte come gli impiegati di un ufficio hanno lo stesso PC, anche se ciascuno fa girare solo un certo tipo di programmi legati all'attività che svolge nell'ufficio.

A partire dal 1995 si sono tenute delle conferenze internazionali su *DNA Based Computers* (siamo già alla dodicesima conferenza di tale ciclo). In un lavoro dello stesso anno Lipton, basandosi sul risultato di Adleman, definì lo schema generale di un algoritmo DNA

per la soluzione di problemi combinatori [13]:
i) generazione dello spazio delle “possibili” soluzioni, codificate con filamenti DNA,

ii) estrazione delle “vere” soluzioni tramite meccanismi opportuni di selezione.

In tale schema la massiva ricombinazione del DNA è il punto cruciale per generare le soluzioni possibili a partire da frammenti di DNA che codificano i dati del problema. Tale schema è stato applicato in moltissimi modi, soprattutto per la soluzione di una famosa classe di problemi combinatori detti SAT (*Boolean Satisfiability Problem*), ed in particolare 3-SAT (n, m), in cui n ed m indicano numeri interi positivi, relativi ad un sistema di equazioni booleane (0,1 con somma, e negazione booleana) di n variabili ed m equazioni, in ciascuna delle quali occorrono al più 3 variabili. Nel riquadro è fornito un esempio di soluzione di un problema SAT, riconducibile allo schema di Lipton. Non entriamo nel dettaglio dell'algoritmo. Esso usa un interessante metodo, detto *mix-and-split*, che genera lo spazio delle soluzioni in un tempo linearmente proporzionale al numero di variabili delle equazioni booleane da risolvere. Ci limitiamo a sottolineare che:

i) si tratta di un procedimento a passi basato su operazioni che agiscono su provette;

ii) in particolare l'assegnamento $:=$ è da intendersi in modo analogo a quello dei linguaggi di programmazione, ovvero, la variabile alla sinistra di $:=$ indica una provetta in cui si pone il risultato di un'operazione applicata ai contenuti delle provette alla destra dell'assegnamento.

Nonostante si siano individuate moltissime strategie risolutive per SAT [17, 18], di cui alcune di grande interesse teorico, il risultato migliore fino ad ora ottenuto è relativo ad un problema con 20 variabili [4]. Questa difficoltà ha orientato sempre di più le ricerche successive verso altre direzioni.

Negli ultimi anni appare evidente un interesse crescente alla analisi algoritmica di procedimenti di manipolazione del DNA che possano migliorare tecniche biotecnologiche o fornire nuovi metodi in indagini diagnostiche o terapeutiche. Si è compreso che i “calcoli” svolti dalla natura e gli stessi protocolli biotecnologici includono aspetti algoritmici interessanti che si prestano a varie analisi e

Algoritmo di Lipton per la soluzione di problemi del tipo 3-SAT(N, M)

T, T_1, T_2, T_3 provette, $L(1, j), L(2, j), L(3, j)$ termini booleani dell'equazione j -esima.

begin

1. Genera lo spazio delle soluzioni di dimensione N via “mix-and-split” e ponilo in T ;

2. for $j = 1$ to M **do**

begin

$T_1 := \text{extract}(T, L(1, j))$;

$T := T - T_1$;

$T_2 := \text{extract}(T, L(2, j))$;

$T := T - T_2$;

$T_3 := \text{extract}(T, L(3, j))$;

$T := \text{merge}(T_1, T_2)$;

$T := \text{merge}(T, T_3)$

end;

3. if $T \neq \emptyset$, **then** prendi un clone e sequenzialo, il risultato è una soluzione
else “Problema insolubile”
end.

possono suggerire interessanti applicazioni in campo biomedico. Per esempio, nei lavori [6, 7] si è sviluppata un'analisi combinatoria della PCR scoprendo aspetti matematici piuttosto complessi sulla base dei quali sono stati condotti vari esperimenti biotecnologici ed è emersa una variante della PCR, chiamata XPCR, che sembra adatta allo sviluppo di nuovi metodi nella elaborazione di molecole DNA, soprattutto per l'estrazione selettiva di tali molecole (evitando metodi di affinità chimica) e per la ricombinazione massiva di DNA. Per una comprensione dei fenomeni sottostanti risulta interessante definire una notazione ed un relativo linguaggio algoritmico che permetta di dominare l'enorme ricchezza combinatoria dei processi di elaborazione di strutture bilineari. In questo senso, la prospettiva iniziale del *DNA computing* sembra in un certo senso capovolta, poiché in questo caso è l'algoritmo su forme matematiche che descrive il processo naturale, piuttosto che il processo di trasformazione di molecole a realizzare algoritmi di interesse matematico. Un caso in cui questo punto di vista trova una realizzazione specifica è stato recentemente sviluppato per la comprensione di meccanismi genetici basilari in forme cellulari primitive [9].

7. CONCLUSIONI

Il *DNA Computing* è nato con l'idea di risolvere problemi combinatori sfruttando:

- i) la potenza di codifica del DNA,
- ii) il parallelismo massiccio con cui si opera su enormi popolazioni di molecole DNA,
- iii) il meccanismo associativo di ibridizzazione.

Tuttavia, tale prospettiva iniziale si è ampliata sempre più, mostrando come il rapporto tra stringhe simboliche e molecole di DNA abbia radici profonde, alla base dei meccanismi fondamentali di elaborazione dell'informazione naturale. Nessun sistema può sviluppare comportamenti complessi se non si avvale di procedimenti sofisticati di elaborazione dell'informazione (rappresentazione, strutturazione, trasformazione, memorizzazione e trasmissione). I sistemi naturali necessitano quindi di calcoli, ma spesso tali calcoli obbediscono a logiche diverse da quelli dei sistemi artificiali. Capire e riprodur-

re tali logiche diventa quindi uno strumento, oltre che di interesse applicativo, di comprensione e conoscenza della natura, uno strumento di vera *philosophia naturalis*, nella sua accezione scientifica più genuina.

Da un punto di vista più specifico, oltre alla ricerca di algoritmi DNA, un'altra linea di ricerca, come già anticipato, è rivolta allo studio di meccanismi di aggregazione di molecole DNA che possano produrre specifiche forme di interesse nel campo delle nanotecnologie. In tale ambito le strutture ottenute per ibridizzazioni a partire da elementi di base possono essere descritte tramite opportuni grafi e quindi, studiando caratteristiche specifiche di tali grafi, si possono dedurre interessanti proprietà delle forme che si vengono a produrre. Una tecnica, nota come *Whirlash PCR*, permette di realizzare in modo naturale automi DNA, in cui semplificando alquanto, lo stato dell'automa è essenzialmente codificato dalla posizione di ibridizzazione di una struttura *hairpin* del tipo indicato nella figura 3 (in basso a sinistra). Gli automi DNA costituiscono una frontiera nelle applicazioni biomediche del DNA computing. Per, esempio, recentemente [3, 23] si sono progettati piccoli automi DNA, battezzati, in modo forse eccessivamente avveniristico, *DNA Doctor*, in grado di riconoscere molecole "nemiche" e quindi di rilasciare antidoti per la loro neutralizzazione.

Per concludere, il *DNA Computing*, dalla soluzione in provetta di problemi combinatori, agli algoritmi biomolecolari, alla analisi computazionale di protocolli biomolecolari e fenomeni genetici, fino agli automi DNA, apre scenari di grande interesse sia teorico che applicativo, mostrando possibilità sperimentali del tutto imprevedibili per l'informatica e nuovi strumenti informatici e matematici al servizio della biologia. Tale campo di ricerca, seppur incrocia temi classici della Bioinformatica (sequenziamenti genomici, e ricerche di similarità in sequenze di biopolimeri), nello stesso tempo si rivolge a temi fondamentali di discipline annunciate ed "in cerca di autore" quali *Cellular Computing*, *Biocomputing* e *Systems Biology* che saranno cruciali per la comprensione dei meccanismi di elaborazione dell'informazione biologica.

Bibliografia

- [1] Adleman L.M.: Molecular Computation of solutions to combinatorial problems. *Science*, Vol. 266, 1994, p. 1021-1024.
- [2] Alberts B., Raff M.: *Essential Cell Biology. An introduction to the molecular biology of the cell*. Garland Science, New York, 1997. (Trad. it. Zanichelli, Bologna, 2005).
- [3] Benenson K., Paz-Elitzur T., Adar R., Keinan E., Livneh Z., Shapiro E.: Programmable and Autonomous Computing Machine Made of Biomolecules. *Nature*, Vol. 414, 2001.
- [4] Braich R.S., Chelyapov N., Johnson C., Rothmund P.W.K., Adleman L.: Solution of a 20-Variable 3-SAT Problem on a DNA Computer. *Science*, Vol. 296, 2002, p. 417-604.
- [5] Ciobanu G., Paun G., Perez-Jimenez M.J.: *Applications of Membrane Computing*. Springer, Berlin, 2006.
- [6] Franco G., Giagulli C., Laudanna C., Manca V.: *DNA Extraction by Cross Pairing PCR*. In: Ferretti C., Mauri G., Zandron C., et al. (Eds.): 10th International Meeting on DNA Based Computers, LNCS 3384. Springer-Verlag, Berlin, 2005, p. 106-114.
- [7] Franco G., Manca V., Giagulli C., Laudanna C.: *DNA Recombination by XPCR*. In: Carbone A., A Pierce N. (Eds.): 11th International Meeting on DNA Computing, LNCS 3892. Springer-Verlag, Berlin, 2006, p. 233-242.
- [8] Martyn A.: *Theoretical and Experimental DNA Computation*. Springer-Verlag, 2005.
- [9] Ehrenfeucht A., Harju T., Petre I., Prescott D.M., Rozenberg G.: *Computation in Living Cell. Gene assembly in ciliates*, Springer-Verlag, Berlin, 2004.
- [10] Garrett R.H., Grisham C.M.: *Biochemistry*. Saunders College Publishing, 1997. (Trad. it. Zanichelli, Bologna, 2002).
- [11] Head T.: Formal language theory and DNA: An analysis of the generative capacity of specific recombinant behaviors. *Bulletin of Mathematical Biology*, Vol. 49, 1987, p. 737-759.
- [12] Holland J.: Genetic algorithms. *Scientific American*, Vol. 267, n. 1, 1992, p. 66.
- [13] Lipton R.J.: DNA solutions of hard computational problems. *Science*, Vol. 268, 1995, p. 542-544.
- [14] Manca V.: On the Logic and Geometry of Bilinear Forms. *Fundamenta Informaticae*, Vol. 64, n. 1-4, 2005, p. 261-276.
- [15] Manca V., Bianco L., Fontana F.: *Evolutions and oscillations of P systems: Theoretical considerations and applications to biochemical phenomena*. In: Membrane Computing, Mauri G., Paun G., Perez-Jimenez M.J., Rozenberg G., Salomaa A., (eds), LNCS 3365, Springer, Berlin, 2005, p. 63-84.
- [16] Manca V.: *A proof of regularity for finite splicing*. In: Jonoska N., Paun, Rozenberg G. (Eds.), Aspects of molecular computing, LNCS 2950, Springer-Verlag, Berlin, 2004, p. 309-317.
- [17] Manca V., Zandron C.: *A Clause string DNA Algorithm for SAT*. In: Jonoska N., Seeman N.C. (Eds.), DNA Computing, LNCS 2340, Springer-Verlag, Berlin, 2003, p. 172-181.
- [18] Manca V.: DNA and Membrane Algorithms for SAT. *Fundamenta Informaticae*, Vol. 49, 2002, p. 171-175.
- [19] Manca V.: *On the logic of DNA bilinearity*. In: Hagiya M., Ohuchi A. (Eds.): DNA Computing, 8th International Meeting on DNA Based Computers, Preliminary Proceedings, Hokkaido, Japan, 2002, p. 330.
- [20] Paun G., Rozenberg G., Salomaa A.: *DNA Computing*. Springer-Verlag, Berlin, 1998.
- [21] Rozenberg G., Salomaa A., (Eds.): *Handbook of Formal Language Theory*, 3 Voll. Springer-Verlag, Berlin, 1997.
- [22] Seeman N.C.: DNA in a material world blocking. *Nature*, Vol. 421, 2003, p. 427-431.
- [23] Shapiro N., Benenson Y.: Bringing DNA Computers to Life. *Scientific American*, April, 2006, p. 44-51.
- [24] Soloveichik D., Winfree E.: *Complexity of Self-assembled Shapes*. LNCS 3384, Springer-Verlag, Berlin, 2005, p. 244-354.
- [25] Wolfram S.: *A new kind of science*, Wolfram Media, 2002.

Figure

Le figure 1, 2, 3, 5, 7, 8 sono tratte dagli appunti del corso "Modelli di calcolo non convenzionali" tenuto dall'autore presso l'Università di Verona nell'a.a. 2005/2006.

VINCENZO MANCA è professore ordinario di Informatica presso l'università di Verona. I suoi interessi di ricerca includono i modelli di calcolo non convenzionali (*DNA e Membrane Computing*) e l'analisi informazionale di sistemi biologici. È membro dell'*European Molecular Computing Consortium* ed è stato Visiting Professor ed Invited Speaker in varie istituzioni e conferenze internazionali. È autore di più di 80 pubblicazioni scientifiche internazionali. Ha diretto vari progetti di ricerca, e ha ideato e coordinato esperimenti biotecnologici su algoritmi DNA.
E-mail: vincenzo.manca@univr.it

ICT E INNOVAZIONE D'IMPRESA

Casi di successo

Rubrica a cura di

Roberto Bellini, Chiara Francalanci

La rubrica *ICT e Innovazione d'Impresa* vuole promuovere la diffusione di una maggiore sensibilità sul contributo che le tecnologie ICT possono fornire a livello di innovazione di prodotto, di innovazione di processo e di innovazione di management. La rubrica è dedicata all'analisi e all'approfondimento sistematico di singoli casi in cui l'innovazione ICT ha avuto un ruolo critico rispetto al successo nel business, se si tratta di un'impresa, o al miglioramento radicale del livello di servizio e di diffusione di servizi, se si tratta di una organizzazione pubblica.



Caso Funambol

Roberto Bellini

1. INTRODUZIONE

Abbiamo incontrato Fabrizio Capobianco, fondatore e CEO di Funambol, al Congresso AICA tenutosi a Cesena, dove era stato invitato a portare la sua testimonianza in quanto amministratore delegato di un'azienda di successo nel campo dell'open source. Funambol nasce cinque anni fa circa, grazie ad una intuizione di Fabrizio Capobianco, il quale individua uno spazio di mercato per un nuovo componente interstiziale nella grande famiglia di componenti iperspecializzati del mercato della tecnologia mobile: il componente riguarda il problema della "sincronizzazione" dei dati che devono essere resi disponibili a grandi numeri di utenti. Fabrizio Capobianco ha fondato la società Funambol con un suo vecchio compagno di studi, Stefano Fornari, che oggi è il Direttore Tecnico di Funambol.

2. IL MERCATO DEI DISPOSITIVI MOBILI

Il mercato di Funambol è quello dei dispositivi mobili. La varietà dei dispositivi digitali mobili è destinata ad ampliarsi sempre di più, all'ampliarsi delle applicazioni nei vari settori merceologici. Una recente indagine del Politecnico di Milano ha messo in evidenza che i setto-

ri più sensibili all'uso di dispositivi mobili sono quelli del farmaceutico, dell'alimentare, della sanità, delle utility e delle banche. Per le imprese facenti parte di questi settori si stanno sviluppando applicazioni come quelle della *Sales Force Automation* (Automazione della Forza Vendita), del *Field Force Automation* (Automazione della Assistenza Tecnica), della Gestione dei Magazzini, dell'*Asset Management* (Gestione del Portafoglio Finanziario), delle Operazioni Mobili, della *Fleet Management* (Gestione della Flotta), del CRM (*Customer Relationship Management*), dell'Ufficio Mobile e senza fili, del Macchina verso Macchina. Gli addetti aziendali che lavorano con questi applicativi al di fuori dello stabilimento o dell'ufficio hanno la necessità di essere sempre reperibili, di mantenere sempre aggiornate le proprie agende e di scambiare dati aggiornati fra server centrali e client del proprio palmare o smart phone o applicazione mobile di processo.

In questo mercato avanzano ulteriori cambiamenti interessanti: la posta elettronica sta diventando un normale componente di servizio a supporto della mobilità, gli standard aperti vengono considerati come i più adatti alla produzione sia dai costruttori di dispositivi che dagli operatori mobili. Gli operatori tradizionali che gestiscono questi mercati, come RIM

e Microsoft, i quali utilizzano standard proprietari ad alto costo e quindi riservati ai dipendenti delle grandi imprese, non sono in grado di catturare la domanda crescente del mercato delle piccole imprese e del consumatore. Nonostante ciò, ad esempio RIM è cresciuta dai 40 milioni di dollari nel 2003, ai 145 nel 2004, per esplodere verso i 630 milioni di dollari nel 2005.

La domanda continua a crescere sia sul lato business che su quello consumer: in Francia, Germania e Inghilterra in media il 52% dei dipendenti delle imprese e il 47% dei consumatori sono favorevoli ad avere accesso alle e-mail sul proprio cellulare.

3. OFFERTA FUNAMBOL E DIFFERENZIALI COMPETITIVI

Fabrizio Capobianco ha intuito che la convergenza delle tecnologie digitali si sviluppa non tanto a livello dei dispositivi che proliferano nelle mani degli utenti finali, quanto a livello delle informazioni che dovranno in futuro essere sempre più centralizzate per essere consultate e utilizzate attraverso canali e terminali diversi. Sulla base di questa idea nasce il prodotto Sync4j di Funambol, costituito da una piattaforma per lo sviluppo e l'erogazione di applicativi mobili per la messaggistica e il PIM (*Personal Information Manager*), il quale permette agli sviluppatori wireless di realizzare applicazioni sul paradigma "collegato-ogni tanto"; l'applicazione lavora fuori linea e i dati vengono sincronizzati quando l'apparecchio è di nuovo in linea.

I prodotti Funambol SyncML, PIM Sync spingono verso una posta elettronica basata sugli standard aperti, supportano una varietà di sistemi dati di back-end e abilitano l'ecosistema mobile open source.

La piattaforma Funambol, rappresentata nella figura 1, offre quattro tipi di servizi che implementano la strategia "dual licence", cioè sia in versione libera scaricabile da Internet che in versione proprietaria a pagamento, arricchita da funzioni ulteriori e assistita. I servizi della piattaforma Funambol sono:

1. il push e-mail: costituisce il principale mezzo di comunicazione con il cliente finale per tutti gli operatori mobili; ha la caratteristica di poter essere inviato a milioni di utenti a costi trascurabili;
2. la sincronizzazione dei contatti e delle agende: è il servizio a valore aggiunto messo a disposizione sia della clientela consumer che degli addetti delle imprese con diverse modalità di offerta; rappresenta la soluzione più efficace per far sì che i server centrali dell'impresa e/o dell'Operatore mobile siano costantemente allineati rispetto agli ultimi aggiornamenti che ciascuno dei milioni di utenti sviluppa nell'ambito della propria attività remota;
3. la piattaforma di sviluppo delle applicazioni mobili: rappresenta la base di innovazione permanente della comunità degli specialisti di open source sia indipendenti che di quelli che lavorano nelle imprese. Questa è la soluzione che ha accreditato Funambol nell'area dell'open source e le ha permesso di ampliare la comunità dei contributori alla innovazione e all'uso dei suoi tool;
4. la gestione dei dispositivi, che garantisce la compatibilità delle applicazioni Funambol per tutti i dispositivi in commercio attuali e futuri. Questa è la principale leva che ha permesso a Funambol di penetrare rapidamente nel mercato dei dispositivi mobili del consumer e accreditarsi quindi anche presso gli operatori mobili e le imprese più innovative.

L'offerta che Funambol ha costruito intorno alla



FIGURA 1
La piattaforma dei servizi Funambol

propria piattaforma è articolata su quattro target diversi:

- edizione per la comunità degli specialisti in open source;
- edizione per reti, per le imprese: tale offerta rappresenta la versione commerciale di quella per la comunità e comprende supporti come la notificazione dell'aggiornamento di prodotto e il portale di supporto al cliente; il prezzo parte da una base di 930 \$ e viene venduta direttamente sul portale Funambol;
- edizione per operatori di telefonia mobile (carrier): questa offerta comprende un portale per dispositivi mobili, la scalabilità dell'offerta, l'alta disponibilità e la tolleranza ai guasti; supporta inoltre le infrastrutture commerciali dei carrier come i server applicativi, i database ecc.;
- edizione per i consumatori, attraverso un portale Funambol, che ospita le realizzazioni gratuite sviluppate nell'ambito della edizione per i carrier (in corso di lancio nel primo trimestre del 2007).

Il prodotto base Sync4j di Funambol, come indicato nella figura 2, è stato ormai prelevato in oltre 700.000 copie alla fine del 2006.

Illuminante diventa l'analisi condotta da Funambol sulle principali funzioni d'uso offerte con la propria piattaforma rispetto a quelle offerte dai principali concorrenti.

I risultati del confronto riguardano la disponibilità di 9 funzioni d'uso da parte di Funambol e di 8 suoi concorrenti, le cui offerte vengono focalizzate rispetto ai tre segmenti di clientela consumer, piccole e medie imprese e grandi imprese.

Le funzioni d'uso considerate sono:

- 1. compatibilità dei dispositivi:** RIM e Microsoft non forniscono alcuna compatibilità con dispositivi privi di sistema operativo proprietario; Critical Path, OZ, Seven, Visto, Nokia e Good hanno una compatibilità parziale; Funambol è l'unico operatore che offre una compatibilità totale con tutti i dispositivi;
- 2. utilizzo di standard:** di nuovo Funambol è l'unico operatore che si appoggia a standard, mentre tutti gli altri operatori li considerano parzialmente o per nulla;
- 3. open source:** nessun operatore, a parte Funambol, adotta l'open source come modalità di sviluppo delle applicazioni e di compatibilizzazione dei dispositivi;

4. contatti e agende: tutti gli operatori, salvo Critical Path e OZ, offrono la gestione dei contatti e l'aggiornamento delle agende;

5. supporto alle email WAP: Funambol, Seven, Visto e Nokia offrono questo supporto, mentre gli altri no;

6. push e-mail: questa funzione d'uso è disponibile per tutti i concorrenti;

7. gestione dei dispositivi: solo Funambol, Nokia e Good offrono la gestione completa dei dispositivi, mentre Microsoft e RIM la offrono parzialmente e gli altri per nulla;

8. applicazioni per il cliente: Funambol offre questo sviluppo parzialmente insieme a Visto, RIM, Microsoft e Nokia, mentre OZ e Good hanno una offerta completa;

9. soluzioni ospitate: offrono questa funzione Critical Path, Seven, RIM e Nokia, mentre Funambol con OZ, Visto e Good la offre parzialmente.

In sintesi, l'analisi della piattaforma Funambol rispetto ai concorrenti mette in evidenza alcuni aspetti di differenziazione molto importanti:

- Funambol è l'unico operatore ad offrire la sua piattaforma in ambiente open source;
- è inoltre l'operatore che più di tutti gli altri

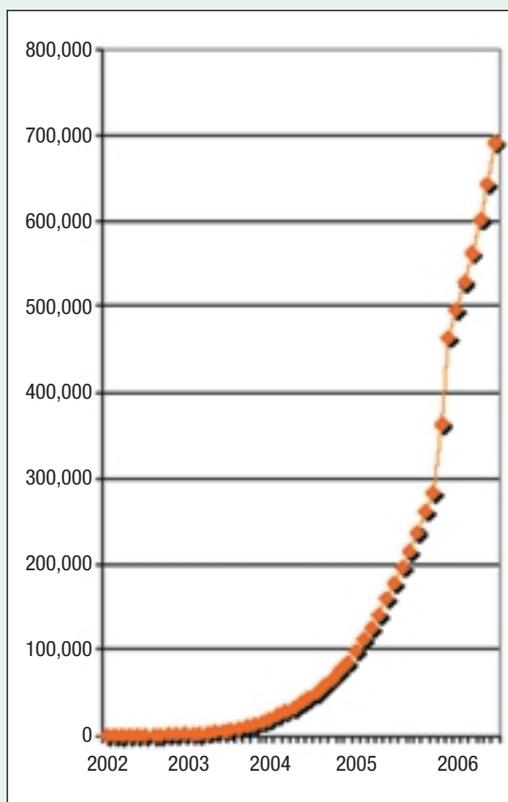


FIGURA 2

Copie prodotte Funambol scaricate

FIGURA 3
Funambol e la sua concorrenza

				Focus PMI		Focus Grandi Imprese			
	II	7-11	OZ	SEVEN	WET	FTW	Microsoft	NOKIA	Google
Device compatibility	●	●	●	●	●	○	○	●	●
Standards based	●	●	●	●	●	○	○	●	●
Open Source	●	○	○	○	○	○	○	○	○
Contacts & Calendar	●	○	○	●	●	●	●	●	●
WAP Email Support	●	○	○	●	●	○	○	●	●
Push Email	●	●	●	●	●	●	●	●	●
Device Management	●	○	○	○	○	●	●	●	●
Client Applications	●	○	●	○	●	●	●	●	●
Hosted Solutions	●	●	●	●	●	●	○	●	●

ha sviluppato un approccio sistematico alla compatibilizzazione dei dispositivi mobili con il proprio software e che gestisce il dispositivo in tutti i suoi aspetti; con questo approccio Funambol “vede” centinaia di milioni di dispositivi, mentre i suoi concorrenti possono lavorare solo con i dispositivi che hanno a bordo il proprio software.

Funambol si qualifica quindi come l'unico operatore in grado di servire sia il consumatore finale, sia la rete dei dipendenti di una grande, media o piccola impresa o organizzazione pubblica, sia un operatore di telefonia mobile.

La figura 3 rappresenta il risultato dell'analisi dei differenziali competitivi di Funambol rispetto ai principali concorrenti.

4. MODELLO DI BUSINESS DI FUNAMBOL

Inizialmente Funambol ha adottato un modello di business classico per le software house: offrire il proprio software direttamente ai potenziali clienti, selezionati fra grandi imprese e operatori di telefonia mobile. Il modello era basato sul fatto che una mobilità crescente, soprattutto in una grande impresa in cui lavorano sul campo numerosi operatori tecnici e commerciali, produce un bisogno crescente di sincronizzazione dei dati a cui accedono o aggiornano una molteplicità di strumenti di mo-

bilità e di server aziendali a livello corporate. Questo modello di business non produceva risultati adeguati, oltre che per il costo del ciclo di vendita B-to-B (necessità di intermediazione da parte di un system integrator, lunghezza del ciclo, costo delle risorse commerciali), anche per l'inadeguatezza del modello rispetto al mercato dell'open source.

Infatti, mentre le soluzioni proprietarie richiedono che il cliente venga individuato e raggiunto dal fornitore, le soluzioni open source sono generalmente scoperte e raggiunte dai clienti: adottare un approccio aggressivo sul mercato significa rinunciare ad uno dei maggiori punti di forza dell'open software.

Funambol si è trovata a sperimentare quanto segue: ogni volta che il numero degli scarichi di Sync4j aumentava, aumentavano anche le richieste da parte di imprese che chiedevano di poter utilizzare il prodotto dopo averne accuratamente testato il software; le imprese che si dichiaravano interessate erano comunque fra le più alte nella graduatoria di Fortune e comprendevano molti carrier di telefonia mobile.

Sulla base delle considerazioni indicate, Funambol ha deciso di distribuire il prodotto v3 con l'approccio dual license.

L'analisi delle fasi del ciclo di vendita rilevate da Funambol nel nuovo modello di business mostra che il suo approccio di vendita si è mo-

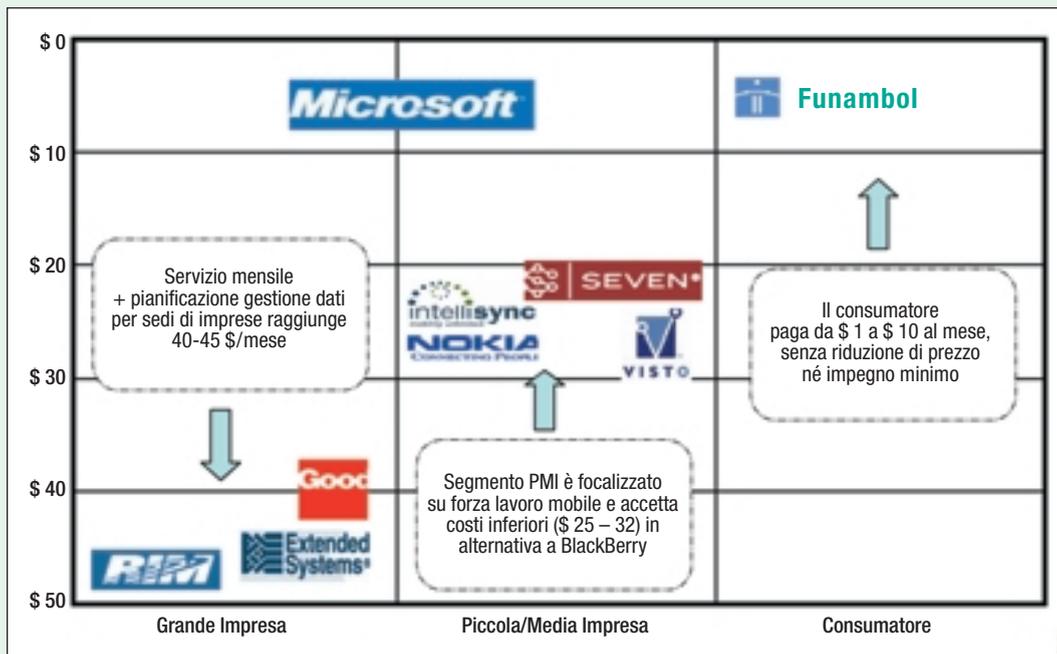


FIGURA 4
Segmentazione del mercato

dificato dall'iniziale "spinto dal venditore" a quello attuale "tirato dall'utente": le trattative commerciali non nascono da aggressivi sforzi commerciali, ma derivano dai contatti spontanei dei potenziali interessati; l'obiettivo di Funambol non è più quello di cercare nuovi clienti, ma quello di servire e supportare con la massima cura i potenziali nuovi clienti che si presentano. La strategia di prodotto e di marketing è diventata quella di alimentare la parte più vicina al mercato della catena del valore di Sync4j: sono gli scarichi del prodotto che devono essere seguiti. Complessivamente il ciclo di vendita è diventato molto più rapido (mediamente di tre mesi o meno) e si basa essenzialmente su una televendita accompagnata da qualche visita, per cui i costi di acquisizione del cliente sono molto ridotti.

L'applicazione del nuovo modello di business basato sul dual license ha portato ad una segmentazione del mercato in cui Funambol occupa la quota più ampia, relativa al mercato consumer, ma con la possibilità di andare a toccare anche il mercato delle imprese grandi, piccole e medie, come indicato nella figura 4.

5. ANALISI DELLA STRUTTURA DI FUNAMBOL

La proprietà delle quote di Funambol è dei fondatori, di investitori Americani ed Europei,

e della trentina di dipendenti. Non è possibile conoscere il fatturato dell'azienda, perché ancora non quotata in borsa, ma si può comunque constatare che si tratta di una società internazionale con clienti internazionali e una struttura operativa particolare rispetto alle start up italiane: l'azienda opera su due sedi, quella marketing e di direzione ha sede in California, mentre la sede della Ricerca & Sviluppo e della produzione è ancora a Pavia. Secondo Fabrizio Capobianco i ricercatori italiani sono preparati tanto quanto quelli americani o indiani, non sono estremamente costosi e sono sicuramente affidabili. Le ragioni di una scelta di questo genere dipendono da una parte dalla necessità di accedere ai capitali di rischio attraverso il venture capital americano, perché quello italiano è molto limitato, dall'altra dalla constatazione che il livello di preparazione degli ingegneri italiani nel campo del software per le applicazioni mobili non ha nulla da invidiare a quelli dei paesi considerati più avanzati.

6. CONCLUSIONI

Funambol introduce sul mercato delle applicazioni mobili un approccio radicalmente innovativo che riesce a combinare sia le opportunità offerte dall'approccio open source che quelle offerte dall'approccio di fornitore clas-

sico. Il successo di Funambol si basa sui seguenti fattori:

- mettere a disposizione del mass market la soluzione a costi più accessibili, perché realizzata in open software, accompagnata dalla più ampia compatibilità con i dispositivi attuali;
- fornire ai clienti una strategica libertà di scelta:
 - per gli operatori di telefonia mobile con soluzioni white label;
 - per gli utenti finali, con la soluzione open

source, sotto controllo dei clienti, lasciando la possibilità di connettere qualunque hardware o software;

- utilizzare il contributo dei prodotti delle terze parti e di una ampia base di talenti per abbattere i costi;
- rendere più veloce lo sviluppo e la disponibilità di nuove funzioni con un ciclo di innovazione più rapido sul mercato mobile e con qualità più elevata.

ROBERTO BELLINI è docente di Marketing e Gestione della Relazione con il cliente nell'ambito del MIP - Politecnico di Milano, con una focalizzazione sulla innovazione nelle reti di imprese. Presiede la Sezione AICA di Milano ed è responsabile per Aica del Cantiere dei Mestieri ICT.

E-mail: roberto.bellini@polimi.it

CHIARA FRANCALANCI è professore associato di Sistemi Informativi al Politecnico di Milano. Ha scritto numerosi articoli sulla progettazione e sul valore economico delle tecnologie informatiche, svolto attività di ricerca e consulenza nel settore finanziario e manifatturiero sia in Italia sia presso la Harvard Business School ed è editor del Journal of Information Technology.

E-mail: francala@elet.polimi.it



ICT E DIRITTO

Rubrica a cura di

Antonio Piva, David D'Agostini

Scopo di questa rubrica è di illustrare al lettore, in brevi articoli, le tematiche giuridiche più significative del settore ICT: dalla tutela del *domain name* al *copyright* nella rete, dalle licenze software alla *privacy* nell'era digitale. Ogni numero tratterà un argomento, inquadrandolo nel contesto normativo e focalizzandone gli aspetti di informatica giuridica.

IL PHISHING

David D'Agostini, Antonio Piva

1. INTRODUZIONE

Con il termine “*phishing*” (dal verbo inglese “fish”, a indicare il fatto che le vittime vengono prese all'amo lanciando delle esche nel mare di utenti di internet) si fa riferimento a un'attività truffaldina che, sfruttando il *social engineering*¹, mira a ottenere l'accesso al conto corrente del malcapitato, dal quale vengono prelevate somme di denaro generalmente trasferite all'estero.

Il phishing è un vero e proprio attacco, condotto soprattutto per mezzo della posta elettronica (Figura 1), che si realizza per esempio nel seguente modo:

1. il truffatore (*phisher*) invia un'e-mail che simula nella grafica e nel contenuto quella di un soggetto noto al destinatario (per esempio il suo istituto di credito);
2. l'e-mail contiene quasi sempre avvisi di problemi verificatisi con il proprio conto corrente o richieste di verifiche dell'account medesimo;
3. il messaggio invita la vittima a seguire un collegamento ipertestuale che, tuttavia, non porta al sito web ufficiale, bensì a una copia apparentemente simile al medesimo messaggio on line dal phisher;
4. quest'ultimo, pertanto, carpisce le credenziali di autenticazione² del correntista e le utilizza per acquistare beni ovvero per bonificare somme di denaro;

5. spesso tale denaro transita sul conto di un terzo che (a conoscenza o meno della provenienza illecita) si presta a trasferirlo all'estero, dietro compenso in percentuale.

Ciò che contraddistingue questo fenomeno sono, inoltre, le modalità con le quali si compie la fase successiva e necessaria del reimpiego del denaro, dei profitti e dei proventi anche grazie all'interposizione di società fittizie intestate a prestanome utilizzate nel “lavaggio” del denaro.

2. LE FASI DEL PHISHING

In assenza di una fattispecie unitaria a cui ricondurre il phishing e di una definizione universale, risulta comunque possibile individuare il *modus operandi* dei truffatori che agiscono attraverso fasi successive.

□ **La preparazione dell'attacco:** gli ideatori determinano le modalità con cui propagare l'attacco in rete, si avvalgono dei supporti tecnologici registrando domini e predisponendo i server su cui convogliare le informazioni riservate fraudolentemente carpite e da cui trasmettere richieste o interferire nelle comunicazioni tra i soggetti coinvolti.

□ **La propagazione dell'attacco:** a seconda delle modalità decise, vengono innescate le procedure rivolte a stabilire un contatto diretto o indiretto con la vittima, al fine di rubarne l'identità ottenendo le credenziali d'autenticazione. In questa

¹ Descrive l'insieme delle tecniche di persuasione e/o di inganno messe in campo per accedere a un sistema informatico; la tematica è stata trattata nella presente rubrica nel numero di giugno 2004 di Mondo Digitale.

² costituite per esempio da Username e Password.

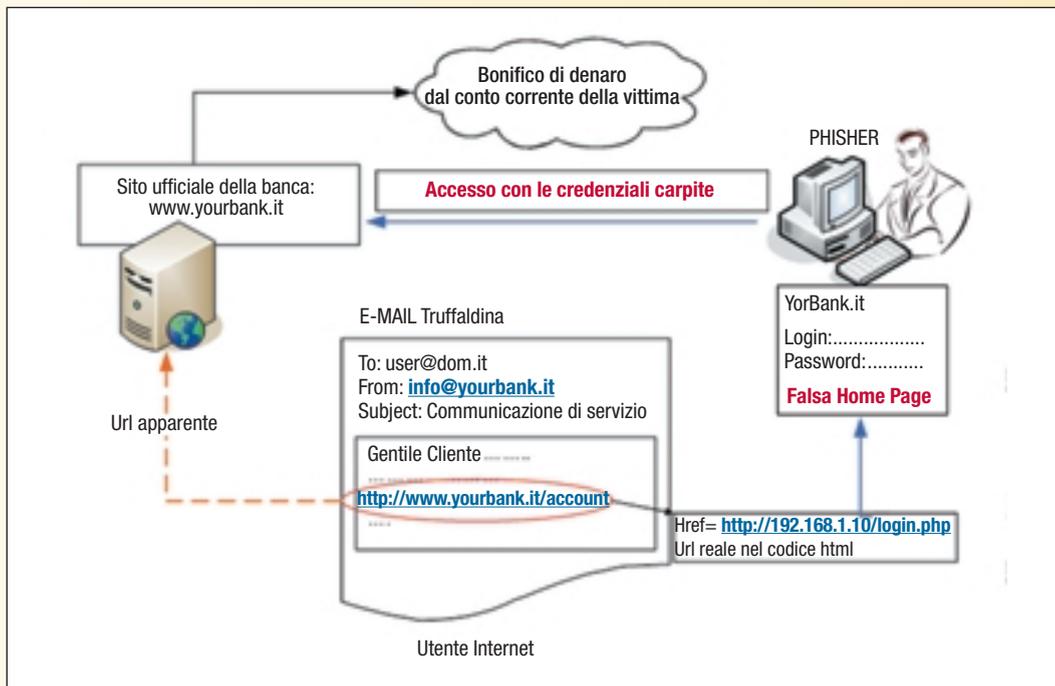


FIGURA 1
L'attacco condotto dal PHISHER

fase il phisher invia un'e-mail, qualificandosi come una banca, una società finanziaria, un ente, o più generalmente, nel caso in cui il phisher abbia scelto in modo mirato le proprie vittime, un soggetto che, a seconda della natura dei rapporti tipicamente intrattenuti dalla vittima, appaia a quest'ultima legittimata a procedere a richieste inerenti dati e chiavi di accesso. Il phisher può operare un invio casuale delle mail/esche a una svariata moltitudine di soggetti oppure decidere il target di identità utile ai propri fini. Nel primo caso lo strumento della posta elettronica è usualmente utilizzato dai truffatori con la logica dello spamming, attraverso l'invio di messaggi³. La mail inviata a un utente distratto e a volte ingenuo, chiede allo stesso di compiere un'attività che determinerà la rivelazione delle informazioni riservate attinenti a user id, password e altre chiavi necessarie all'accesso o all'utilizzo a opera del phisher. L'azione della vittima viene pertanto provocata attraverso il riferimento a situazioni che catturano l'attenzione dello stesso e si caratterizzano per la necessità e/o l'urgenza: notifica di problemi e disagi inerenti i rapporti intrattenuti dal soggetto medesimo con l'ente o l'istituto finanziario, necessità di procedere a modifiche della password o aggiornamento dei

³ Tanto più efficaci quanto più credibili, circostanziati e affidabili.

dati, in risposta addirittura a possibili frodi e attacchi informatici denunciati dalla banca o istituto di credito; l'offerta di affari ovvero di nuovi servizi a termine validi solo in caso di immediata attivazione dell'utente.

❑ **L'azione della vittima:** la vittima, convinta o semplicemente distratta, risponde all'attacco seguendo il link fornito dal phisher. Conscio della caratteristica del client di posta elettronica, di visualizzazione dei messaggi creati in HTML, il phisher ha inserito nel corpo dell'e-mail un link che, anche se esteriormente riporta un indirizzo web autentico, nascosto nel codice html (nel tag href), è stato impostato un collegamento ipertestuale che punta al falso sito web o a del codice da scaricare. In tal modo la vittima viene diretta al sito ufficiale dell'intermediario a cui vengono abilmente apposte delle finestre di pop up oppure a siti creati dal phisher che possono rappresentare il clone di quello ufficiale attraverso l'adozione di tecniche, sempre più sofisticate, rivolte all'occultamento dell'indirizzo IP (cosiddetto *IP spoofing*). Non da ultimo, la richiesta dei dati può essere contestuale e insita alla mail.

❑ **Il furto d'identità:** il furto d'identità si compie nel momento in cui l'utente fornisce i propri dati al phisher digitandoli in un messaggio di risposta o nel sito internet clone oppure per mezzo dei programmi infetti attivati inconsapevolmente dal soggetto all'interno del proprio

computer; può infatti accadere che le informazioni vengano prelevate con l'ausilio di keylogger e web trojan⁴, precedentemente scaricati e installati dall'ignaro utente.

□ **La consumazione della frode e della truffa:** il momento in cui si consuma il reato è quello in cui avviene da un lato l'indebita percezione del profitto (vale a dire l'incasso del denaro) e dall'altro lato la realizzazione del danno per l'utente internet: ciò si realizza nel momento in cui il phisher utilizza le credenziali fraudolentemente o abusivamente carpite e ottiene il trasferimento del capitale a mezzo dell'istituto di credito con conseguente sottrazione e impoverimento del patrimonio del correntista.

□ **La dispersione e il reimpiego dei profitti del reato:** la condotta fraudolenta del phisher non si esaurisce nel momento consistente nella violazione del patrimonio del correntista; subentra infatti una fase di reclutamento di soggetti compiacenti e non, che si rendono disponibili al transito di capitali attraverso i propri conti correnti. Si parla di transito, in quanto, i proventi convogliati su conti correnti bancari e postali vengono immediatamente ritirati attraverso prelievi in contanti e trasferiti per il mezzo di agenzie di money transfer o bonifici on line a conti intestati a società fittizie oppure a persone residenti all'estero. Minimo comune denominatore delle modalità scelte per il riciclaggio dei proventi del phishing è costituito dalla combinazione di internet e della transnazionalità delle operazioni, dalla creazione di società intestate a prestanome, dallo sfruttamento di regimi tipici dei cosiddetti paradisi fiscali⁵.

3. QUALI REATI?

Sotto il profilo giuridico, l'assenza di una norma *ad hoc* rende difficoltoso ricondurre le condotte che costituiscono il phishing a una singola ipotesi di reato; tuttavia, sulla scorta dell'analisi delle fasi del fenomeno, si possono determinare le violazioni configurabili secondo l'ordinamento giuridico penale.

3.1. Il trattamento illecito di dati

Nel momento in cui il phisher ottiene abusivamente le credenziali di autenticazione dell'utente, senza il legittimo consenso di quest'ultimo, è configurabile l'ipotesi di reato di trattamento illecito di dati personali, prevista dall'art 167 del Codice in materia di protezione dei dati personali, approvato con il d.lgs. 30 giugno 2003, n. 196.

L'acquisizione fraudolenta dei dati personali, la raccolta delle informazioni inviate ai server dei siti web compromessi dal phisher e il loro successivo utilizzo per accedere attraverso l'identità violata del titolare dei dati, integra una serie di trattamenti dei dati personali⁶. La norma dell'art 167 punisce, tra gli altri, il trattamento dei dati personali posto in essere in violazione del consenso da parte dell'interessato. Infatti il phishing determina un accesso ai dati della vittima, senza che la stessa dia un consenso espresso e libero: Il phisher deliberatamente acquisisce dati che verranno trasmessi, raccolti, registrati, organizzati e utilizzati, al fine di trarre un illegittimo profitto con danno per la vittima, consistente per lo più nell'impoverimento patrimoniale.

3.2. La truffa

La disposizione dell'art. 640 c.p., nel prevedere il reato di truffa, punisce chi con artifizii e raggiri induca qualcuno in errore e procura a sé o ad altri un ingiusto profitto con altrui danno.

Il modello della truffa è quello che meglio si attaglia alla fase in cui il phisher procede a "lanciare l'esca/mail" all'utente. Infatti, salvo il caso di spamming, molto spesso il target delle vittime viene scelto ed in questo caso, per aumentare la credibilità, la mail sarà tanto più circostanziata e convincente quanto più il phisher conoscerà il soggetto vittima della propria offensiva.

L'utente che riceve l'e-mail fraudolenta viene indotto in errore e l'errore consiste nel ritenere di intrattenere dei rapporti con un determinato soggetto, di cui l'autore del reato ha assunto le sembianze, ragione per cui le azioni della per-

⁴ Il keylogger è uno strumento in grado di intercettare tutto ciò che un utente digita sulla tastiera del proprio computer. Spesso i keylogger sono trasportati ed installati nel computer da virus denominati worm o trojan ricevuti tramite Internet ed hanno in genere lo scopo di intercettare password, credenziali di accesso e/o numeri di carte di credito.

⁵ Caratterizzati dalle difficoltà inerenti alla cooperazione di polizia e giudiziaria al fine della repressione delle frodi.

⁶ Intesi come qualunque operazione o complesso di operazioni, effettuati anche senza l'ausilio di strumenti elettronici, concernenti la raccolta, la registrazione, l'organizzazione, la conservazione, la consultazione, l'elaborazione, la modificazione, la collezione, l'estrazione, il raffronto, l'utilizzo, l'interconnessione, il blocco, la comunicazione, la diffusione, la cancellazione e la distruzione di dati.

sona offesa non sono volontariamente e consapevolmente determinate.

Perché possa ritenersi avverata l'ipotesi di reato di cui all'art 640 c.p., è necessario che si verifichi l'induzione in errore della vittima. Colui che sporge denuncia per questi fatti è spesso proprio l'utente internet comune che, nonostante tema i pericoli della rete, essendo inesperto spesso non valuta in concreto le potenzialità che lo strumento informatico offre e non conosce i reali rischi. Pertanto anche condotte poco raffinate di artificio e raggiro⁷ possono indurre in errore l'utente medio e il phisher ne tiene conto nella scelta delle proprie vittime. Nel caso del phishing l'artificio avviene, per esempio, attraverso l'indirizzamento dell'utente al sito internet dell'istituto bancario su cui vengono scientemente applicate finestre di pop up, nelle quali la vittima fornisce le credenziali richieste; oppure cliccando su un collegamento ipertestuale (link) indicato nell'e-mail, l'utente viene mandato verso il sito web clonato o offuscato attraverso il mascheramento dell'indirizzo IP.

Il raggiro, in base alla casistica illustrata, risiede nella fase della "call in action" dell'utente attraverso il messaggio e-mail in cui vengono usate le argomentazioni che permettono la prosecuzione della violazione nel momento successivo dell'artificio. Il raggiro può essere contestuale all'artificio qualora vengano riprodotte nel messaggio i caratteri tipici e dei segni distintivi del soggetto di cui è simulata l'identità.

3.3. La frode informatica

Un'analisi del fenomeno del phishing negli ordinamenti stranieri permette di riscontrare che lo stesso è definito internazionalmente come "identity theft" o "fraud"; la sostanza del fenomeno, al di là dei tentativi di ricondurlo a qualche fattispecie giuridica dell'ordinamento italiano, è proprio questa: la frode susseguente a un furto di identità, momento centrale dell'intera vicenda. Per questi motivi, risulta calzante il riferimento al "ciclo di vita del phishing", raffigurabile graficamente con un'immagine circolare a

spirale, per cui il fenomeno si autoalimenta, segue un circolo, ma il cerchio non si chiude e passa attraverso le fasi della captazione illecita di informazioni riservate e l'utilizzo delle stesse⁸.

Quale illecito è configurabile nel momento in cui i dati carpiti con l'inganno o in modo abusivo vengono utilizzati in un altro sistema informatico (quello della banca)?

In questo caso non si può fare riferimento alla truffa, in quanto il sistema informatico non può essere né raggirato, né verso lo stesso può avvenire un artificio: infatti tali concetti sono riferibili alle persone fisiche, non alle macchine.

Si può, allora, richiamare la norma di cui all'art. 640 ter c.p. la quale sanziona chi, alterando in qualsiasi modo il funzionamento di un sistema informatico o telematico o intervenendo senza diritto con qualsiasi modalità su dati, informazioni o programmi contenuti in un sistema informatico o telematico, procura a sé o ad altri un ingiusto profitto.

Si precisa che per sistemi informatici il legislatore intende i "sistemi di scrittura o di tale automazione d'ufficio ad uso individuale o particolare, nonché quelli di elaborazione dati"; i sistemi telematici invece sono costituiti da "reti di telecomunicazione sia pubbliche che private, locali o geografiche, nazionali o internazionali, operanti da e per il nostro Paese, e ogni altra loro componente (software, dati, informazioni, flussi di comunicazione, messaggi); il software, sia esso di base, di supporto, generalizzato o applicativo, inglobando nel concetto qualunque programma informatico realizzato dal costruttore dell'hardware, da strutture di produzione ad hoc, da singoli utenti".

Tale norma punisce l'alterazione del sistema informatico e telematico e l'intervento sul dato informatico, avvenuta in qualsiasi modo, pertanto la frode informatica a differenza della truffa, si applica a quelle condotte del phisher che, sebbene fraudolente, non comportano artifici o raggiri. Con riferimento a quest'ultimo aspetto, nel caso di frode informatica lo stesso legislatore assimila alle condotte dell'artificio e del raggiro, l'alterazione del sistema informatico e ritiene per ciò

⁷ Da un punto di vista giuridico l'artificio consiste nella manipolazione, trasfigurazione della realtà esterna, provocata mediante la simulazione di circostanze inesistenti o la dissimulazione di circostanze esistenti; mentre il raggiro è un'attività simulatrice sostenuta da parole o argomentazioni atte a far scambiare il falso per il vero.

⁸ In sostanza il fenomeno appare continuativo: può avvenire un accesso abusivo tramite la diffusione di virus; viene portato a compimento il furto dell'identità attraverso modalità fraudolente, le credenziali offerte vengono utilizzate attraverso lo stesso soggetto di cui il phisher ha assunto le vesti, un ulteriore accesso abusivo e un ulteriore sistema di frode.

integrata la fattispecie illecita, senza che sia necessario dimostrare l'induzione in errore.

In particolare, l'art. 640 ter c.p. disciplina queste due possibilità:

- l'alterazione, in qualsiasi modo, del funzionamento del sistema informatico e telematico: risulta violata l'integrità del sistema di informazione;
 - l'intervento, senza diritto e con qualsiasi modalità su dati, informazioni, programmi contenuti in un sistema informatico o telematico: risulta violata la sicurezza del dato informatico.
- Il Legislatore ha voluto determinare una distinzione tra le due ipotesi delle quali la prima presuppone un intervento più incisivo, che coinvolge tutto il sistema determinandone un'alterazione significativa⁹; la seconda invece fa pensare a casi in cui viene isolato un elemento del sistema, un dato informatico, un determinato programma, senza coinvolgere necessariamente tutto il sistema.

4. CONCLUSIONI

Come più volte sottolineato, non esiste un'unica tipologia di *phishing* in quanto vengono continuamente sviluppate nuove e più subdole forme di inganno *on line*.

Per tutelarsi contro questi rischi, possono essere utilizzati appositi programmi¹⁰ che bloccano email o collegamenti sospetti, confrontando gli indirizzi reali dei mittenti con le black list¹¹ presenti in internet. Tali sistemi, in taluni casi, possono essere comunque elusi e le black list non sempre sono perfettamente aggiornate, quindi gli esperti di sicurezza informatica consigliano in particolare di non rispondere a richieste di informazioni personali ricevute tramite posta elettronica, nonché di visitare i siti internet digitando l'indirizzo direttamente nella barra degli indirizzi e con l'accortezza di verificare che il sito utilizzi sistemi di crittografia (per cifrare la password)¹². Risulta, inoltre, prudente esaminare con regolarità gli estratti conto bancari e i report della carta di credito al fine di

riscontrare eventuali anomalie e denunciare prontamente i sospetti movimenti illeciti.

In definitiva, considerate le difficoltà di natura investigativa e di ordine pratico, mai come per contrastare questo fenomeno la prevenzione risulta l'arma migliore. Di ciò ne sono consapevoli anche gli intermediari finanziari direttamente coinvolti, *in primis* le banche, che negli ultimi tempi hanno attivato numerose campagne d'informazione e servizi tecnologici (come gli sms che avvisano il cliente in caso di bonifici) volti ad arginare le frodi telematiche.

Bibliografia

- [1] Phishing Dhs Report: *Online Identity Theft: Phishing Technology, Chokepoints and Countermeasures*. Aaron Emigh, Radix Labs, 3 ottobre 2005 reperibile su www.antiphishing.org/Phishing-dhs-report.pdf
- [2] Call in action Report, a publication from the national consumer League, marzo 2006, www.ncl-net.org/news/2006/Final%20NCL%20Phishing%20Report.pdf

ANTONIO PIVA laureato in Scienze dell'Informazione, Vice Presidente dell'ALSI (Associazione Nazionale Laureati in Scienze dell'Informazione ed Informatica) e Presidente della commissione di informatica giuridica. Docente a contratto di diritto dell'informatica all'Università di Udine.

Consulente sistemi informatici e Governo Elettronico nella PA locale, valutatore di sistemi di qualità ISO9000 ed ispettore AICA ECDL.

E-mail: antonio@piva.mobi

DAVID D'AGOSTINI avvocato, ha conseguito il master in informatica giuridica e diritto delle nuove tecnologie, fornisce consulenza e assistenza giudiziale e stragiudiziale in materia di *software*, *privacy* e *sicurezza*, contratti informatici, *e-commerce*, nomi a dominio, computer crimes, firma digitale. Ha rapporti di partnership con società del settore ITC nel Triveneto.

Collabora all'attività di ricerca scientifica dell'Università di Udine e di associazioni culturali.

E-mail: david.dagostini@adriacom.it

⁹ Si pensi ai casi di attacchi di phishing verificatisi nella pratica, per cui l'attacco, raggiunge l'apice in un ristretto lasso temporale determinando il blocco del sistema della banca o dell'istituto finanziario con conseguente paralisi dell'attività.

¹⁰ Si pensi a Netcraft o per utenti di outlook: Delphish: programmi che avvisano l'utente quando arrivano email sospette (inoltre a volte tali email vengono fermate a monte già dall'e-mail server, tramite i filtri appositi, che utilizzano le black list presenti su internet).

¹¹ Le black list sono elenchi di indirizzi di posta elettronica, siti o domini, utilizzati solitamente per lo spam o diffusione di virus o phishing; tali elenchi sono pubblicati su internet e liberamente consultabili, vengono normalmente utilizzati dai server che li consultano ogniqualvolta ricevono una email verificando la presenza o meno del mittente in tali elenchi; se la verifica è positiva viene segnata come email spam o potenzialmente pericolosa.

¹² Di solito per tali collegamenti viene utilizzato il protocollo l'SSL: l'utente si accorge di ciò in quanto l'indirizzo riportato sulla barra del browser, inizia con <https://..> al posto del solito <http://..>; in genere tale situazione è anche visualizzabile dalla lucchetto chiuso, in basso a destra, nel browser stesso.

DENTRO LA SCATOLA

Rubrica a cura di

Fabio A. Schreiber

Dopo aver affrontato negli scorsi anni due argomenti fondanti dell'Informatica – il modo di codificare l'informazione digitale e la concreta possibilità di risolvere problemi mediante gli elaboratori elettronici – con questa terza serie andiamo ad esplorare "Come parlano i calcolatori". La teoria dei linguaggi e la creazione di linguaggi di programmazione hanno accompagnato di pari passo l'evolversi delle architetture di calcolo e di gestione dei dati, permettendo lo sviluppo di applicazioni sempre più complesse, svincolando il programmatore dall'architettura dei sistemi e consentendogli quindi di concentrarsi sull'essenza del problema da risolvere.

Lo sviluppo dell'Informatica distribuita ha comportato la nascita, accanto ai linguaggi per l'interazione tra programmatore e calcolatore, anche di linguaggi per far parlare i calcolatori tra di loro – i protocolli di comunicazione. Inoltre, la necessità di garantire la sicurezza e la privacy delle comunicazioni, ha spinto allo sviluppo di tecniche per "non farsi capire" da terzi, di qui l'applicazione diffusa della crittografia.

Di questo e di altro parleranno le monografie quest'anno, come sempre affidate alla penna (dovrei dire tastiera!) di autori che uniscono una grande autorevolezza scientifica e professionale ad una notevole capacità divulgativa.



Linguaggi Formali per Basi di Dati

Letizia Tanca

1. INTRODUZIONE

Come anticipato in un precedente articolo di questa rubrica¹, le basi di dati relazionali godono della particolarità di fornire, oltre che linguaggi di interrogazione adottati dai vari DBMS, anche linguaggi *formali*, quindi con una sintassi e una semantica ben definite che permettono di condurre ragionamenti formali e prove affidabili. I linguaggi formali più noti e usati sono l'*Algebra Relazionale* e il *Calcolo Relazionale* (in due versioni, dette *Calcolo delle tuple* e *Calcolo dei domini*), il primo ispirato a una notazione di tipo algebrico e il secondo basato sul calcolo dei predicati. I due linguaggi sono perfettamente equivalenti, nel senso che possono rappresentare esattamente lo stesso insieme di interrogazioni. Tale insieme è quello considerato "soddisfacente" per un linguaggio formale di interrogazione dalla comunità delle basi di dati; questo tipo di completezza espres-

siva viene detto "nel senso di Bancilhon-Paredaens".

Un altro linguaggio, con specificità un po' diverse, è stato introdotto e intensamente studiato negli anni '80 e '90; si tratta del linguaggio *Datalog*, basato sul paradigma della programmazione logica. Scopo principale della sua introduzione è il tentativo di estendere il potere espressivo dei linguaggi di interrogazione con la possibilità di esprimere interrogazioni ricorsive, e quindi di permettere anche ai linguaggi di interrogazione di esprimere computazioni; in buona sostanza, si tratta di un tentativo di superare il cosiddetto *impedence mismatch* o disaccoppiamento di impedenza, definito nell'articolo precedente.

Nei prossimi paragrafi descriveremo in modo non troppo formale questi linguaggi, e discuteremo la questione della ricorsione e del potere espressivo di Datalog. Dapprima esamineremo l'*Algebra Relazionale*, che verrà esposta con molti dettagli, anche perchè essa si trova alla base delle tecniche di ottimizzazione delle interrogazioni. Nella sezione successiva verranno discussi i linguaggi basati sulla logica: da una parte, il cal-

¹ Letizia Tanca: Linguaggi per Basi di Dati. *Mondo Digitale*, n. 18, Giugno 2006 p. 65-72.

colo relazionale, nella versione *Calcolo delle tuple*, e dall'altra il Datalog. L'articolo terminerà con una breve discussione sul potere espressivo di questi linguaggi, confrontato anche con quello di SQL, già introdotto nel precedente articolo.

2. L'ALGEBRA RELAZIONALE

L'Algebra Relazionale è un linguaggio funzionale, nel senso che esprime il calcolo effettuato da un programma come una funzione matematica. Su questo linguaggio si fonda il formalismo interno adoperato dai Sistemi di Gestione di Basi di Dati per rappresentare le interrogazioni, e per compiere i primi passi di ottimizzazione. È quindi molto importante comprenderne i fondamenti, perché essa fornisce allo studioso, da una parte, gli strumenti per esprimere una interrogazione in modo efficiente, e, dall'altra, la possibilità di un confronto rigoroso con gli altri formalismi di rappresentazione. Per questo motivo, ci dilungheremo su questo più che sugli altri linguaggi formali, per i quali lavoreremo invece basandoci su esempi.

L'Algebra Relazionale è basata su cinque operatori fondamentali, che si applicano a relazioni e, come previsto per un linguaggio relazionale, restituiscono relazioni. Mediante l'uso di questi cinque operatori si compongono espressioni di complessità crescente.

I cinque operatori fondamentali dell'algebra relazionale sono:

1. la *selezione*, rappresentata dal simbolo σ ,
2. la *proiezione*, rappresentata dal simbolo Π ,
3. il *prodotto cartesiano*, rappresentato dal simbolo \times ,
4. l'*unione* insiemistica, rappresentata dal simbolo \cup , e
5. la *differenza* insiemistica, rappresentata dal simbolo $-$.

Si possono poi ottenere altri operatori definendoli a partire da questi cinque.

I primi due operatori sono unari, nel senso che prendono in input una sola relazione, mentre gli altri tre sono binari, cioè si applicano a due relazioni. Ovviamente al risultato di una operazione si può applicare ancora un'altra operazione, perciò un'operazione binaria può diventare n-aria semplicemente iterando la sua applicazione. Espressioni complesse possono essere composte applicando operatori a formule ottenute usando altri operatori.

Consideriamo per prima la selezione. Riprendiamo, come esempio, la piccola base di dati dell'articolo precedente:

CORSI (CodiceCorso, NomeCorso, Docente, Semestre, N_Crediti)

STUDENTI (MatricolaStudente, Nome, Cognome, Indirizzo, CorsoDiLaurea, Anno)

ESAMI (MatricolaStudente, CodiceCorso, Voto)

e immaginiamo di voler estrarre tutti gli studenti del quarto anno di corso. Occorrerà utilizzare la tabella STUDENTI, ed estrarne tutte le tuple che soddisfano la proprietà che l'anno di corso sia uguale a 4.

Questa interrogazione si esprime in SQL come segue:

```
SELECT MatricolaStudente, Nome, Cognome, Indirizzo, CorsoDiLaurea, Anno
FROM STUDENTI
WHERE ANNO = 4
```

Per fare ciò si può utilizzare una selezione nel seguente modo:

```
 $\sigma_{\text{Anno}=4}$  STUDENTI
```

Il significato di questa espressione è per l'appunto *estrarre tutte le tuple della relazione STUDENTI che soddisfino la proprietà indicata a pedice del simbolo di selezione*. Si noti quindi che il predicato (proprietà) che si trova a pedice del simbolo di selezione esprime la clausola WHERE del linguaggio SQL.

In generale, il risultato di una selezione è una tabella che ha lo stesso schema e un numero di tuple (cardinalità) minore o uguale di quella originaria.

Consideriamo ora la proiezione. Immaginiamo di voler estrarre tutti i nomi e cognomi degli studenti. Occorrerà utilizzare la tabella STUDENTI, ed estrarne le colonne **Nome** e **Cognome**. Questa interrogazione si esprime in SQL come segue:

```
SELECT Nome, Cognome
FROM STUDENTI
```

In algebra relazionale, si usa la proiezione nel seguente modo:

```
 $\Pi_{\text{Nome, Cognome}}$  STUDENTI
```

Il significato di questa espressione è per l'appunto *estrarre le colonne (attributi) della relazione STUDENTI indicate a pedice del simbolo di proiezione*. Si noti quindi che il pedice del simbolo di proiezione esprime la clausola SELECT del linguaggio SQL. Purtroppo, come capita spesso nell'area delle basi di dati, accade che alcuni termini abbiano significato diverso in diversi contesti, e così succede con la parola SELECT che ha due significati diversi (ortogonali!) nei due linguaggi.

In generale, il risultato di una proiezione è una tabella che ha un numero di attributi inferiore, ed eventualmente cardinalità minore o uguale di quella originaria. Questo secondo punto va spiegato brevemente: si immagina, nel nostro esempio, che tra gli studenti elencati in tabella esistano alcuni omonimi. In tal caso, le relative tuple, nella tabella risultante dall'applicazione della proiezione, saranno indistinguibili. La semantica del modello relazionale prevede che le tabelle siano degli insiemi, e perciò privi di duplicati. Eventuali duplicati vanno quindi rimossi, e il numero di tuple della relazione risultante sarà, in generale, minore o uguale di quello della relazione originaria. Si noti che, nel caso tra gli attributi della proiezione compaia la chiave della relazione, evidentemente non vi saranno duplicati e la proiezione non abbasserà la cardinalità originaria.

Applicando il prodotto cartesiano a due tabelle, otteniamo una tabella che ha per tuple tutte le coppie possibili di tuple della prima e della seconda tabella. In generale, il risultato del prodotto cartesiano è una tabella che ha per schema l'unione dei due schemi, e per cardinalità il prodotto delle due cardinalità, delle tabelle in ingresso.

Il prodotto cartesiano, pur essendo un operatore fondamentale, di per sè non riveste grande interesse; infatti, giustapporre righe di tabelle in tutti i modi possibili non appare molto utile. L'interesse di questo operatore consiste, però, nell'operatore derivato che se ne ottiene *associando il prodotto cartesiano con la selezione*. Supponiamo di voler estrarre tutti i dati degli studenti con i loro voti, e consideriamo la corrispondente interrogazione SQL:

```
SELECT STUDENTI.MatricolaStudente, Nome, Cognome, Indirizzo, CorsoDiLaurea, Anno, MatricolaStudente, CodiceCorso, Voto
FROM STUDENTI, ESAMI
WHERE STUDENTI.MatricolaStudente= ESAMI.MatricolaStudente
```

Abbiamo preso tutte le colonne delle due tabelle STUDENTI ed ESAMI, e abbiamo imposto la condizione che la matricola dello studente fosse uguale nelle tuple delle due tabelle. In realtà, ciò corrisponde ad accoppiare in tutti i modi possibili (prodotto cartesiano) le tuple delle due tabelle, e poi a prendere solo quelle che hanno la matricola uguale.

Per fare ciò in Algebra Relazionale, si possono utilizzare il prodotto cartesiano e la selezione nel seguente modo:

σ (STUDENTI \times ESAMI)

STUDENTI.MatricolaStudente=
ESAMI.MatricolaStudente

Esiste però un altro modo, più compatto, di esprimere la stessa interrogazione, ed è adoperando l'operatore *join* (\bowtie), definito appunto come composizione di selezione e prodotto cartesiano:

σ (STUDENTI \bowtie ESAMI)

STUDENTI.MatricolaStudente=
ESAMI.MatricolaStudente

Il significato di questa espressione è esattamente lo stesso dell'espressione precedente, ma la notazione risulta più compatta.

Cerchiamo ora di mettere insieme gli operatori visti finora per comporre un'interrogazione più complessa.

L'interrogazione "*Trovare nomi e cognomi degli studenti che hanno ottenuto almeno il voto 28 all'esame di fondamenti di informatica*", che in SQL risulta:

```
SELECT Nome, Cognome
FROM STUDENTI, ESAMI, CORSI
WHERE STUDENTI.MatricolaStudente=ESAMI.MatricolaStudente
AND ESAMI.CodiceCorso=CORSI.CodiceCorso AND
CORSI.NomeCorso="Fondamenti di Informatica"
and ESAMI.Voto>=28.
```

si esprime in algebra relazionale come segue:

Π [σ (STUDENTI \bowtie ESAMI \bowtie CORSI)]

Nome, Cognome CORSI.NomeCorso=
"Fondamenti di Informatica"
AND ESAMI.Voto \geq 28

Come si vede, una interrogazione è quindi formulata mediante una espressione algebrica i cui operandi sono i nomi delle relazioni cui vengono applicati operatori vari. Nell'esempio, alle tre relazioni viene applicato il cosiddetto join

naturale, che utilizza come default il predicato che impone l'uguaglianza degli attributi con lo stesso nome. In pratica, si tratta delle proprietà espresse dalla prima riga della clausola WHERE dell'interrogazione SQL. Applicando poi la selezione si prendono solo le tuple che soddisfano il predicato $CORSI.NomeCorso = \text{"Fondamenti di Informatica"} \wedge ESAMI.Voto \geq 28$ (si ricorda che \wedge rappresenta l'operatore "AND" logico). La proiezione poi opera anch'essa una riduzione, prendendo solo gli attributi Nome e Cognome degli studenti.

Come si diceva all'inizio di questa sezione, l'algebra relazionale ben si presta a ragionamenti sull'ottimizzazione delle interrogazioni; si osserva, infatti, la seguente formulazione, semanticamente equivalente, della stessa interrogazione:

$$\Pi_{\text{Nome,Cognome}} \left[\left(\text{STUDENTI} \bowtie \sigma_{ESAMI.Voto \geq 28} \right) \bowtie \sigma_{CORSI.NomeCorso = \text{"Fondamenti di Informatica"}} \right]$$

Risulta abbastanza intuitivo che le due operazioni di join, in questa nuova formulazione, verranno applicate a relazioni che in generale saranno molto più piccole delle precedenti, e quindi risulteranno meno costose. Ragionamenti analoghi, basati su simili trasformazioni di equivalenza, permettono di adoperare agevolmente l'algebra relazionale come formato interno al DBMS per l'ottimizzazione delle interrogazioni.

Concludiamo con la presentazione degli operatori insiemistici, l'unione e la differenza, dai quali, come succede anche nella teoria degli insiemi classica, si può facilmente definire anche l'operatore di intersezione (\cap). Supponiamo che la nostra base di dati contenga una tabella ulteriore, relativa agli studenti di dottorato:

STUDENTI_DOTTORATO (*MatricolaStudente, Nome, Cognome, Indirizzo, CorsoDiDottorato, Anno*)

e di voler conoscere l'elenco di tutti gli studenti dell'università. Questo risultato si ottiene applicando l'operatore di unione:

STUDENTI \cup STUDENTI_DOTTORATO.

Analogamente, applicando l'operatore di differenza potremmo pensare di ricavare l'elenco degli studenti del corso di laurea che non hanno proseguito col dottorato.

STUDENTI – STUDENTI_DOTTORATO.

Si noti che gli operatori insiemistici sono applicabili solo a coppie di tabelle che hanno schema compatibile, cioè i cui attributi, presi ordinatamente a due a due, hanno lo stesso tipo. In SQL gli operatori insiemistici si esprimono mediante analoghi costrutti, come ad esempio la UNION:

```
SELECT MatricolaStudente, Nome, Cognome, Indirizzo, CorsoDiLaurea, Anno
FROM STUDENTI
UNION
SELECT MatricolaStudente, Nome, Cognome, Indirizzo, CorsoDiDottorato, Anno
FROM STUDENTI_DOTTORATO
```

3. IL CALCOLO RELAZIONALE E IL DATALOG

Nel Calcolo Relazionale le interrogazioni si esprimono specificando l'insieme che soddisfa una certa proprietà logica. Esso si presenta in due varianti, la cui differenza riguarda il tipo di variabile che viene usata. Nel *Calcolo delle Tuple*, la proprietà viene espressa in termini di tuple, nel *Calcolo dei Domini*, le variabili hanno come insieme di definizione non la tupla intera, ma i suoi elementi.

Per motivi di brevità noi presenteremo il Calcolo delle Tuple, che esprime le interrogazioni nella seguente forma standard:

$$\{t \mid p(t)\}$$

Il significato di questa espressione è l'insieme di tutte le tuple t per le quali è vera la proprietà p . $p(t)$ è una formula costruita tramite atomi, e cioè formule logiche elementari come $t_1 \in \text{STUDENTI}$ oppure $t_3[\text{NomeCorso}] = \text{"Fondamenti di Informatica"}$. Vediamo subito un esempio provando a esprimere qualcuna delle interrogazioni elementari già viste in Algebra Relazionale. Consideriamo l'interrogazione:

$$\Pi_{\text{Nome,Cognome}} \sigma_{\text{Anno}=4} \text{STUDENTI}$$

ottenuta componendo le prime due interrogazioni di esempio, che richiede il nome e cognome degli studenti del quarto anno. Questa si

esprimerà, in calcolo delle tuple, nel modo seguente:

$$\{t \mid \exists t1 \in \text{STUDENTI} : (t[\text{Nome}] = t1[\text{Nome}] \wedge t[\text{Cognome}] = t1[\text{Cognome}] \wedge t1[\text{Anno}] = 4)\}.$$

Il risultato è quindi specificato definendo le tuple t costituite da un attributo Nome e un attributo Cognome, il cui valore è uguale a quello assunto dalle tuple $t1$ della relazione STUDENTI che verificano la proprietà che l'anno di corso sia il quarto. Come si può vedere, la specifica dell'interrogazione in calcolo delle tuple ricalca formalmente la sua formulazione in linguaggio naturale, ed è totalmente dichiarativa.

Per un esempio più complicato, vediamo come si esprime la solita interrogazione sugli studenti che hanno preso almeno 28 all'esame di Fondamenti di Informatica:

$$\{t \mid \exists t1 \in \text{STUDENTI}, \exists t2 \in \text{ESAMI}, \exists t3 \in \text{CORSI} \\ (t[\text{Nome}] = t1[\text{Nome}] \wedge t[\text{Cognome}] = t1[\text{Cognome}] \wedge \\ t1[\text{MatricolaStudente}] = t2[\text{MatricolaStudente}] \wedge \\ t2[\text{CodiceCorso}] = t3[\text{CodiceCorso}] \wedge \\ t2[\text{Voto}] \geq 28 \wedge \\ t3[\text{NomeCorso}] = \text{"Fondamenti di Informatica"})\}.$$

Si noti come la condizione di join, cioè l'uguaglianza delle matricole e dei codici dei corsi, sia espressa in questo linguaggio.

L'integrazione della programmazione logica con le basi di dati è stata di grande interesse intorno agli anni '90, e aveva il grande obiettivo di permettere l'espressione di interrogazioni e computazioni all'interno dello stesso paradigma di programmazione. Il fondamento di questo tentativo giace nel fatto che la forma relazionale delle tabelle ha delle forti analogie sia strutturali che semantiche con la forma dei predicati logici. Per esempio, dire che la tupla $\langle \text{AG213}, \text{Giovanni}, \text{Rossi}, \text{Via Pacini 25 MILANO}, \text{Ingegneria Informatica} \rangle$ appartiene alla tabella **STUDENTI** è equivalente ad asserire la verità del seguente predicato con costanti:

STUDENTI (*AG213, Giovanni, Rossi, Via Pacini 25 MILANO, Ingegneria Informatica*)

Un'interrogazione Datalog si esprime mediante una o più regole logiche. Una regola è compo-

sta da un lato sinistro (detto anche *conseguente*) e da un lato destro (*antecedente*). Il lato destro può contenere uno o più predicati, che esprimono condizioni che devono essere soddisfatte per rendere soddisfatto anche il lato sinistro. Le relazioni sono quindi rappresentate da predicati logici, e gli attributi da variabili con lo stesso nome. I predicati possono contenere come argomenti sia delle costanti sia delle variabili: le costanti servono per vincolare l'attributo della corrispondente posizione ad assumere il valore di quella costante, mentre le variabili servono per associare tra loro predicati che le contengono.

Facciamo subito un esempio riformulando in Datalog qualcuna delle interrogazioni precedenti.

L'interrogazione sui nomi e cognomi degli studenti del quarto anno di corso si esprimerebbe in Datalog mediante la seguente regola logica:

StudentiAnziani(Nome, Cognome):-
STUDENTI (__, Nome, Cognome, __, __, 4).

Come si vede, per imporre che l'anno di corso sia 4 abbiamo inserito il valore 4 nella posizione che compete all'anno di corso del predicato STUDENTI. Inoltre, le variabili Nome e Cognome, che pure appaiono nel predicato STUDENTI, vengono "trasferite" al predicato StudentiAnziani, che in pratica raccoglie il risultato della interrogazione. I trattini bassi sono invece dei segnaposto, che indicano posizioni di variabili di cui non si fa uso nella specifica regola.

Le condizioni di uguaglianza tra attributi sono espresse dall'uso della stessa variabile nei vari predicati nel lato destro; consideriamo la solita interrogazione sugli studenti di fondamenti di informatica:

Studentebravo(Nome, Cognome):- STUDENTI (MatricolaStudente, Nome, Cognome, __, __, __),
CORSI (CodiceCorso, NomeCorso, __, __, __),
ESAMI(MatricolaStudente, CodiceCorso, Voto),
Voto >= 28, NomeCorso = "Fondamenti di Informatica".

Qui, la variabile MatricolaStudente si trova sia nel predicato STUDENTI che nel predicato CORSI, realizzando il join, e lo stesso vale per la variabile CodiceCorso e per i predicati CORSI ed ESAMI.

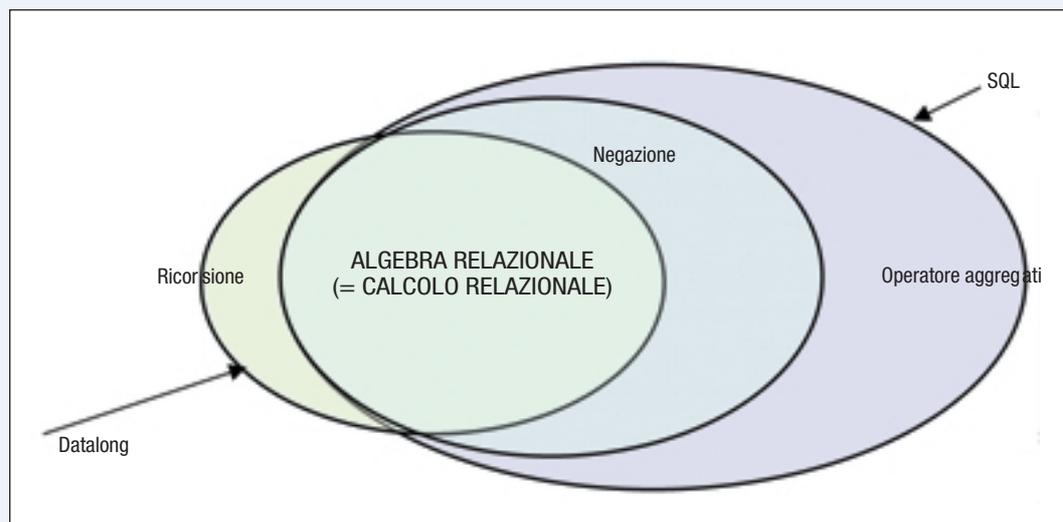


FIGURA 1
Poteri espressivi
dei vari linguaggi

Mediante un tale linguaggio, si riescono ad esprimere computazioni più complesse che in algebra relazionale, come per esempio la computazione ricorsiva espressa dalla seguente coppia di regole:

Antenato(X,Y) :- Genitore(X,Y).
Antenato(X,Y) :- Antenato(X,Z), Genitore(Z,Y).

Partendo da una relazione di base GENITORE, che contiene tutte le coppie genitore-figlio, queste due regole specificano l'insieme di tutte le coppie antenato-discendente. La prima regola costituisce il caso base: "se sei mio genitore sei anche un mio antenato", la seconda regola specifica la computazione ricorsiva: "se sei un antenato di un mio genitore allora sei anche mio antenato".

Si noti che un programma Datalog ha come risultato l'insieme delle tuple che soddisfano le condizioni logiche espresse dalle sue regole, perciò è anch'esso orientato agli insiemi. Il linguaggio Datalog, con le sue varie estensioni, può costituire quindi un'alternativa all'accoppiata SQL/linguaggio di programmazione, in quanto permetterebbe di esprimere uniformemente sia le interrogazioni che le computazioni generiche, e perciò di superare il disaccoppiamento di impedenza. D'altra parte, questa soluzione, molto brillante a livello teorico, comporta una serie di problemi non banali, che non ne hanno permesso l'applicazione pratica nei DBMS commerciali. Vi sono infatti, relativamente ai programmi logici ricorsivi, sia problemi semantici relativi all'uso della

negazione e dei costrutti di aggregazione, sia problemi di ottimizzazione.

4. CONCLUSIONI E CONSIDERAZIONI SUL POTERE ESPRESSIVO DEI LINGUAGGI PER BASI DI DATI

La figura 1 ben riassume le relazioni tra i poteri espressivi dei vari linguaggi trattati in questo articolo. Come si può vedere, l'Algebra Relazionale (equivalente al Calcolo Relazionale) è pienamente contenuta nell'SQL. Infatti, questo ultimo linguaggio offre, dal punto di vista della manipolazione delle tabelle, esattamente lo stesso potere espressivo dell'Algebra e del Calcolo, ma in più offre alcuni costrutti per svolgere semplici computazioni aggregate, come la media, la somma, il minimo e massimo, e la possibilità di contare quante sono le tuple che soddisfano una certa proprietà.

D'altro canto, il Datalog estende l'Algebra Relazionale mediante il costrutto di ricorsione, non presente nelle versioni comuni di SQL², ma non permette di esprimere la negazione (corrispondente, in Algebra, alla differenza) se non a prezzo di alcuni artifici semantici di cui non interessa parlare in questo contesto.

In conclusione, i linguaggi formali per basi di dati forniscono strumenti matematici per trattare con grande rigore questioni di ottimizzazio-

² Gli standard più recenti di SQL offrono qualche forma di ricorsione, ma di fatto tale possibilità non viene introdotta nella maggior parte dei sistemi commerciali.

ne, di potere espressivo e di complessità, difficilmente affrontabili su linguaggi come l'SQL, più verbosi e meno adatti al trattamento mediante strumenti matematici.

Bibliografia

Anche questa volta, piuttosto che elencare le migliaia di riferimenti ai concetti e linguaggi introdotti, indichiamo soltanto alcuni testi classici, da aggiun-

gere a quelli proposti nell'articolo introduttivo, che peraltro già contengono molte delle nozioni viste in questo articolo.

- [1] Abiteboul Serge, Hull Richard, Vianu Victor: *Foundations of Databases*. Addison-Wesley 1995.
- [2] Ceri Stefano, Gottlob Georg, Tanca Letizia: *Logic Programming and Databases*. Springer 1990.
- [3] Ullman Jeffrey D.: *Principles of Database and Knowledge-Base Systems*, Vol. I e II. Computer Science Press 1989.

LETIZIA TANCA è Professore Ordinario di Basi di Dati presso il Politecnico di Milano, e presidente del Consiglio di Corso di Studi di Ingegneria Informatica del Politecnico di Milano, campus Milano Leonardo. È autrice di diverse pubblicazioni internazionali sulle basi di dati e sulla teoria delle basi di dati, e del libro "Logic Programming and Databases", scritto con S. Ceri e G. Gottlob. Ha partecipato a vari progetti nazionali e internazionali. I suoi interessi di ricerca riguardano tutta la teoria delle basi di dati, in particolare le basi di dati deduttive, attive e orientate agli oggetti, i linguaggi a grafi per basi di dati, la rappresentazione e l'interrogazione di informazione semistrutturata, le basi di dati per piccoli dispositivi.
E-mail: tanca@elet.polimi.it

CERTIFICAZIONE DELLE CONOSCENZE INFORMATICHE NELL'UNIVERSITÀ ITALIANA

L'articolo discute l'importanza delle competenze e conoscenze informatiche nella formazione universitaria e il ruolo della certificazione di queste competenze. Dopo un esame della situazione italiana riguardo alle competenze informatiche, l'articolo analizza le attività svolte nel recente passato dalle Università e presenta le principali linee evolutive rese disponibili da AICA nell'ambito delle certificazioni informatiche.

1. INTRODUZIONE

La missione principale di AICA (Associazione Italiana per l'Informatica ed il Calcolo Automatico) è diffondere la cultura informatica in Italia. Sin dal 1961, anno della sua fondazione, AICA ha raccolto diverse comunità professionali del mondo ICT (*Information and Communication Technologies*) ovvero i produttori di tecnologie, gli utenti delle medesime, i docenti di discipline informatiche, favorendone gli incontri, la promozione professionale e i dibattiti sulle conseguenze sociali dell'evoluzione tecnologica. AICA non è l'unica associazione italiana nel mondo ICT, ma è stata la prima ed è senz'altro la più autorevole. In particolare, AICA è il membro italiano del CEPIS (*Council of European Professional Informatics Societies*, <http://www.cepis.org>), la federazione europea delle associazioni nazionali dei professionisti informatici.

Nel 1997 AICA ha iniziato ad occuparsi di certificazione delle abilità informatiche, con l'esperienza ECDL (*European Computer Driving License*, <http://www.ecdl.it>), un programma di

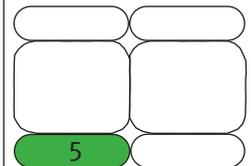
certificazione informatica, più noto come "Patente Europea del Computer", nato in ambito CEPIS con il supporto dell'Unione Europea. Il programma ECDL, che si è prefisso di diffondere a livello di massa le abilità informatiche di base, senza legarsi a particolari tecnologie proprietarie, ha avuto un notevole successo, testimoniato da quasi un milione e mezzo di certificati rilasciati fino ad oggi in Italia.

Dal 2001 AICA ha esteso l'attività di certificazione, occupandosi, con il programma EUCIP (*EUropean Certification of Informatics Professionals*, <http://www.eucip.it>), anche della certificazione delle competenze professionali più avanzate sulle tecnologie ICT. Anche EUCIP nasce a livello europeo sotto l'egida del CEPIS.

Uno dei passi più rilevanti di AICA per sostenere le proprie certificazioni fu, infatti, la collaborazione con la CRUI (Conferenza dei Rettori delle Università Italiane) per fare riconoscere la certificazione ECDL, in termini di Crediti Formativi Universitari (CFU), nei curriculum universitari. Con il programma EUCIP, AICA ha rimodulato e rafforzato il suo impegno



Cristiana Rita Alfonsi
Maria Carla Calzarossa
Paolo Ciancarini
Paolo Maresca
Luisa Mich
Fulvia Sala
Nello Scarabottolo



nell'ambito delle certificazioni, in quanto:

□ ha ristrutturato la propria offerta complessiva di certificazione professionale: mentre ECDL è rivolta prevalentemente alla massa degli utenti "front office" di tecnologie e applicazioni informatiche di produttività individuale, EUCIP è rivolta specificatamente al mondo professionale dei progettisti di sistemi "back office", ovvero costruttori e gestori di sistemi informativi;

□ al contrario di quanto avvenuto per il programma ECDL, ha coinvolto il mondo universitario italiano nella definizione dei contenuti del programma EUCIP, avviando una collaborazione con il CINI (*Consorzio Interuniversitario Nazionale per l'Informatica*) che raccoglie molte delle Università italiane con gruppi di ricerca attivi nell'ambito delle scienze e tecnologie informatiche;

□ ha stipulato nel 2004 una convenzione con la Fondazione CRUI, la struttura operativa della CRUI, per diffondere la certificazione EUCIP a livello universitario, ed in particolare nell'ambito dei percorsi formativi finalizzati alle professioni dell'informatica, favorendone anche il riconoscimento in termini di crediti formativi; a seguito di questa convenzione è stato lanciato un progetto specifico denominato EUCIP4U (EUCIP for University);

□ ha messo a punto, in collaborazione con la Fondazione CRUI, una proposta di formazione di livello universitario all'uso avanzato del computer, finalizzata alla soluzione dei problemi, attraverso il progetto IT4PS (*Information Technologies for Problem Solving*); è attualmente allo studio una nuova certificazione (IT4PS) che potrà offrire la possibilità di rendere trasparenti le capacità di risolvere problemi professionali attraverso l'uso avanzato del foglio elettronico e delle basi di dati. Scopo di questo articolo è analizzare la diffusione e l'impatto delle certificazioni ICT nel mondo universitario italiano, descrivendo il ruolo di AICA, e studiando la situazione che si è venuta a creare nell'ambito della formazione universitaria.

L'organizzazione di questo articolo è la seguente. Nella prima sezione si presenta il quadro di riferimento politico, sociale e normativo che sottolinea la rilevanza della diffusione delle conoscenze informatiche; nella seconda sezione si affronta il tema della for-

mazione certificata e della sua importanza ai fini della diffusione delle conoscenze informatiche. Nelle sezioni successive si analizza la diffusione in ambito universitario delle certificazioni ECDL, IT4PS e EUCIP. Infine, nell'ultima sezione si discutono gli effetti di tale diffusione e le prospettive di sviluppi futuri.

2. L'IMPORTANZA DELLE CONOSCENZE INFORMATICHE NELLA FORMAZIONE UNIVERSITARIA

Il contesto di riferimento nel quale inserire le considerazioni riportate in questo articolo è ben descritto dai documenti prodotti dal Consiglio Europeo nell'ambito del Vertice dei capi di Stato dei Paesi europei tenutosi a Lisbona nel marzo 2000, documenti che hanno dato luogo alla cosiddetta "Strategia di Lisbona: il rinnovamento economico, sociale e ambientale dell'U.E." [1].

In particolare, le conclusioni della Presidenza del Consiglio Europeo iniziano sottolineando la nuova sfida che si apre per l'Europa nel decennio 2000-2010:

"L'Unione Europea si trova dinanzi a una svolta epocale risultante dalla globalizzazione e dalle sfide presentate da una nuova economia basata sulla conoscenza. Questi cambiamenti interessano ogni aspetto della vita delle persone e richiedono una trasformazione radicale dell'economia europea. L'Unione deve modellare tali cambiamenti in modo coerente con i propri valori e concetti di società, anche in vista del prossimo allargamento".

Il richiamo alla società della conoscenza, ripreso più volte nel documento e sovente collegato all'innovazione tecnologica, sottolinea l'importanza della formazione e la necessità che tale formazione abbia particolare attenzione nei confronti dell'ICT.

Su mandato del Consiglio Europeo, il Consiglio per l'Istruzione ha presentato a Stoccolma, nel marzo 2001, un documento sugli obiettivi futuri e concreti dei sistemi di istruzione e di formazione necessari per realizzare la strategia di Lisbona, in cui si sottolinea ancora più esplicitamente l'importanza delle conoscenze nel settore ICT per affrontare le sfide aperte [2].

Alla formazione a tutti i livelli nel settore ICT fa

riferimento ancora più esplicitamente la Comunicazione della Commissione Europea sulle "Strategie per l'occupazione nella società dell'informazione" [3] che recita fra l'altro:

"Il lavoratore e il posto di lavoro nella società dell'informazione saranno molto diversi da quelli che conosciamo oggi. Nella società dell'informazione un numero crescente di persone svolge mansioni legate all'informazione e alla conoscenza e fa un uso crescente degli strumenti e servizi della società dell'informazione, sia durante il lavoro che nel tempo libero.

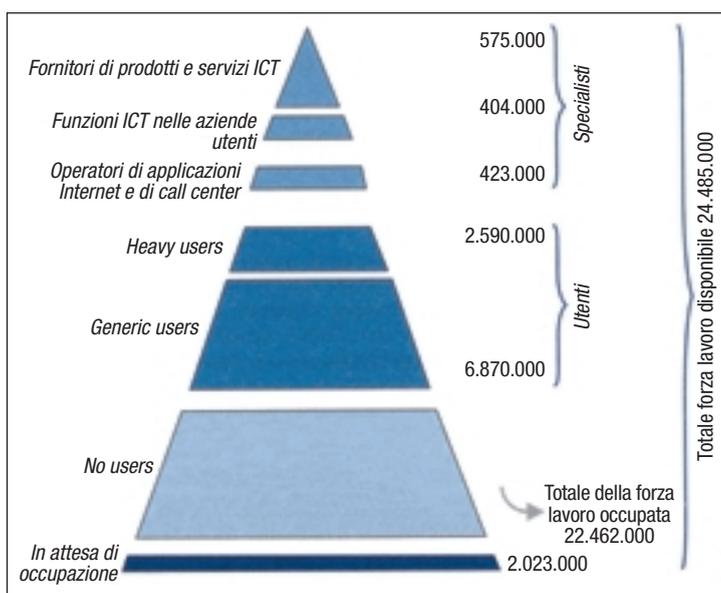
I lavoratori dell'era digitale devono quindi essere alfabetizzati alle ICT, altamente qualificati, autonomi, mobili e pronti a sottoporsi ad una formazione continua (apprendimento lungo tutto l'arco della vita). Analogamente la società dell'informazione solleva un'enorme domanda di specialisti della società dell'informazione, domanda che finora è rimasta inevasa. Poiché il lavoratore digitale potrà essere sia uomo che donna e questo tipo di lavoro ridurrà i vincoli dettati da disabilità, distanza e tempo che costituivano barriere all'occupazione, la società dell'informazione rappresenterà per tutti un più ampio accesso al lavoro".

A questa richiesta di conoscenze informatiche, sia a livello di utenti competenti di strumenti informatici, sia di specialisti delle tecnologie ICT, fa purtroppo riscontro una situazione ancora carente a livello europeo, e particolarmente deficitaria a livello italiano.

In uno studio effettuato da AICA sotto il patrocinio della Commissione Europea e del Ministero per l'Innovazione Tecnologica [4] viene valutato il costo dell'ignoranza nella società dell'informazione. Tale studio è stato ripreso ed ampliato, dando luogo ad un rapporto sulle conoscenze informatiche in Italia [5], che fornisce un interessante spaccato sulla situazione del nostro Paese nei confronti dell'ICT.

Le conoscenze informatiche necessarie alla forza lavoro occupata nel nostro Paese sono ben evidenziate dalla piramide di figura 1 (tratta dal già citato rapporto [5]) che evidenzia come, su una popolazione di lavoratori di poco superiore ai 22 milioni:

□ più di 9 milioni (il 42% del totale: i cosiddetti



ti utenti) debbano utilizzare il principale strumento informatico – il Personal Computer – quotidianamente, e in molti casi (più di 2,5 milioni) con un elevato livello di competenza;

□ quasi 1,5 milioni (il 6% del totale: gli specialisti) debbano avere competenze avanzate nelle tecnologie ICT, in quanto direttamente coinvolti nella progettazione e nella manutenzione di sistemi informatici.

A fronte di queste esigenze professionali, l'effettiva erogazione di formazione informatica specifica mostra quanta strada resti ancora da percorrere: a livello europeo, solo il 27% della forza lavoro classificata come *utenti* ha ricevuto formazione informatica di base, ma tale percentuale precipita al 17,9% per l'Italia (seguita solo da Spagna, Grecia e Portogallo).

Tale mancanza di formazione si traduce in uno spreco di risorse umane legato essenzialmente al tempo perso dagli *utenti* per risolvere problemi operativi spesso banali. Una stima di tale spreco (quantificato in [4, 5]) porta a valutare l'improduttività del singolo lavoratore in circa 2.500 €/anno, il che comporta un costo annuo per il nostro Paese attorno ai 19 miliardi di Euro.

Anche a livello di *specialisti* la situazione si conferma critica: come indicato nello stesso rapporto [5], tra i Paesi europei più industrializzati, l'Italia si pone in penultima posizione per il numero di posti di lavoro disponibili, e nonostante ciò mostra un progressivo incre-

FIGURA 1

Il mercato del lavoro in Italia dal punto di vista degli skill ICT (IV trimestre 2003)

mento di posizioni scoperte per mancanza di personale adeguatamente formato.

A questa generale carenza di formazione informatica – sia di base che specialistica – l'Università italiana è chiamata a dare risposte concrete.

Sul fronte della formazione di base, o meglio della formazione destinata agli *utenti* professionali di strumenti informatici, si è rivelato particolarmente significativo l'orientamento emerso dalla recente riforma degli ordinamenti universitari, entrata in vigore sulla base del Decreto Ministeriale 509/99, che ha introdotto il Credito Formativo Universitario come unità di misura dell'impegno allo studio richiesto per conseguire qualsiasi titolo universitario (laurea triennale, successiva laurea specialistica biennale, laurea a ciclo unico quinquennale).

La formazione informatica di base ha infatti trovato spazio nei nuovi Corsi di Studio a due livelli:

- con l'inclusione, per la grande maggioranza dei Corsi di Studio triennali, dell'Informatica tra le discipline *di base* (insieme alla Matematica e alla Fisica) cui dedicare CFU;
- con la definizione di altre attività formative, oltre a quelle di base e a quelle caratterizzanti il singolo Corso di Studio, destinate "ad acquisire ... ulteriori abilità informatiche e telematiche...".

Quanto alla formazione specialistica, fornita come ovvio dai Corsi di Studio di area Informatica e dell'Ingegneria dell'Informazione, il suddetto decreto ha dato la possibilità di differenziare l'offerta formativa, progettando ed attivando Corsi di Studio mirati alla creazione di figure professionali destinate a specifici contesti applicativi.

Inoltre, il dibattito che è nato sul ruolo dell'informatica come "lingua franca" ha spinto il mondo universitario ad interrogarsi sulla necessità da una parte di orientare maggiormente i percorsi formativi per i professionisti del settore verso la trasparenza delle competenze per il mercato del lavoro, dall'altra di impegnarsi a fornire a tutti i futuri lavoratori le conoscenze di base per diventare più competitivi in un contesto europeo.

A causa, tuttavia, delle più recenti modifiche dell'impianto dell'offerta formativa universitaria delineata dalla riforma dell'autonomia

didattica e del dibattito accademico sull'assegnazione dei crediti, si rischia di deviare dalla strada intrapresa negli ultimi anni dai diversi attori e a diverso titolo. Il rischio è quello di trovarsi di fronte ad una battuta di arresto rispetto al percorso di apertura e di confronto con le richieste e le esigenze del mondo del lavoro.

3. L'IMPORTANZA DELLA CERTIFICAZIONE DELLE CONOSCENZE INFORMATICHE

Il problema della formazione alle conoscenze informatiche, come pure a qualsiasi altro tipo di conoscenza, comporta – oltre all'erogazione vera e propria – due momenti fondamentali:

- la definizione *ex ante* delle conoscenze e delle competenze che si intende far acquisire al fruitore della formazione;
- la verifica *ex post* che tali conoscenze e competenze siano state effettivamente acquisite (ed eventualmente permangano nel tempo).

A questo riguardo, esistono due approcci fondamentalmente opposti, anche se integrabili in vario modo. Un primo approccio delega al singolo erogatore di formazione il compito di definire gli obiettivi in termini di conoscenze e competenze da acquisire, e i metodi di verifica dei risultati conseguiti. Un secondo approccio delega invece ad un ente NON coinvolto nella formazione (un ente terzo, pubblico o privato) il compito di:

- definire conoscenze e competenze in un *syllabus* reso noto a tutti gli erogatori di formazione interessati;
- definire e controllare le modalità di verifica del raggiungimento degli obiettivi formativi, ovvero *certificare* che il soggetto fruitore possiede in misura adeguata le conoscenze e le competenze previste dal *syllabus*.

Il primo approccio è quello tradizionalmente adottato dalle Università, italiane e non. Questo approccio privilegia l'autonomia del singolo Ateneo (seppure all'interno di uno schema di massima definito a livello ministeriale), e permette di impostare piani formativi adeguati alle esigenze del tessuto socio-economico locale e di valorizzare al meglio le competenze del personale docente dell'Ateneo stesso.

Questo approccio ha però due limiti di fondo:

1. una intrinseca auto-referenzialità del processo valutativo finale, poiché è il singolo Ateneo, o meglio, i singoli docenti, che verificano che ogni laureato possieda realmente le conoscenze e le competenze previste;
2. un ostacolo alla effettiva mobilità professionale (nazionale e internazionale) del laureato: la sola denominazione del titolo conseguito, pur se integrata da una lista degli esami superati, non fornisce infatti informazioni sufficienti a stabilire le conoscenze e competenze effettivamente possedute dal laureato.

Esistono naturalmente aspetti che rendono meno critici i due limiti sopra citati. Per esempio, la qualificazione didattica di un Ateneo può addirittura valorizzare l'auto-referenzialità sopra citata. Invece l'adozione del *diploma supplement* come descrizione (in italiano e in inglese) delle effettive conoscenze e competenze acquisite dal laureato nel suo percorso formativo integra il tradizionale diploma di laurea [http://www.miur.it/0002Univer/0023Studenten/0831Diplom/index_cf2.htm].

Ci sono tuttavia validi motivi per considerare l'opportunità di una formazione universitaria *certificata*, sia pure anche solo per una parte di quanto effettivamente appreso. Questa tesi è facilmente sostenibile quando ci si riferisce a conoscenze e competenze di base, solitamente trasversali rispetto alle tematiche specifiche del singolo Corso di Studio, quali il livello di competenza nell'uso degli strumenti ICT, o il livello di conoscenza di una lingua straniera.

Non è però insensato certificare anche un insieme minimo di conoscenze e competenze, quando ci si aspetta che siano possedute da *tutti* gli specialisti del settore ICT onde favorire la riconoscibilità e dunque la mobilità e in buona sostanza l'accettazione da parte del mercato del lavoro. Tale certificazione risulta ancora più significativa qualora al progetto del suo syllabus abbia partecipato un consesso internazionale comprendente anche aziende produttrici e aziende utilizzatrici di tecnologia ICT, i cui suggerimenti sull'insieme minimo di conoscenze e competenze costituiscono un prezioso contributo.

A questo riguardo, non è un caso che, nelle già citate conclusioni della Presidenza del Consiglio Europeo di Lisbona [1] siano presenti frasi come:

“I sistemi europei di istruzione e formazione devono essere adeguati alle esigenze della società dei saperi e alla necessità di migliorare il livello e la qualità dell'occupazione. Dovranno offrire possibilità di apprendimento e formazione adeguate ai gruppi bersaglio nelle diverse fasi della vita: giovani, adulti disoccupati e persone occupate soggette al rischio che le loro competenze siano rese obsolete dai rapidi cambiamenti. Questo nuovo approccio dovrebbe avere tre componenti principali: lo sviluppo di centri locali di apprendimento, la promozione di nuove competenze di base, in particolare nelle tecnologie dell'informazione, e qualifiche più trasparenti”.

e come:

“Un quadro europeo dovrebbe definire le nuove competenze di base da fornire lungo tutto l'arco della vita: competenze in materia di tecnologie dell'informazione, lingue straniere, cultura tecnologica, imprenditorialità e competenze sociali; dovrebbe essere istituito un diploma europeo per le competenze di base in materia di tecnologia dell'informazione, con procedure di certificazione decentrate, al fine di promuovere l'alfabetizzazione “digitale” in tutta l'Unione”.

La CRUI ha svolto un importante ruolo di diffusione del concetto di certificazione delle conoscenze informatiche nell'ambito del progetto *CampusOne*, con particolare riferimento alle competenze operative di base certificate dall'attestato ECDL.

All'impatto che la certificazione ECDL ha avuto nell'Università italiana negli ultimi anni è dedicata la prossima sezione di questo articolo.

Esistono però ampi spazi di evoluzione delle certificazioni informatiche, sia in termini di maggiori competenze per gli utenti evoluti di tecnologia, sia relativamente agli specialisti ICT. A tali opportunità saranno dedicate le sezioni successive.

4. LA CERTIFICAZIONE ECDL NELL'UNIVERSITÀ ITALIANA

Come sottolineato in precedenza, AICA ha iniziato ad occuparsi di certificazione della formazione informatica nel 1997 con l'intro-

duzione in Italia del programma ECDL, il cui scopo è quello di attestare il possesso di abilità informatiche di base nell'uso di strumenti informatici.

Anche il mondo universitario italiano ha dovuto (e deve) affrontare il problema della formazione e certificazione delle abilità informatiche di base. Queste attività, che a regime non dovrebbero essere compiti istituzionali delle Università, derivano dal fatto che gli studenti, che si immatricolano ad un Corso di Studio universitario, non sempre hanno acquisito tali abilità informatiche nella loro carriera scolastica precedente.

A seguito del protocollo d'intesa siglato dall'allora Ministero della Pubblica Istruzione (MPI) e dalla CRUI nel marzo del 2001, che all'art. 2 recita: "La CRUI si impegna a favorire presso le diverse Università il riconoscimento del credito formativo derivante dall'aver conseguito la patente informatica", gli Atenei sono stati incoraggiati ad adottare la certificazione ECDL come strumento di qualificazione dell'alfabetizzazione informatica per i vari Corsi di Studio. Un ulteriore impulso al programma ECDL a livello di formazione universitaria è stato dato dalla Fondazione CRUI, attraverso il progetto CampusOne.

Gli Atenei si sono quindi trovati a dover affrontare complessi problemi di organizzazione e gestione delle attività didattiche e dell'accreditamento legati all'introduzione della certificazione ECDL.

Per analizzare la diffusione e l'impatto nel mondo universitario Italiano della certifica-

zione ECDL e delle certificazioni informatiche avanzate, nel 2001 AICA, in collaborazione con CINI e Fondazione CRUI, ha istituito uno specifico Osservatorio (<http://osservatorio.consortio-cini.it>). L'Osservatorio cura indagini annuali mirate a raccogliere e diffondere le esperienze maturate dagli Atenei nell'ambito delle certificazioni informatiche. I risultati delle indagini svolte sono stati pubblicati come supplemento della rivista "Mondo Digitale" e su riviste e atti di convegni internazionali [6, 7, 8, 9, 10, 11, 12].

Le indagini svolte fino ad ora hanno mostrato come la certificazione ECDL sia diventata uno strumento consolidato ed ampiamente utilizzato dagli Atenei per accreditare le abilità informatiche di base degli studenti. Se nel 2001 si erano censiti 17 "pionieri" che sperimentavano la certificazione ECDL a livello di Ateneo, Facoltà o Corso di Studio, già nel 2002 i programmi ECDL erano presenti presso 33 Atenei dei 65 che avevano aderito all'indagine promossa dall'Osservatorio. Negli anni successivi, grazie soprattutto all'attivazione del progetto CampusOne, la presenza di programmi ECDL presso gli Atenei si è sempre più intensificata (46 Atenei nel 2003 e 50 nel 2004). Anche nell'anno solare 2005 il livello di diffusione dei programmi ECDL negli Atenei è rimasto sostanzialmente invariato (48 Atenei dei 53 che avevano aderito all'indagine).

I dati quantitativi relativi al programma ECDL sono un'ulteriore testimonianza del suo importante ruolo in ambito universitario e dello sforzo organizzativo richiesto agli Atenei. La figura 2 sintetizza le prestazioni degli studenti, mostrando l'andamento del numero di esami svolti per i singoli moduli ECDL in relazione al numero di esami superati dagli studenti (sulla base di dati raccolti dall'Osservatorio). Come si nota, il tasso di successo non varia in maniera significativa di anno in anno e si attesta intorno all'83%.

A fronte degli esami superati, si consegue una certificazione ECDL START superando gli esami corrispondenti a 4 dei 7 moduli previsti dal programma ECDL, e una certificazione ECDL FULL superando gli esami corrispondenti a tutti i 7 moduli ECDL.

Nell'anno solare 2003, agli studenti universitari sono state rilasciate circa 19.000 certificazioni ECDL (pari a circa il 18% del numero

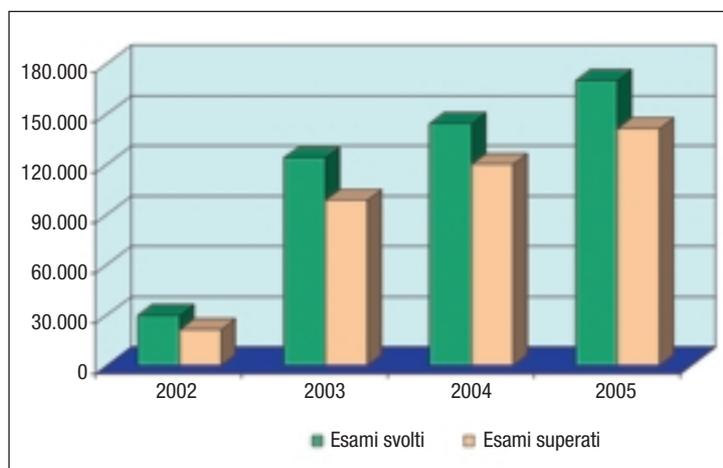


FIGURA 2
Distribuzione del numero di esami svolti e superati dagli studenti universitari

complessivo di “patenti” rilasciate in Italia), mentre negli anni solari 2004 e 2005, ne sono state rilasciate circa 20.000 e 24.000, rispettivamente. Come si può notare dalla figura 3, fino al 2004, il numero di certificazioni ECDL START supera di alcune migliaia quello delle ECDL FULL. Al contrario, nel 2005 le “patenti” ECDL START e ECDL FULL sono distribuite in maniera pressoché bilanciata. Questo evidenzia una tendenza degli Atenei a rivolgersi, dopo un periodo iniziale di sperimentazione, ad un modello di certificazione più completa, come può essere la certificazione ECDL FULL. Si osserva infatti che nel 2005, il numero di certificazioni ECDL FULL rilasciate a studenti universitari supera di circa il 28% il numero di certificazioni rilasciate nell’anno precedente, mentre è sostanzialmente invariato il numero di certificazioni ECDL START rilasciate.

Per quanto riguarda l’organizzazione delle attività di formazione mirate al conseguimento della certificazione ECDL, gli Atenei solitamente non utilizzano una sola modalità di erogazione, ma affiancano l’auto-apprendimento da parte degli studenti ad una didattica di tipo frontale.

Come si può osservare dalla figura 4, il numero medio di ore di formazione frontale dedicata a ciascun modulo ECDL varia in funzione dei moduli, ma risulta contenuto tra un minimo di

6,5 h per il modulo “Uso computer e gestione file”, e un massimo di 11,2 h per il modulo “Basi di dati”. Si nota anche che nel 2002, rispetto agli anni successivi, gli Atenei dedicavano mediamente un numero maggiore di ore di formazione a tutti i moduli ECDL, tranne che al modulo “Concetti teorici di base”. Occorre però sottolineare anche che non tutti gli Atenei offrono formazione frontale per i 7 moduli ECDL. In particolare, le attività di formazione frontale per i moduli “Uso computer e gestione file” e “Elaborazione testi” sono presenti presso circa il 75% degli Atenei, mentre è minore (circa 60%) la frazione di Atenei che of-

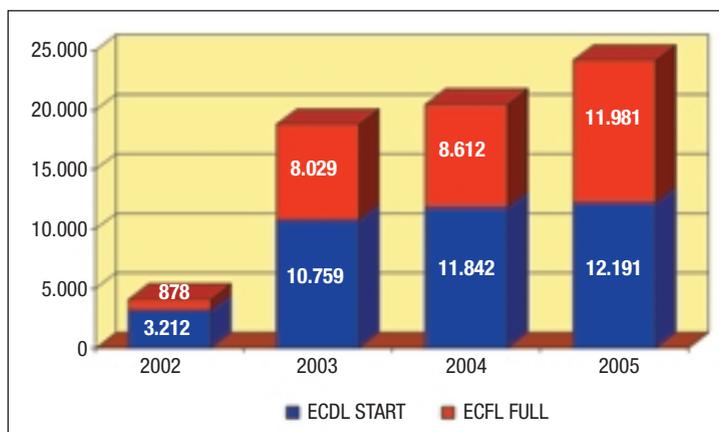


FIGURA 3
Distribuzione del numero di certificazioni ECDL START e ECDL FULL rilasciate dagli Atenei

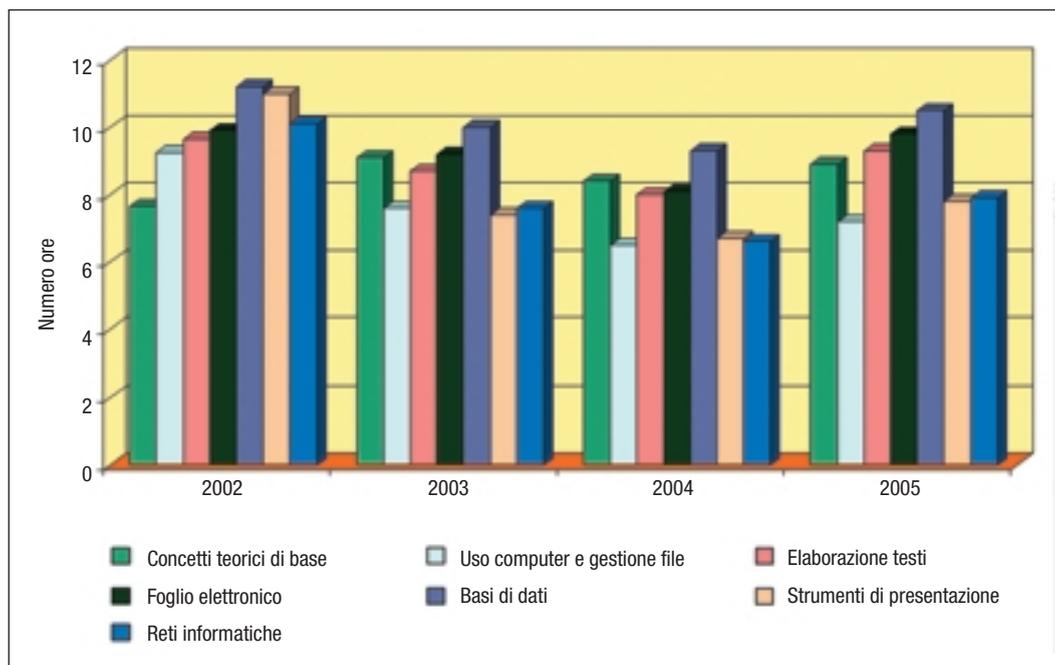


FIGURA 4
Distribuzione del numero medio di ore di formazione dedicate a ciascun modulo ECDL

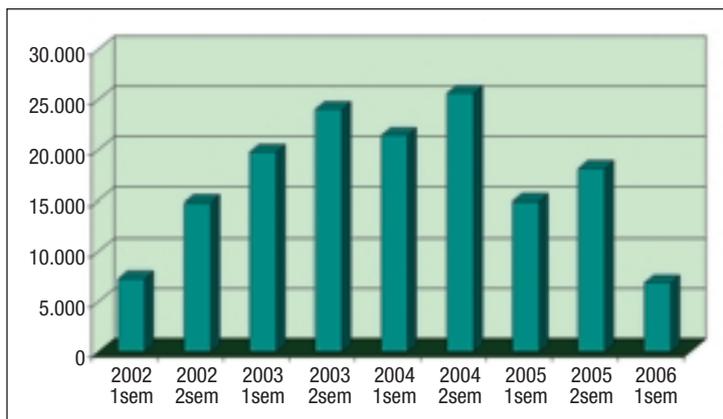


FIGURA 5
Stretta correlazione fra la diffusione di ECDL nelle Università italiane e il progetto CampusOne

frono formazione frontale per i moduli “Basi di dati” e “Strumenti di presentazione”.

A fronte di un quadro generale apparentemente stabilizzato – e che sembra dimostrare come la cultura delle certificazioni informatiche rappresenti una concreta realtà presso la maggioranza degli Atenei Italiani – occorre però sottolineare che l’andamento tendenziale mostra una riduzione dell’importanza dell’ECDL nell’Università italiana.

Come evidenziato infatti nella figura 5, dopo un picco di penetrazione dell’ECDL negli anni 2003 e 2004 (rispettivamente con 44.000 e 47.000 Skills Card distribuite agli studenti universitari) si è scesi nel 2005 a 33.000, e i dati del primo semestre 2006 mostrano numeri più che dimezzati rispetto a quelli dell’anno precedente.

Questo andamento potrebbe essere giustificato dalla conclusione del progetto CampusOne (avvenuta nel settembre 2004) e dalla mancanza del relativo supporto finanziario, come pure dall’interesse dell’Università verso una formazione a competenze informatiche di livello più avanzato dell’ECDL.

Resta ovviamente da capire in quale misura lo studente che entra nell’Università abbia familiarità con le tecnologie ICT; il progressivo disimpegno degli Atenei italiani nei confronti di corsi di alfabetizzazione informatica a favore di una formazione più avanzata può essere infatti ostacolato da una diffusa mancanza delle competenze di base.

Va inoltre riconosciuto come ECDL abbia avuto un ruolo fondamentale, per il contesto universitario italiano, nell’evidenziare l’esigenza di disporre di un insieme minimo di abilità informatiche in tutti i Corsi di Stu-

dio, in particolare in quelli più lontani dalla cultura ICT.

È comunque un segnale di allarme il fatto che alla riduzione delle certificazioni ECDL non abbia sinora fatto riscontro l’introduzione di programmi di certificazione alternativi, di livello superiore, eventualmente finanziati con il supporto diretto degli studenti. Una conferma di questa tendenza senza una corrispondente evoluzione della formazione informatica di base e della sua certificazione rischiano infatti di vanificare gli sforzi legati all’introduzione dell’ECDL e al suo riconoscimento nell’università italiana.

Proprio nell’ottica di fornire all’università italiana programmi di formazione più avanzati, supportati da un sistema di certificazione che ne garantisca la riconoscibilità, sono stati attuati i progetti IT4PS e EUCIP4U, che saranno presentati nelle prossime sezioni.

5. IL PROGETTO IT4PS

Il progetto IT4PS (*Information Technologies for Problem Solving* [13]) – condotto dalla Fondazione CRUI insieme ad AICA – nasce dalla pluriennale esperienza dei suoi promotori nella formazione, all’interno degli Atenei italiani, all’utilizzo dei principali strumenti informatici (elaboratori di testi, fogli elettronici, gestori di basi di dati).

Il progetto – destinato agli utenti evoluti di strumenti ICT – si propone di superare i limiti principali della formazione alla certificazione ECDL:

- il livello relativamente elementare di tale formazione, sufficiente a garantire dimestichezza con gli strumenti informatici ma senz’altro poco adeguato per un utilizzo professionale, quale quello che ci si può aspettare da studenti universitari (futuri professionisti dell’informatica);
- la natura stessa della formazione ECDL, tipicamente *package oriented*, che presenta cioè le potenzialità dei vari strumenti informatici in modo avulso da uno specifico contesto applicativo.

La certificazione ECDL attesta dunque la familiarità dello studente con lo strumento informatico, ma non dice nulla circa l’effettiva capacità dello studente di utilizzare in modo proficuo lo strumento stesso per risolvere

i propri problemi curriculari (e in futuro professionali).

Per dare una risposta concreta a queste limitazioni, il progetto IT4PS si è proposto di predisporre una formazione, di livello universitario, di competenze operative avanzate ICT - con riferimento in particolare al foglio elettronico ed alle basi di dati - secondo un approccio didattico orientato al *problem solving* e contestualizzato all'ambito disciplinare di specifiche famiglie di Corsi di Studio.

Il progetto ha prodotto - nella sua prima fase - sei moduli formativi all'uso contestualizzato del foglio elettronico e delle basi di dati in tre diversi contesti curriculari: Economia, Medicina e Farmacia, Statistica nelle Scienze Sociali. I moduli formativi sono stati pubblicati in altrettanti volumi, [14, 15, 16, 17, 18, 19] e sono stati inoltre realizzati due sistemi autore per consentire la predisposizione di esercizi di autovalutazione a soluzione guidata dal docente [20]. Una sperimentazione sul campo - che ha coinvolto una decina di Atenei e qualche centinaio di studenti - ha permesso di verificare l'efficacia dell'approccio e il suo gradimento da parte di formatori e studenti.

Le competenze avanzate di *problem solving* devono però poter essere verificate e certificate da attestati che testimonino una competenza e una comprensione profonda delle possibilità dello strumento *per la soluzione concreta di problemi curriculari*. Lo studio di tali strumenti di certificazione costituisce la fase finale del progetto IT4PS, destinata a concludersi nel giugno 2007.

6. LA CERTIFICAZIONE EUCIP E IL PROGETTO EUCIP4U

Se le competenze degli utenti di strumenti informatici meritano particolare attenzione da parte dei sistemi di certificazione, non vanno tuttavia dimenticate le competenze degli *specialisti*, e l'utilità di prevedere profili condivisi di tali competenze, verificabili e certificabili in modo analogo a quanto succede per gli utenti.

Fra le varie proposte presenti sul mercato, merita una particolare attenzione l'iniziativa europea lanciata - come la certificazione ECDL - dal CEPIS e denominata EUCIP (*EUropean Certification of Informatics Professionals*).

Come si legge sul sito italiano della certificazione EUCIP (<http://www.eucip.it/>):

“La crescente dipendenza delle attività economiche e sociali dalle Tecnologie dell'Informazione (ICT) rende critico il problema di formare e reperire profili di competenza specifici e aggiornati nel settore.

Nella Società dell'Informazione, il corretto funzionamento dei sistemi ICT è cruciale per lo sviluppo economico e sociale.

In altre parole, diventa di fondamentale importanza garantire al mondo dell'utenza, ai cittadini e alle organizzazioni, che i sistemi ICT vengano progettati, realizzati e gestiti tenendo conto di alcuni requisiti di fondo, tra cui almeno due assolutamente prioritari:

- la *“robustezza”*, ossia che tali sistemi siano ingegnerizzati in modo da essere a prova d'uso improprio e da garantire comunque un livello di servizio prevedibile, affidabile e ragionevolmente efficiente anche in presenza di situazioni di carico eccezionali;

- la *“sicurezza”*, ossia che gli stessi sistemi siano progettati in modo da mantenere integre e recuperabili le informazioni, anche nei casi di malfunzionamento, assicurando la protezione dei dati *“sensibili”*.

Il professionista ICT cui viene affidata la concezione, la realizzazione e l'esercizio del sistema, deve possedere e dimostrare, proprio attraverso la certificazione periodica delle sue competenze, oltre ad una solida conoscenza specifica, anche una vasta esperienza, mantenuta continuamente aggiornata con i progressi della tecnologia.

Si stima che fra 10 anni l'80% delle tecnologie ICT oggi operative sarà diventato obsoleto e dovrà essere rimpiazzato.

Per quell'epoca l'80% degli specialisti del settore lavoreranno sulla base di una formazione scolastica risalente a più di 10 anni addietro. In sostanza, mentre la forza lavoro invecchia, la tecnologia ringiovanisce e questo non è mai stato vero come nel caso delle ICT. In altre parole, nel prossimo futuro, sarà ancora più difficile di oggi garantire che l'esperto ICT sia veramente all'altezza del compito assegnatogli.

È per far fronte a questo problema che il CEPIS ha deciso di avviare nel 1999 un programma di Certificazioni Europee di quelle competen-

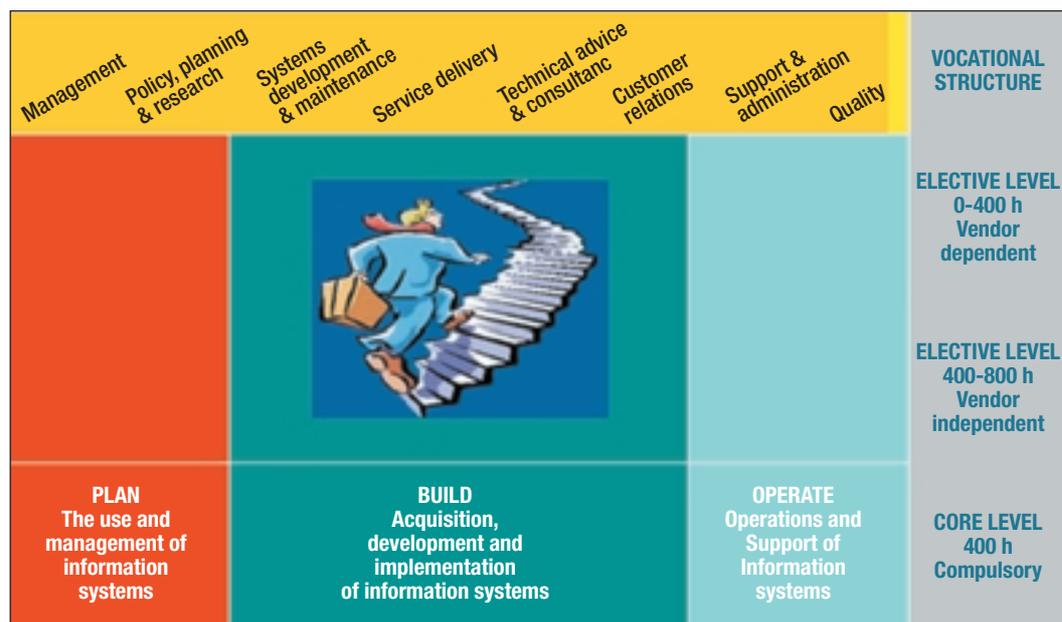


FIGURA 6
Struttura
della certificazione
EUCIP

ze ICT ritenute indispensabili per esercitare la professione di specialista”.

La struttura della certificazione EUCIP, riportata nella figura 6, evidenzia la presenza di due livelli successivi di competenza richiesti allo specialista:

- un livello *Core* (suddiviso in tre aree *Plan*, *Build* e *Operate*) ritenuto fondamentale per qualsiasi professionista del settore ICT e di settori “limitrofi”, nei quali i contenuti tecnologici sono particolarmente importanti (per esempio nel moderno management aziendale);
- un livello *Elective*, destinato a configurare competenze specifiche di professionisti ICT in particolari settori applicativi.

Le competenze di livello *Core*, di tipo trasversale, vengono certificate mediante un test a risposta multipla, in lingua inglese, relativo a ciascuna delle tre aree *Plan*, *Build* e *Operate*. Le competenze di livello *Elective* sono invece certificate con un esame orale fra il candidato e una commissione composta da almeno due esaminatori accreditati, cui il candidato stesso presenta il *portfolio* delle proprie esperienze formative e professionali e di eventuali altre certificazioni in suo possesso. L'importanza del progetto EUCIP è testimoniata anche dall'adozione dei vari *syllabus* EUCIP come basi per la definizione dei profili di competenze ICT nell'ambito dell'iniziativa europea

e-Skills Forum (<http://ec.europa.eu/enterprise/ict/policy/ict-skills.htm>).

Per diffondere la sensibilità a tale certificazione nei Corsi di Studio universitari di area prettamente informatica (di Ingegneria e di Scienze), AICA, CINI e Fondazione CRUI hanno lanciato il progetto EUCIP4U. A tale progetto hanno aderito 66 Corsi di Studio, in grande maggioranza appartenenti alle Classi 9 (Classe delle Lauree in Ingegneria dell'Informazione) e 26 (Classe delle Lauree in Scienze e Tecnologie Informatiche). I Corsi di Studio sono distribuiti su 30 Atenei diversi. Inoltre, nell'ambito di tale progetto, sono stati istituiti presso 28 Atenei italiani altrettanti Centri di Competenza Universitari (CCU) abilitati allo svolgimento dei test di certificazione. È stato anche realizzato da parte del CINI materiale didattico per la preparazione in modalità *e-learning* alla certificazione EUCIP *Core*.

I primi risultati del progetto mostrano che nel triennio 2004-2006, sono stati formati 862 studenti e sono stati svolti più di 1.600 esami di certificazione, con un tasso di successo che si attesta tra il 60% e il 65%.

Il progetto EUCIP4U, tuttora in corso, si ripromette di estendere ulteriormente la penetrazione della certificazione EUCIP *Core*, coinvolgendo studenti di Corsi di Studio vicini all'area prettamente informatica, ma soprattutto di stimolare la diffusione di percorsi formativi mirati all'ottenimento delle certificazioni EUCIP

Elective, particolarmente interessanti per la loro caratteristica di valutare globalmente le competenze in possesso di uno specialista ICT.

7. CONCLUSIONI

Negli ultimi 10 anni AICA è entrata nel mercato della certificazione professionale, prima con le certificazioni delle abilità informatiche definite da ECDL, e successivamente con le certificazioni delle conoscenze e capacità professionali definite da EUCIP. La tabella 1 riassume la "pila" delle certificazioni attualmente attive.

Come illustrato precedentemente, queste certificazioni hanno suscitato l'attivo interesse delle Università italiane, che ha portato ad una loro diffusione su scala nazionale. Occorre comunque evidenziare e commentare alcune criticità.

La certificazione ECDL si è diffusa a livello universitario giovandosi del particolare momento storico attraversato dagli Atenei italiani, rivoluzionati dal processo di riforma didattica avviato nel 2000. Come abbiamo visto, è in atto una diminuzione della richiesta di certificazione ECDL a livello universitario. Diminuzione spiegabile sia dalla conclusione del progetto *CampusOne*, che dalla convinzione, che si sta sempre più affermando, tra gli operatori della formazione ICT in ambito accademico, che la certificazione ECDL sia da conseguire a livello della scuola media superiore (o anche inferiore) per cui dovrebbe configurarsi come pre-requisito all'istruzione universitaria.

A fronte del progressivo disimpegno delle Università italiane nei confronti di una formazione alle abilità informatiche di base, un problema che rimane comunque aperto è il livello di alfabetizzazione alle tecnologie ICT relativo agli studenti che entrano nel mondo universitario. In base a statistiche recenti della Commissione Europea (Eurostat 2005) in Italia il 33% dei giovani fra 16 e 24 anni non possiede competenze informatiche o le possiede a basso livello, contro una media europea del 23% e un valore, per i Paesi scandinavi, inferiore al 5%.

La volontà, da parte degli Atenei italiani, di concentrare risorse nella formazione ICT più avanzata, potrebbe essere ostacolata, o vanificata, specialmente nei Corsi di Studio non a carattere scientifico, da una diffusa debolezza delle competenze di base.

Programma	Sforzo	Obiettivo
EUCIP Elective	800 h + "altro"	conoscenze professionali specializzate
EUCIP Core	400 h	conoscenze professionali di base
ECDL Advanced	75 h × modulo	strumenti di problem solving
ECDL	75 h	abilità di base

TABELLA 1

La "pila" delle certificazioni offerte da AICA

Allo scopo di non dissipare l'esperienza maturata riguardo al processo di formazione informatica certificata a livello universitario, e di avviarne una graduale evoluzione, è urgente accelerare il processo di affiancamento e progressiva sostituzione di ECDL con certificazioni innovative definite sulla base di esigenze di specifiche Facoltà, strada inaugurata da AICA e Fondazione CRUI con il progetto IT4PS. La pervasività degli strumenti informatici nelle varie discipline richiede una specializzazione della formazione al *problem solving* che deve permettere agli utenti-studenti di risolvere i problemi specifici proposti nei loro diversi ambiti di studio.

Un discorso diverso merita la certificazione EUCIP. L'esperienza del progetto EUCIP4U con i Corsi di Studio di discipline informatiche è stata positiva. Tuttavia la strada da percorrere è ancora lunga, e si dovrebbe sviluppare su almeno due dimensioni. La prima dimensione, che potremmo definire "orizzontale" riguarda la diffusione di EUCIP in percorsi formativi accademici non strettamente tecnologici, quali per esempio i Corsi di Studio di Economia Aziendale o di Matematica Applicata. Questo è un terreno scarsamente esplorato e molto impegnativo. La seconda dimensione, più "verticale", riguarda la diffusione di EUCIP nella formazione permanente professionale. Molte Università cercano di produrre syllabi e contenuti per corsi di master professionalizzanti. EUCIP, specie al livello Elective, ha caratteristiche che ben si adatterebbero ad un'alta formazione, che si potrebbe perciò proporre anche negli ambiti professionali più strutturati, quale ad esempio quello dell'Ordine

degli Ingegneri dell'Informazione. Un ulteriore ambito verticale è quello della formazione e aggiornamento degli insegnanti di materie informatiche nelle scuole superiori. Anche in questo caso una specifica offerta formativa EUCIP Elective potrebbe risultare interessante per il mercato.

Ringraziamenti

Si ringraziano il prof. Marco Ferretti del CINI e la dott.ssa Paola Ferrari della segreteria organizzativa CINI-EUCIP per la collaborazione sui dati del progetto EUCIP4U.

Bibliografia

- [1] Bollettino UE 3-2000, Consiglio Europeo di Lisbona. Disponibile on-line all'indirizzo <http://europa.eu/bulletin/it/200003/i1001.htm#anch0002>.
- [2] Bollettino UE 3-2001, Consiglio Europeo di Lisbona. Disponibile on-line all'indirizzo <http://europa.eu/bulletin/it/200103/i1001.htm#anch0002>.
- [3] Comunicazione della Commissione Europea: *Strategie per l'occupazione nella società dell'informazione*. COM, 2000, Vol. 48, Bruxelles, 04/02/2000.
- [4] Camussone P.F., Occhini G.: *Il costo dell'ignoranza nella società dell'informazione*. Rapporto AICA, 2003.
- [5] Camussone P.F., Occhini G., Sala F.: *Le conoscenze informatiche in Italia: Siamo pronti per la società dell'informazione?* Egea, 2006.
- [6] Calzarossa M.: Indagine ECDL 2002. Disponibile on-line sul sito: <http://osservatorio.consortio-cini.it>.
- [7] Calzarossa M., Ciancarini P., Maresca P., Mich L., Scarabottolo N.: Indagine sull'Alfabetizzazione Informatica nell'Università Italiana. *Mondo Digitale*, anno III, *Supplemento* al n. 1, Marzo 2004.
- [8] Calzarossa M., Ciancarini P., Maresca P., Mich L., Scarabottolo N.: La Certificazione ECDL nell'Università Italiana. *Mondo Digitale*, anno III, *Supplemento* al n. 4, Dicembre 2004.
- [9] Calzarossa M., Ciancarini P., Maresca P., Mich L., Scarabottolo N.: Le Certificazioni Informatiche nell'Università Italiana. *Mondo Digitale*, anno IV, *Supplemento* al n. 4, Dicembre 2005.
- [10] Calzarossa M., Ciancarini P., Maresca P., Mich L., Scarabottolo N.: *The ECDL Certification of ICT Usage Skills in the Italian Universities*. In: Proc. of the Twelfth International Conference on Distributed Multimedia Systems (DMS2006) – Distance Education Technologies Workshop (DET06), 2006, p.238-243.
- [11] Calzarossa M., Ciancarini P., Maresca P., Mich L., Scarabottolo N.: Le Certificazioni Informatiche nelle Università Italiane. *Mondo Digitale*, anno V, *Supplemento* al n. 4, Dicembre 2006.
- [12] Calzarossa M., Ciancarini P., Maresca P., Mich L., Scarabottolo N.: The ECDL Programme in Italian Universities, Computers & Education. *Elsevier*, 2007 (in corso di stampa).
- [13] Alfonsi C.R., Pedreschi D., Scarabottolo N., Simi M.: Il progetto IT4PS: il computer per la soluzione di problemi. *Mondo Digitale*, anno V, n. 3, *Settembre* 2006, p. 3-15.
- [14] Atzeni P., De Checchi A., Sindoni G., Tirelli M., Fabrizio A., Pacini G.: *Il foglio elettronico per Economia*. McGraw-Hill, 2005.
- [15] Brogi A., Martinelli A., Gervasi V., Manghi P., Fabrizio A., Pacini G.: *Il foglio elettronico per Medicina e Farmacia*. McGraw-Hill, 2005.
- [16] Bagnati D., Nicolini G., Viscusi N., Salini S., Fabrizio A., Pacini G.: *Il foglio elettronico per la Statistica nelle Scienze sociali*. McGraw-Hill, 2005.
- [17] Atzeni P., De Checchi A., Sindoni G., Tirelli M., Fiorentino G., Pala A.P.: *Le basi di dati per Economia*. McGraw-Hill, 2006.
- [18] Manghi P., Brogi A., Gervasi V., Martinelli A., Fiorentino G., Pala A.P.: *Le basi di dati per Medicina e Farmacia*. McGraw-Hill, 2006.
- [19] Bagnati D., Nicolini G., Salini S., Viscusi N., Fiorentino G., Pala A.P.: *Le basi di dati per la Statistica nelle Scienze sociali*. McGraw-Hill, 2006.
- [20] Fabrizio A., Fiorentino G., Pacini G.: *I sistemi autore PSWelcome e Access Test Manager*. McGraw-Hill, 2006.

CRISTIANA RITA ALFONSI dal 1996 è Responsabile dell'Unità Progetti, Servizi e Formazione della Fondazione CRUI, occupandosi in particolare della progettazione, della gestione e del coordinamento, del monitoraggio e della valutazione, della formazione del personale universitario non docente, dell'orientamento. I progetti più importanti di cui si è occupata in questo periodo sono: Campus (1996-2000); Credits (1998-2000); Apollo (1999-2000); CampusOne (2000-2003); IT4PS (2003/2004); Eucip4U (2005/2007); B1-on-line (2005/2007). Ha diretto e coordinato la ricerca sul tutorato universitario e il management didattico. Ha progettato l'Osservatorio sull'e-learning universitario in Italia ed è Responsabile del Progetto ELUE (E-Learning and University Education) co-finanziato dall'Unione Europea.
E-mail: alfonsi@fondazionecru.it

MARIA CARLA CALZAROSSA è professore ordinario di impianti di elaborazione presso la Facoltà di Ingegneria dell'Università di Pavia. Dal 2002 coordina il Gruppo di Lavoro che ha realizzato l'Osservatorio Permanente delle Certificazioni Informatiche negli Atenei Ita-

liani (<http://osservatorio.consortio-cini.it>). Collabora con la Commissione Europea per le attività di monitoraggio del Sesto Programma Quadro.
E-mail: mcc@unipv.it

Paolo Ciancarini è ordinario di informatica all'Università di Bologna, dove lavora dal 1992. È socio e consigliere di AICA, nonché membro del Comitato Scientifico di Mondo Digitale. Fa parte del Gruppo di Lavoro che ha realizzato l'Osservatorio Permanente delle Certificazioni Informatiche negli Atenei Italiani.
E-mail: ciancarini@cs.unibo.it

PAOLO MARESCA è professore associato di sistemi per l'elaborazione delle informazioni presso il Dipartimento di Informatica e Sistemistica dell'Università di Napoli Federico II.

Fa parte del Gruppo di Lavoro che ha realizzato l'Osservatorio Permanente delle Certificazioni Informatiche negli Atenei Italiani. È membro dell'AICA ed è valued member dell'IEEE.

Autore di circa 100 lavori in congressi, e riviste nazionali ed internazionali egli è referee e associated editor di riviste internazionali nel settore informatico. È coordinatore nazionale della comunità Italiana di Eclipse.
E-mail: paomares@unina.it

LUISA MICH è professore associato in ingegneria informatica presso l'Università di Trento. Le attività di ricerca attuali si collocano nell'area dell'ingegneria dei requisiti e sono riconducibili a tre temi: qualità dei siti web, web semantico e tecniche di creatività. Fa parte del Gruppo di Lavoro che ha realizzato l'Osservatorio

Permanente delle Certificazioni Informatiche negli Atenei Italiani; è promotore dell'ECDL (European Computer Driving License) nell'Università di Trento, prima Università Italiana ad avviare tale iniziativa. Ha fatto parte del comitato di programma di conferenze e workshop nazionali e internazionali; è referee di riviste internazionali e ha pubblicato più di 140 lavori. Socio dell'AICA e membro delle associazioni IEEE Computer Society, ACM, IFITT (International Federation for Information Technology and Tourism).
E-mail: luisa.mich@economia.unitn.it

FULVIA SALA dopo aver conseguito un Master in Statistica presso l'Università di Berkeley, ha ricoperto ruoli direttivi nell'area del marketing e della pianificazione nel settore informatico. Collabora con AICA svolgendo attività di consulenza nelle stesse aree.
E-mail: fulvia.sala@aicanet.it

NELLO SCARABATTOLO è professore ordinario di Informatica presso il Dipartimento di Tecnologie dell'Informazione dell'Università di Milano, collabora da tempo con la Fondazione CRUI e con AICA su progetti di definizione, diffusione e monitoraggio delle certificazioni ICT nelle università italiane.

Fa parte del Gruppo di Lavoro che ha realizzato l'Osservatorio Permanente delle Certificazioni Informatiche negli Atenei Italiani. È Honorary Secretary del CEPIS (il Council of European Professional Informatics Societies) l'ente che riunisce le Associazioni europee di informatica, di cui AICA è la rappresentante per l'Italia.
E-mail: nello.scarabottolo@unimi.it