

Curriculum Scientifico e Didattico

Maria Cristina Pinotti

Affiliazione: Dipartimento di Matematica e Informatica, Università di Perugia
Sede lavoro: Via Vanvitelli 1, 06123 Perugia
Telefono: (+39) 075 5855055 (office)
E-mail: pinotti@unipg.it
Web: <http://www1.isti.cnr.it/~pinotti>
Residenza: Via A. Volta 29, 56017 Ghezzano, San Giuliano Terme, Pisa
Telefono: 050 877057, 335 8448245 (cellulare)
Nascita: Mantova, 11/08/1962

1 Carriera e Studi

Posizioni: Professore Straordinario (S.S.D. INF/01)
Facoltà di Scienze Matematiche Fisiche e Nat., Università di Perugia, dal 22/12/2004
Professore Associato (S.S.D. INF/01),
Facoltà di Scienze Mat. Fis. e Nat., Università di Perugia, dal 1/11/2003 al 21/12/2004
Facoltà di Scienze Mat. Fis. e Nat., Università di Trento, dal 1/09/2000 al 31/10/2003
Ricercatore,
I.E.I.-Consiglio Nazionale delle Ricerche, Pisa, dal 30/12/1987 al 31/08/2000
Centro Ricerche Olivetti, Pisa, da Marzo a Dicembre 1987
Laurea: Scienze dell'Informazione, Università di Pisa, 19/06/1986, *cum laude*

Collaborazioni Scientifiche

- Associato alla ricerca presso *Istituto di Scienza e Tecnologia dell'Informazione "A. Faedo"*, C.N.R., Pisa

2 Periodi di Ricerca all'Estero

06/2007 Visiting Researcher, Laboratoire d'Informatique de Paris-Nord,
University Paris 13, Villetaneuse, Paris, France
10/1998 - 11/1998 Visiting Researcher, Department of Computer Science,
Old Dominion University, Norfolk (VA), USA
05/1997 - 08/1997 Visiting Researcher, Department of Computer Science,
Old Dominion University, Norfolk (VA), USA,
beneficiaria di una borsa di studio del
Comitato per la Scienza e le Tecnologie dell'Informazione, *Bando n. 203.15.07*
05/1995 - 10/1995 Visiting Researcher, Department of Computer Science,
University of North Texas, Denton (TX), USA,
beneficiaria di una borsa di studio
del Comitato delle Scienze Matematiche, *Bando n. 203.01.63*
08/1994 - 12/1994 Visiting Researcher, Department of Computer Science,
University of North Texas, Denton (TX), U.S.A.

3 Attività Scientifica

Interessi di ricerca: Progettazione ed analisi di algoritmi con applicazioni all'organizzazione delle reti di sensori, reti wireless, e reti cellulari. Algoritmi per la biologia computazionale. Progettazione ed analisi di algoritmi e di strutture dati con applicazioni agli aspetti computazionali dei sistemi paralleli, dei sistemi distribuiti e dei sistemi VLSI.

Classificazione delle pubblicazioni per problematiche affrontate:

- **Progetto ed analisi di algoritmi per l'organizzazione di reti di sensori S1**
 - Algoritmi per la localizzazione [A1, C1, C2, C3, T3]
- **Progetto ed analisi di algoritmi per reti wireless e reti cellulari S2,S3,S4**
 - Allocazione di dati a canali wireless multipli per trasmissioni in diffusione [A3, A6, C4, C7, C10, C12, I1, I2, I5, T1]
 - Architetture per trasmissioni su canali wireless [A4, A5, A16, C5, C6, C8, C9, C11, C1, C18]
 - Allocations di serventi in infostation [A8, C15]
 - **Assegnamento di canali in reti non cablate** [A2, A7, A11, I4, I3, C13 C14, C17, C20]
 - Localizzazione di utenti in reti cellulari [A14, C32]
- **Progetto e analisi di algoritmi per la biologia computazionale**
 - Variazioni di un singolo nucleotide [A9]
- **Progetto ed analisi di algoritmi per sistemi paralleli, distribuiti e architetture VLSI**
 - Calcolo della mediana in sistemi paralleli e distribuiti [A10, C22]
 - Ordinamento in sistemi paralleli [A18, A23, C28, C33]
 - Calcolo delle somme prefisse in sistemi paralleli con bus riconfigurabili [A20, C26, C30]
 - Calcolo delle somme prefisse in sistemi paralleli con bus ottici [A17, C23, C25]
 - Architetture parallele VLSI per unità aritmetiche [A25, A27, A29, A30, A31, A32, A34, C43, T2]
 - Reti di interconnessione [C35]
 - Reti logiche [A36, A37]
- **Progetto ed analisi di strutture di dati**
 - Strutture di dati per basi di dati [A12, C24, C27]
 - Implementazione di strutture di dati in memorie a banchi [A13, A15, A19, I6, I7, C19, C21, C31, C34, C36, C37, C38, C44]
 - Code di priorità per reti a grado limitato [A22, A24, A26, C29, C39, C41]
 - Code di priorità per il modello Parallel Random Access Machines (PRAM) [A21, A28, A33, A35, C40, C42, C45, C46]

4 Sommario delle Pubblicazioni

4.1 Progetto ed analisi di algoritmi per l'organizzazione di reti di sensori

Algoritmi per la localizzazione. Le reti di sensori considerate sono costituite da una stazione base e da un insieme di sensori wireless distribuiti casualmente e densamente in uno spazio piano. I sensori sono sprovvisti di GPS, sono anonimi, e non sono manutenibili. Hanno energia limitata non rinnovabile e possono transire fra stato attivo (sveglia) e passivo (sonno). Essi possono comunicare fra loro solo entro un raggio limitato. La stazione base ha energia illimitata, è provvista di un'antenna isotropica il cui raggio massimo di trasmissione delimita l'area controllata dalla stazione stessa. La stazione base può variare il raggio di trasmissione e raggiungere in un solo hop tutti i sensori attivi la cui distanza dalla stazione base sia minore o uguale al raggio della trasmissione attuale. In questo modello, un sensore si dice localizzato quando conosce la propria posizione nel piano rispetto ad un sistema di coordinate polari, con origine nella stazione base, che divide l'area controllata in settori equiangolari e corone concentriche. In [A1, C3] si forniscono due protocolli sincroni per l'apprendimento della corona concentrica a cui appartiene il sensore. I protocolli proposti consistono di una fase centralizzata, gestita completamente dalla stazione base, seguita da una fase distribuita. Le computazioni di entrambi i protocolli possono essere viste come particolari visite di alberi. Nel primo protocollo si tratta di un albero binario, nel secondo di un albero q -ario. Le prestazioni dei protocolli sono misurate in termini del tempo complessivo richiesto per completare per tutti i sensori l'apprendimento della posizione e in termini dell'energia spesa da ciascun sensore durante l'apprendimento. Entrambi i protocolli migliorano le prestazioni di un algoritmo per la localizzazione nello stesso modello, precedentemente presentato in letteratura. In particolare, il secondo protocollo può richiedere un numero costante (e quindi ottimo) di transizioni fra lo stato attivo e passivo del sensore, senza peggiorare il tempo totale necessario affinché tutti i sensori apprendano la loro posizione. In [C1, C2], si propongono, per lo stesso modello di reti di sensori, protocolli asincroni per la localizzazione. Tali protocolli sono asincroni nel senso che ciascun sensore diventa attivo 'at random' e per tutta la durata del protocollo alterna periodi di sveglia a periodi di sonno di prefissata lunghezza, senza nessuna esplicita sincronizzazione con la stazione base. Durante l'algoritmo di apprendimento, la stazione base ciclicamente trasmette con raggio di trasmissione decrescente, iniziando il ciclo con il raggio della corona più esterna e terminando con quello della corona più interna. Precisamente, ad ogni trasmissione, la stazione trasmette l'identificatore della corona concentrica il cui raggio coincide con il raggio della trasmissione attuale. Il processo è ripetuto fino a che tutti i sensori non abbiano appreso. Le prestazioni in tempo ed energia dei protocolli asincroni sono state valutate sia analiticamente che sperimentalmente. Dato il comportamento ciclico sia dei sensori che della stazione base, i risultati analitici della complessità nel caso pessimo sono ottenuti sfruttando proprietà dell'aritmetica modulare. Infine, è studiata la complessità nel caso medio dei protocolli. Tale analisi è inclusa nella versione completa dei risultati sui protocolli asincroni per reti di sensori attualmente sottomessa a rivista [T3].

4.2 Progetto ed analisi di algoritmi per reti wireless e reti cellulari

Allocazione di dati a canali wireless multipli per trasmissioni in diffusione. In [A3, A6, I1, I2, I4, I5, C4, C10, C12, T1], si considera un'architettura per la diffusione di contenuti che diffonde ciclicamente N dati su K canali radio, in modo da minimizzare il tempo medio di

attesa degli utenti. Si adotta una allocazione sbilanciata dei dati ai canali, in base alla popolarità dei dati, con trasmissione ciclica dei dati assegnati al medesimo canale. Si forniscono in [A6, C12] vari algoritmi di programmazione dinamica che trovano la soluzione ottima. Se i dati hanno lunghezze identiche, si propone un algoritmo con complessità $O(NK \log K)$, abbassando la complessità del migliore algoritmo precedentemente noto. Se, invece, le lunghezze dei dati non sono identiche, si dimostra che il problema è NP-arduo per $K = 2$ e fortemente NP-arduo per K arbitrario, fornendo per $K = 2$ un algoritmo ottimo di complessità pseudo-polinomiale. Per K qualsiasi, sono proposte in [A3, I5, C10] varie euristiche, una delle quali trova velocemente soluzioni che si discostano al più del 2% da quelle ottime su istanze note nella letteratura. Tali euristiche sono generalizzate in [I1] al caso di canali soggetti ad errori di trasmissione non correggibili, modellati da una distribuzione geometrica. A seguito di una trasmissione erronea, l'utente scarta il dato corrotto e attende la prossima occorrenza dello stesso dato. Questo comportamento è ripetuto fino a che l'utente non riceve una copia integra del dato. Successivamente, in [I2, T1] è stato studiato il problema della diffusione di dati su canali soggetti a burst di errori, quali possono essere i canali wireless in presenza di fading. Il canale è stato modellato da una catena di Markov a due stati discreti, noto come modello semplificato di Gilbert-Elliot. In ciascun istante di tempo il canale può trovarsi o in uno stato *buono* nel quale trasmette correttamente o in uno stato *cattivo* nel quale è soggetto ad errori. I tempi medi di permanenza del canale nei due stati dipendono dalle probabilità di transizione fra gli stati. Anche per questo modello le euristiche proposte sono state valutate sperimentalmente. Le soluzioni proposte ben contrastano la presenza di piccole, ma realistiche, probabilità di errore in quanto il tempo medio di attesa percepito dell'utente è confrontabile con quello che si otterrebbe se fossero utilizzati canali non soggetti ad errori.

Architetture per trasmissioni su canali wireless. L'architettura di reti wireless, già presente attorno a noi, consiste in uno o più livelli di infrastrutture. Alla rete fisica, si sovrappone una infrastruttura per l'accesso, una infrastruttura per la distribuzione dei dati, e sempre più spesso un'infrastruttura per i servizi personalizzati. Ogni infrastruttura è concepita in modo distribuito ed è il frutto dell'integrazione di soluzioni assai diverse, dettate da criteri di opportunità tecnologiche, geografiche, e finanziarie. Nell'ambito dell'infrastruttura per i servizi, si studia l'organizzazione delle trasmissioni per la distribuzione di contenuti tramite un sistema di server, capace di trasmettere, per diffusione e su richiesta, dati ad un numero arbitrario di utenti. In particolare, si progettano architetture per la distribuzione dei contenuti che adeguano la modalità di trasmissione dei dati alla popolarità dei dati stessi. I dati popolari sono diffusi (*pushed*) con uno scheduling gestito autonomamente dal sistema di servers, mentre i dati di 'nicchia' sono trasmessi (*pulled*) solo su richiesta esplicita dell'utente. La modalità di trasmissione di un dato può variare se si modifica il valore di soglia della popolarità che separa i dati diffusi da quelli trasmessi. In [C5, C6, C8, C9], varie architetture sono state proposte, ciascuna che pone attenzione ad aspetti diversi del problema. Innanzitutto, le architetture studiate scelgono la soglia di separazione in modo che sia minimo il tempo che l'utente rimane in ascolto per ricevere il dato diffuso a cui è interessato, qualunque esso sia. Tale problema è stato risolto in modo ottimo quando si assuma un contesto statico in cui la popolarità dei dati è nota a priori. Le architetture proposte si differenziano, poi, a seconda di come gestiscono i dati trasmessi su richiesta. Nel caso di un solo canale di trasmissione, la soluzione più semplice fa seguire a ciascuna diffusione una trasmissione su richiesta, se ve ne è almeno una in attesa. Un'altra soluzione contempla un canale di ritorno dall'utente al server con banda non trascurabile per ge-

stire una coda di richieste per ciascun dato e seleziona ad ogni trasmissione il dato più richiesto. Tale selezione può tenere conto del comportamento (impazienza e/o pigrizia) degli utenti. Ad esempio, richieste troppo ravvicinate per lo stesso dato possono essere ignorate se si assume che provengono da un utente impaziente, mentre richieste sporadiche provenienti da un nuovo utente possono essere servite per privilegiare un nuovo cliente. Infine, è stata studiata una strategia on-line, che non conosce a priori le popolarità dei contenuti da diffondere, e utilizza canali multipli. I dati sono partizionati fra i canali. e ciascun canale è gestito con uno scheduling ibrido push-pull. Una sintesi dei risultati ottenuti è proposta in [A4]. Infine, in [A5, C11] si modellano e si valutano le prestazioni di un'architettura ibrida push-and-pull. In particolare, si determinano i tempi di risposta individuale per ogni item diffuso e il tempo necessario a svuotare la coda delle richieste da trasmettere. Infine, si estende l'algoritmo di scheduling in modo da ottimizzare la qualità del servizio sostenibile.

Assegnamento di canali in reti non cablate. Si considerano reti senza fili in cui possono presentarsi interferenze causate dalla trasmissione simultanea di pacchetti da parte di due stazioni diverse, ma vicine, che usino la stessa frequenza per comunicare. In [C20] si è studiato il problema di assegnare le frequenze alle stazioni base delle reti cellulari così che siano evitate le interferenze fra stazioni adiacenti, sia rispettata la distanza di riuso delle frequenze, e il problema sia risolto utilizzando il minor numero possibile di frequenze. In particolare, rappresentate le frequenze con colori (per semplicità indicati con numeri interi), le stazioni base della rete cellulare con i vertici di un grafo, e posto un arco fra due vertici se le stazioni sono una nell'area di trasmissione dell'altra, il problema si riduce ad un problema di colorazione di grafo in cui la differenza fra i colori assegnati a due vertici adiacenti deve essere di almeno 2, mentre lo stesso colore può essere riutilizzato in vertici a distanza almeno 3 o 4. Il problema di minimizzare i colori usati, o minimizzare l'intervallo delle frequenze usate, è NP-hard per generici grafi. Pertanto, sono proposte soluzioni ottime per famiglie speciali di grafi, quali anelli, alberi, reti esagonali e reti cellulari, che ben si prestano a modellare reti senza fili. Tali soluzioni calcolano in tempo costante o al più logaritmico la frequenza assegnabile a ciascuna stazione. Inoltre, in [A11, C17], si è generalizzato il problema di assegnamento come segue. Modellata la rete con un grafo, e dati la distanza di riuso s ed un vettore di separazione $[d(1), d(2), \dots, d(s-1)]$, il problema di allocare le frequenze consiste nell'assegnare ad ogni nodo x del grafo un colore $f(x)$ tale che lo scarto tra $f(y)$ ed $f(x)$ non scenda sotto $d(i)$ ogniquale volta la distanza (numero minimo di archi) tra le stazioni x ed y è uguale a i ed il massimo colore usato è minimizzato. Per reti regolari sono progettati algoritmi ottimi o approssimati di complessità polinomiale per distanza di riuso $s = 4$ e vettori di separazione $[1, 1, 1]$ e $[2, 1, 1]$; per distanza di riuso arbitraria s e vettori di separazione $[1, 1, \dots, 1]$ e $[2, 1, \dots, 1]$. In [A2], per reti modellate da alberi e da grafi di intervalli, sono presentate $L(\delta_1, \delta_2, \dots, \delta_t)$ -colorazioni, ottenute basandosi sulla nozione di vertici 'strongly-simplicial' (cioè vertici che formano con i propri vicini, visitati in un ordine opportuno, un grafo completo). Gli algoritmi proposti forniscono $L(\delta_1, \delta_2, \dots, \delta_t)$ -colorazioni α -approssimate e richiedono tempo polinomiale nel numero dei vertici del grafo e nella distanza di riuso. La costante α di approssimazione dipende da t e dal vettore di separazione. Inoltre, per grafi di intervalli unitari, è discusso un algoritmo che richiede tempo lineare nel numero di intervalli per calcolare una 3-approssimazione di una $L(\delta_1, \delta_2)$ -colorazione del grafo. In [A7], si presentano algoritmi efficienti per trovare $L(2, 1)$ -, $L(2, 1, 1)$ - e $L(1, \dots, 1)$ -colorazioni ottime di una griglia esagonale. Una griglia esagonale modella una rete in una regione piana, senza barriere geografiche, dove le stazioni base sono posizionate seguendo una tessellazione regolare

del piano. Tali griglie esagonali sono interessanti perché richiedono meno colori per soddisfare i vincoli di separazione di quelli richiesti da una griglia cellulare o quadrata con lo stesso numero di vertici. I lavori di rassegna [I3, I4] riassumono e riorganizzano questi e precedenti risultati nell'area dell'assegnamento di canali in reti non cablate.

Allocazioni di serventi in infostation. In [A8, C15], si affronta il problema di minimizzare il numero di serventi necessari in una *infostation* per servire on-line e senza ritardo le richieste degli utenti mobili che l'attraversano. Ogni richiesta è caratterizzata da una durata e da una categoria di servizio (ad es., email, fax, web browsing, etc.). Ogni servente può soddisfare simultaneamente al più k richieste di cui al più h_c dello stessa categoria c . Si mostra che il problema è NP-hard, e si propone un algoritmo on-line che fornisce un'approssimazione assoluta di $2 - h/k$, dove $h = \min h_c$. Si considerano generalizzazioni dove richieste e categorie sono caratterizzate da pesi (ampiezze di banda) ed i vincoli di capacità coinvolgono pesi anziché cardinalità. Anche in questi casi sono proposti algoritmi on-line che forniscono approssimazioni assolute.

Localizzazione di utenti in reti cellulari. L'obiettivo delle reti senza fili è quello di fornire ad un largo numero di utenti, per lo più mobili, connessioni ad alta banda, connessioni sempre e ovunque (senza limitazioni strutturali), e connessioni arricchite da informazioni legate al contesto (ossia legate al profilo dell'utente, o alla zona geografica da cui si connette, etc.). Il modello di architettura senza fili più diffuso fino a oggi prevede un livello di infrastruttura fissa, le cosiddette stazioni di base connesse tra loro da una rete a larga banda e distribuite su una vasta area geografica, a cui si connettono tanti e diversi dispositivi (telefonini, palmari, laptop, dispositivi per il rilevamento della posizione, dispositivi ad infrarossi, etc.) Un importante problema da affrontare è il disegno e l'analisi delle strategie di *tracking*, o localizzazione, degli utenti mobili. Le strategie più comuni sono *Always-Update* and *Never-Update*. La prima informa la stazione di base ogni volta che l'utente mobile si muove ed ha ovviamente alti costi di gestione, ma localizza velocemente l'utente mobile al tempo della chiamata. La seconda strategia paga l'assenza totale di rendez-vous fra l'utente e la stazione di base con un alto costo per localizzare l'utente mobile. Recentemente è proposta una nuova strategia di notifica *Reporting Center Strategy*. Il problema dei reporting centers è quindi definito come il problema di localizzare i reporting centers sulla rete cellulare in modo che il loro numero sia minimo e che il numero massimo di comunicazioni necessarie per rintracciare l'utente mobile sia limitato superiormente da una costante Z . In [A14, C32], il reporting center problem, che in generale è NP-hard, è risolto ottimamente con tecniche greedy e con tecniche di programmazione dinamica quando: (1) la rete cellulare è modellata da grafi di intersezione e Z è fissato a 2, (2) la rete cellulare è modellata da grafi propri di intersezione e Z assume un qualsiasi valore intero.

4.3 Progetto ed analisi di algoritmi per la biologia computazionale

Variazioni di un singolo nucleotide. Si è intrappreso lo studio di algoritmi per la biologia computazionale. In particolare, in [A9], nell'ambito dei problemi relativi alle variazioni dei singoli nucleotidi (Single Nucleotide Polymorphism SNP) si studia la complessità computazionale del problema di trovare un insieme di haplotipi di cardinalità minima capace di spiegare una famiglia data di genotipo.

4.4 Progetto ed analisi di algoritmi per sistemi paralleli, distribuiti e architetture VLSI

Calcolo della mediana in sistemi paralleli e distribuiti. In [A10, C22], è affrontato il problema della selezione del mediano da un insieme di $N = rs$ numeri, memorizzabili in una matrice $r \times s$. Il problema è risolto modificando l'algoritmo *columnsort* di Leighton, sostituendo un r -ordinatore i (un circuito di confrontatori che ordina r numeri) con un più semplice r -classificatore (un circuito di confrontatori che separa gli $r/2$ numeri più piccoli dagli $r/2$ più grandi). L'algoritmo proposto utilizza l' r -classificatore lo stesso numero $O(s)$ di volte che *columnsort* utilizza l' r -ordinatore. L'algoritmo può essere realizzato in hardware richiedendo tempo ottimo $O(s + \log r)$ e lavoro (cioè tempo \times confrontatori) $O(rs \log r + r^2 \log r)$. In particolare, si mostra che è possibile progettare un classificatore per $N = r \log r$ numeri che richiede tempo ottimo $O(\log r)$ e lavoro ottimo $O(r^2 \log r)$.

Ordinamento in sistemi paralleli In [A23, C33], un'architettura VLSI capace di ordinare insiemi di N dati è progettata utilizzando come modulo di base una rete di confrontatori per l'ordinamento di piccoli insiemi di p interi, con $p \ll N$. L'architettura, che simula l'algoritmo di Odd-even Merge Sort sostituendo ciascun confronto con l'ordinamento di p dati, richiede $O(\frac{N}{p} \log N)$ tempo. Tale prestazione in tempo è quasi ottima ed è indipendente dalle prestazioni in tempo della rete di confrontatori usata come elemento base. Una architettura VLSI per l'ordinamento di grandi agglomerati di dati, le cui prestazioni in tempo $O(\frac{N \log N}{p \log p})$ sono ottime, è discussa in [A18, C28]. Il modulo di base per ordinare gli insiemi di p elementi è la ben nota, e non ottima in tempo, Batcher's Bitonic Sort Network. Il risultato di maggior interesse è l'aver ottenuto un algoritmo con prestazioni ottime in tempo pur usando un elemento sub-ottimo per l'ordinamento di piccoli insiemi. L'algoritmo sfrutta una tecnica di sampling (estrazione di un campione) complessa e richiede sofisticate tecniche di pipeline per ottimizzare il tempo di esecuzione.

Calcolo delle somme prefisse in sistemi paralleli con bus riconfigurabili. In [A20, C26, C30], è considerata un'architettura VLSI fornita di bus molto corti capaci di trasmettere un segnale e di modificarlo, in accordo a particolari registri di stato, mentre lo propagano. Per tale architettura, si propone un'implementazione ottima, nel numero di broadcast richiesti, per eseguire il calcolo dei prefissi della somma di una stringa di bits. In particolare, si mostra che si può eseguire in tempo ottimo la somma dei prefissi di un numero elevato di bits anche quando si assuma un ritardo di propagazione del segnale sul bus proporzionale alla lunghezza del bus stesso.

Calcolo delle somme prefisse in sistemi paralleli con bus ottici. In [C23, C25], si affronta il problema del calcolo dei prefissi della somma di una stringa di bits in un array di processori connessi in riga ed in colonna con bus ottici molto semplici e non riconfigurabili. La soluzione proposta richiede un numero costante di operazioni e permette la soluzione efficiente di altre operazioni di base quali la selezione, l'ordinamento.

Alcune primitive di comunicazione (*point-to-point*, *broadcast*, *multicasting*) sono invece studiate per architetture con bus ottici a condivisione di tempo. Si sono considerati dapprima bus ottici a condivisione di tempo a tre linee, cioè bus ottici con una linea per i dati, e due linee per l'indirizzamento dei processori. La tecnica di indirizzamento basata sulla codifica unaria

dell'identificatore del processore (*coincident pulse technique*), di solito usata per questa architettura, è inadatta per collegare un elevato numero di processori. Si è pertanto proposto, in [A17, C23] un bus ottico, con un numero variabile di linee, capace di indirizzare un elevato numero di processori, in minor tempo sfruttando una codifica compatta dell'indirizzo del processore. Diversi schemi di indirizzamento sono proposti, i quali suggeriscono diversi compromessi tra il numero di linee richieste e il tempo di indirizzamento.

Architetture parallele VLSI per unità aritmetiche La tecnologia VLSI ha reso possibile la progettazione di architetture ad alte prestazioni da affiancare a sistemi tradizionali per la risoluzione di problemi specifici. I Sistemi Numerici Residui (RNS) per le loro naturali caratteristiche di modularità e di parallelismo e la loro capacità di elaborare sempre più velocemente informazioni numeriche sono usati per la progettazione di unità VLSI aritmetiche non convenzionali. Nei sistemi residui, gli interi sono rappresentati per mezzo di un insieme indipendente di cifre, e l'addizione e la moltiplicazione sono eseguite in parallelo cifra a cifra. Tuttavia, i sistemi residui rivelano i loro limiti in tutti quei calcoli che richiedono una conoscenza esplicita della grandezza del numero da processare. Di fatto, la grandezza di un numero è una funzione di tutte le cifre residue, e per essere calcolata è richiesta un'operazione intermodulare. Purtroppo, il tempo speso per convertire interi da un sistema residuo ad un sistema posizionale può essere più grande del tempo risparmiato nell'eseguire addizioni e moltiplicazioni. Per superare questi inconvenienti ed estendere le applicazioni dei sistemi residui, sono seguiti diversi approcci, quali: la parallelizzazione delle operazioni intermodulari più comuni (ad es., l'estensione di base, lo scaling e le conversioni tra sistema posizionale e sistema residuo) e l'introduzione dei sistemi numerici ibridi. In questo scenario, sono rivisti i due metodi di conversione da sistemi pesati a sistemi residui, ossia, l'espressione polinomiale del Teorema Cinese del Resto (CRT) e la conversione a Radici Miste (MRC). Nelle formulazioni classiche, il primo metodo può facilmente essere parallelizzato, ma richiede che tutti gli operandi siano rappresentati a pieno campo. Il secondo metodo mantiene la modularità delle operazioni residue, ma è naturalmente sequenziale e il risultato della conversione (un insieme di cifre a radici miste) può richiedere di essere ulteriormente manipolato prima di essere dato in input ad una nuova computazione (si pensi al caso in cui il risultato della conversione debba essere input di unità aritmetica convenzionale). Alla luce di queste considerazioni, una parallelizzazione della conversione MRC è proposta in [A31]. Essa calcola l'espressione polinomiale del CRT nel sistema a radici miste associato al sistema residuo evitando perciò computazioni a pieno campo. In [A32], si è poi presentata una nuova formulazione del CRT completamente parallela dove le operazioni sono tutte svolte in aritmetica binaria in campi di dimensioni non superiori a quelli residui, con l'eccezione di un singolo passo finale di aggiustamento. La progettazione di un'architettura per implementare questo algoritmo è discussa nello stesso lavoro e in [A30] si prova che ha complessità in area e tempo, misurata dalla funzione AT^2 , ottima. Infine, in [A27], sono affrontate le due operazioni intermodulari di estensione di base e di scaling, rispettivamente per espandere e contrarre il campo residuo. In questo caso, è affinato l'uso delle *look-up tables* per ottenere uno *scaling* esatto (senza approssimazione) ed una veloce estensione di base senza usare ridondanza. Ancora una volta, la bontà dell'algoritmo proposto è valutata in termini di complessità asintotica in area e tempo.

I Sistemi Numerici Ibridi (HNS's) rappresentano un intero con un indice di grandezza, espresso in posizionale, ed una parte residua. I sistemi residui rappresentano una buona soluzione quando sono *flessibili*, ossia quando è possibile passare velocemente da un sistema

prevalentemente residuo ad uno prevalentemente pesato. In questo modo, a seconda del tipo di operazioni da eseguire, si può optare dinamicamente per la rappresentazione più opportuna. In [A34] è proposta e valutata in area e tempo un'architettura che rende flessibile un sistema ibrido. Infine, l'efficienza per diverse operazioni aritmetiche, quali la moltiplicazione e l'elevamento a potenza migliora se la codifica degli operandi è ridondante. Sebbene codificare gli operandi in rappresentazioni minime ridondanti, ossia rappresentazioni ridondanti con il minimo numero di cifre diverse da zero, migliori l'efficienza delle operazioni aritmetiche, spesso le rappresentazioni ridondanti non-minime sono preferite perché la loro generazione richiede tempo costante mentre la generazione delle rappresentazioni minime ridondanti richiede tempo proporzionale al numero delle cifre da codificare. In [A25, C43], ricondotto il calcolo della codifica ad un'opportuna computazione dei prefissi la cui complessità asintotica in tempo è proporzionale al logaritmo del numero delle cifre da codificare, si è proposta un'implementazione VLSI di un algoritmo parallelo per la generazione di due rappresentazioni ridondanti minime, *Canonical Signed Digit* (CSD) e *Generalized Non-Adjacent* (GNAF). Sempre nell'ambito dello studio dei sistemi numerici residui, è affrontato il problema della decodifica dei codici aritmetici residui, da tempo apprezzati per la loro ridondanza minima. Purtroppo tali codici presentano l'inconveniente di una laboriosa e lenta decodifica per molteplicità non banale di errori ammessi. Per questo, in [A29], si è affrontato il problema generale della decodifica proponendo una nuova definizione della sindrome d'errore che, in casi particolari, ha semplificato la procedura di decodifica dell'errore. Infine, in T2, si studiano le proprietà dei codici prodotto (AN) in un sistema numerico a rappresentazione pesata. Sono date le condizioni necessarie e sufficienti per rilevare l'errore singolo e doppio e si mostra come tali codici possono essere usati in unità aritmetico-logiche standard che rappresentano i numeri relativi con notazione in complemento alla base. Infatti, sebbene i codici prodotto rappresentano l'unico serio esempio di codici aritmetici in sistemi a rappresentazione pesata, essi non hanno avuto fino ad ora impatto pratico perché sono stati caratterizzati da difficili e laboriose decodifiche dell'errore e perché sono stati definiti su un intervallo virtuale multiplo del generatore A . In contrasto, in questo lavoro si mostra che i codici prodotto possono lavorare in unità aritmetico-logiche standard, basate su una qualsiasi base b , e quindi in particolare sulla base 2. Inoltre, in tali unità standard, le condizioni di rilevazione/correzione dell'errore sono semplificate. In particolare, gli errori singoli possono essere facilmente rilevati concorrentemente all'esecuzione delle operazioni con limitato hardware aggiuntivo.

Reti di interconnessione. In [C35], si considera la rete di interconnessione *fat-tree*, che è modellabile da un albero le cui foglie funzionano da terminali di input/output, i cui nodi interni sono sottoreti con capacità di instradare messaggi. L'arco (i, j) che connette i nodi i and j della rete è capace di trasmettere simultaneamente tanti messaggi quanti sono i processori nel sottoalbero radicato nel nodo i . Per questo modello di interconnessione è analizzata la complessità asintotica in tempo dell'operazioni di broadcast e di somma dei prefissi.

Reti Logiche. Sempre nell'ambito della progettazione VLSI, si sono affrontati i problemi di ridurre l'area occupata dall'interconnessioni fra chips [A36] e quello di minimizzare l'area occupata da un *programmable logic array* (PLA) [A37]. In [A36], si studiano le proprietà algebriche di un nuovo insieme di operatori per variabili a p valori con $p \geq 3$. Quando $p = 2$, i nuovi operatori si riducono ai ben noti operatori booleani *AND* e *OR*. Si è affrontato poi il problema della sintesi in forma minima (cioè l'espressione algebrica che usa il minimo numero

di nuovi operatori) di un'arbitraria funzione $f : p \rightarrow p$. Più funzioni booleane definite su uno stesso insieme di variabili possono essere implementate da un unico *programmable logic array* (PLA), che consiste di due matrici, una di porte logiche AND ed una di porte logiche OR, ciascuna con tante righe quante le funzioni da sintetizzare e tante colonne quante le variabili in input. Tra le diverse tecniche proposte per progettare PLA di area minima, cioè PLA in cui la maggior parte delle porte AND e OR delle due suddette matrici sono attive, vi è quella del *multiple column folding* che permuta righe e colonne delle due matrici in modo che più variabili di input possano condividere la stessa colonna. Determinare l'array *folded* di area minima è un problema *NP-hard*, pertanto in [A37] è proposta un'euristica che seleziona la riga attuale da inserire nell'array folded guidata da due condizioni *greedy* che se si verificassero ad ogni selezione restituirebbero un array *folded* di area minima. Pertanto diciamo che la riga selezionata in accordo alle due condizioni è ottima. L'euristica proposta non usa la tecnica del backtracking e perciò la sua implementazione risulta più efficiente di quella di altre tecniche esistenti.

4.5 Progetto e analisi di strutture dati

Strutture di dati per Basi di Dati. In [A12, C24, C27], si studia l'allocazione in memoria delle collezioni di record multidimensionali (*cartesian product files*), strutture dati elementari per le basi di dati. Ciascun record è costituito da n campi, e l' i -esimo campo assume i valori nell'intervallo $[0, m_i - 1]$. Dato un record q con alcuni campi *don't care* ed un cartesian product file F , risolvere la *partial match query* associata a q (o semplicemente, la *partial match query* q) significa accedere in F a tutti i records che si *qualificano* per q , ossia tutti i records che assumono qualsiasi valore nei campi *don't care* di q e coincidono con q per i restanti campi. Spesso data la mole di dati immagazzinati e il basso costo della memoria secondaria, i cartesian product files sono distribuiti fra più dischi. Risolvere efficientemente una partial match query q in F è allora equivalente ad accedere senza conflitti al sottoinsieme di F che si qualifica per q . Il problema si riduce alla definizione di un mapping di F su più dischi tale da garantire l'accesso senza conflitti a sottoinsiemi opportuni di F stesso. La soluzione proposta è basata sui codici residui e risolve ottimamente ogni partial match query.

Implementazione di strutture di dati in memorie a banchi. In un sistema multiprocessore, la memoria è una risorsa critica perché condivisa. Per servire simultaneamente più richieste la memoria è spesso organizzata in più banchi. Il rapporto fra il numero di banchi di memoria disponibili e il numero di processori è detto *fattore di espansione*, e la sua importanza è tale da essere stato incluso tra i parametri di una recente estensione di uno dei più noti modelli per sistemi paralleli, il *Bulk Synchronous Parallel Model (BSPM)*. Sistemi a memoria condivisa disponibili in commercio, come NEC SX-3 e il CRAYJ90, hanno configurazioni di base con da 4 a 16 processori e 1024 moduli di memoria, mentre il Tera MTA ha 26 processori e 2^{15} moduli. Anche un fattore di espansione pari ad 1, non esclude che in uno stesso ciclo di CPU occorran *conflitti*, ossia che siano richiesti più dati tutti memorizzati nello stesso banco di memoria. Se ciò accade, i tempi di risposta della memoria degradano. Risultati sperimentali hanno provato che se il fattore di espansione è sufficientemente grande e l'accesso alla memoria è irregolare, distribuire i dati fra i banchi di memoria con un *random mapping* è sufficiente per minimizzare i conflitti e quindi per migliorare la prestazione dell'intero sistema. Tuttavia, se, la memoria è acceduta secondo schemi regolari, distribuzioni sofisticate, disegnate ad hoc per particolari tipi di accessi, si rendono utili. Non è difficile, ad esempio, immaginare applicazioni

in cui si richieda di accedere senza conflitti colonne, righe di matrici, cammini o sottoalberi di alberi. Le tecniche più innovative sviluppate in questo contesto e una rassegna di tutti i risultati ottenuti sono presentati in [A15, A19, I6, I7]. In particolare, è formalizzato in [I6] il problema dell'accesso senza conflitti uniformando la terminologia in uso e riassumendo i risultati noti in letteratura. Inoltre si sono proposti mapping ad hoc per accedere efficientemente in parallelo le seguenti strutture:

- cammini di lunghezza variabile di matrici, anelli e alberi completi [A13, C21],
- cammini e sottoalberi di alberi completi [C31],
- sottografi connessi di grafi arbitrari [C34].
- sottoalberi di alberi binomiali [C36],
- sottoalberi di alberi binari completi, sottocubi di ipercubi e di ipercubi [C37],
- sottoalberi di alberi k -ari completi [C38],
- cammini dalle foglie alla radice di alberi completi [C44],

Oltre a studiare caso per caso il lower bound al numero di banche di memoria per accedere senza conflitti alle varie sottostrutture dati, i mapping proposti garantiscono, per una fissata configurazione di memoria, il minimo numero di conflitti. In generale, essi sono *bilanciati*, cioè distribuiscono in modo equo la struttura dati fra i processori, *diretti*, cioè calcolano, localmente (senza conoscenza dell'intera struttura) ed in tempo costante, il modulo a cui ciascun elemento della struttura è assegnato. Infine, in [A15, C19], si studiano mapping *versatili* che minimizzano il numero di conflitti sia per cammini che per sottoalberi. Le tecniche usate in [C21] per derivare lower and upper bounds per il numero di banche di memoria necessari per l'accesso senza conflitti sono di particolare interesse. In [A19], sono riportati la maggior parte dei risultati di [C36, C37, C38, C44].

Code di priorità per reti a grado limitato. Lo studio di realizzazioni di strutture dati in reti di processori a memoria distribuita, come è discusso in [C41], ha sollevato diversi interessanti problemi. Infatti, a differenza del modello PRAM dove il tempo di accesso alla memoria è assunto indipendente dalla locazione richiesta, nei modelli distribuiti locazioni di memoria diverse hanno tempi di accesso diversi da processore a processore. Infatti, il tempo di accesso alla memoria locale del processore P_j da parte del processore P_i è proporzionale alla minima distanza fra P_j e P_i . È quindi importante distribuire la struttura dati fra le memorie dei processori in modo che le comunicazioni tra processori per la gestione dei dati siano minimizzate. Inoltre, poiché la struttura dati varia nel tempo, la distribuzione deve poter essere preservata dinamicamente. In [A26], si è mappato su hypercubes una coda di priorità, basato sul tipo astratto Parallel Priority Queues, garantendo che ogni nodo dell'architettura gestisca in memoria la stessa quantità di dati ed esegua lo stesso carico di lavoro ad ogni operazione.

Un mapping per min-max heaps su hypercubes è poi discusso in [A24, C39] che si basa sul calcolo dei suffissi di un vettore V di n elementi distribuiti su un cammino Hamiltoniano dell'hypercube di dimensione h .

Una nuova struttura dati, *k-Parallel Union Find Trees*, è proposta in [C42], per la gestione dinamica di una partizione di un insieme. Su tale partizione si possono eseguire operazioni di ricerca di elementi, aggregazioni di sottoinsiemi e recuperare lo stato della partizione in istanti di tempo precedenti. Per tutte le operazioni, eccetto la ricerca di un elemento, è richiesto tempo parallelo costante.

In [A22, C29], si studia un'implementazione di code di priorità per le reti di confrontatori, uno tra i più semplici modelli paralleli. Le reti di confrontatori eseguono solo confronti fra dati, e tali confronti possono essere eseguiti in parallelo se non hanno dati in comune. Le reti più studiate sono le reti per la fusione, per l'ordinamento e per la selezione. In [A22, C29], è proposta una rete ottima per la costruzione di uno heap. Essa richiede $O(n \log \log n)$ confrontatori per la costruzione di uno heap di dimensione n , ed ha profondità $O(\log n)$. È il primo esempio di rete di confrontatori che richiede un numero di confrontatori $c(n) \in (\Omega(n), o(n \log n))$.

Code di priorità per il modello Parallel Random Access Machines (PRAM). Una struttura dati parallela può o basarsi su un tipo di dato astratto nuovo che permetta di eseguire ad ogni operazione parallela più operazioni sequenziali simultaneamente (ossia, una operazione per ciascun processore) o mantenere lo stesso tipo di dato astratto definito per l'ambiente sequenziale ma eseguire in parallelo le operazioni come definite in ambiente sequenziale (ossia, tutti i processori cooperano all'esecuzione della stessa operazione). In [A35, C46], è introdotto il tipo di dato astratto *Parallel Priority Queues (PPQ)*, che memorizza un insieme di valori interi e permette inserzioni o cancellazioni multiple. Le inserzioni e le cancellazioni multiple, cioè operazioni in cui ogni processore del sistema simultaneamente inserisce o cancella un dato della struttura, sono peculiari del nuovo tipo di dato astratto, pensato ad hoc per ambienti paralleli. Due implementazioni del dato astratto PPQ, *n-Bandwidth-Heap* e *n-Bandwidth-Leftist-Heap*, sono proposte per il modello computazionale parallelo a memoria condivisa, Concurrent-Read-Exclusive-Write Parallel Random Access Machines (CREW-PRAM), con accesso concorrente in lettura da memoria. Nel modello di computazione parallelo a memoria condivisa Exclusive-Read-Exclusive-Write PRAM (EREW-PRAM), con accesso alla memoria esclusivo sia in lettura che in scrittura, è discussa in [A28, C45] la gestione parallela dei tradizionali heap binari ottenendo algoritmi paralleli per l'inserzione e la cancellazione di un singolo elemento dalla struttura che eseguono la stessa quantità di lavoro della computazione sequenziale ma in tempo proporzionale al logaritmo del tempo richiesto dalla stessa. Il risultato è di particolare interesse per l'operazione delete-min che appare, a prima vista, intrinsecamente sequenziale. Attenzione particolare, in [A33], è rivolta all'operazione di costruzione di uno heap tradizionale nel modello EREW-PRAM, mostrando la non correttezza di un algoritmo noto in letteratura per questo problema, e proponendone uno nuovo ed efficiente.

Infine, si è studiata in [A21, C40] la gestione parallela degli heap binomiali. Oltre ad ottenere algoritmi ottimi e veloci per le consuete operazioni su code di priorità (i.e., inserzione, estrazione del minimo, unione), si sono proposti algoritmi per le operazioni *decrease-key* e *delete-key*. Lo studio degli heap binomiali pone in evidenza come approcci totalmente diversi da quelli usati nell'ambiente sequenziale debbano essere escogitati in parallelo per la gestione di una struttura dati se l'efficienza è uno degli obiettivi da perseguire. Al riguardo, di particolare interesse è la fusione di due heap binomiali che in parallelo è eseguita simulando l'esecuzione parallela del calcolo dei riporti in una somma di due numeri binari. Una nuova struttura dati, *k-Parallel Union Find Trees*, è proposta in [C42], per la gestione dinamica di una partizione di un insieme. Su tale partizione si possono eseguire operazioni di ricerca di elementi, aggregazioni di sottoinsiemi e recuperare lo stato della partizione in istanti di tempo precedenti. Per tutte le operazioni, eccetto la ricerca di un elemento, è richiesto tempo parallelo costante.

5 Attività Didattica

Insegnamenti

- In qualità di docente dell'Università di Perugia:

A.A. 2007 – 08	“Algoritmi e Strutture Dati 1”	Laurea Triennale in Informatica
A.A. 2006 – 07	“Algoritmi e Strutture Dati 1”	Laurea Triennale in Informatica
A.A. 2006 – 07	“Algoritmi e Strutture Dati 2”	Laurea Triennale in Informatica
A.A. 2005 – 06	“Algoritmi e Strutture Dati 1”	Laurea Triennale in Informatica
A.A. 2005 – 06	“Algoritmi e Strutture Dati 2”	Laurea Triennale in Informatica
A.A. 2005 – 06	“Algoritmi 3”	Laurea Specialistica in Informatica
A.A. 2005 – 06	“Basi di Dati e Sistemi Informativi Avanzati”	Master in Sistemi Informativi per la Pubblica Amministrazione
A.A. 2004 – 05	“Algoritmi per Reti Wireless”	Laurea Specialistica in Informatica
A.A. 2004 – 05	“Basi di Dati Distribuite e Tecniche di Data Mining”	Laurea Specialistica in Informatica
A.A. 2004 – 05	“Algoritmi e Strutture Dati 2”	Laurea Triennale in Informatica
A.A. 2003 – 04	“Basi di Dati e Sistemi Informativi 1”	Laurea Triennale in Informatica
A.A. 2003 – 04	“Basi di Dati e Sistemi Informativi 2”	Laurea Triennale in Informatica
A.A. 2003 – 04	“Basi di Dati e Sistemi Informativi Avanzati”	Corso di Alta Formazione “Specialista in Reti di Computer”

- In qualità di docente dell'Università di Trento:

A.A. 2002 – 03	“Algoritmi e Strutture Dati 2”	Laurea Triennale in Informatica
A.A. 2002 – 03	“Algoritmi Avanzati ”	Laurea Specialistica in Informatica
A.A. 2001 – 02	“Informatica Generale 2 U.D.”	Laurea in Matematica
A.A. 2001 – 02	“Architettura degli Elaboratori 1”	Laurea Triennale in Informatica
A.A. 2001 – 02	“Architettura degli Elaboratori 2”	Laurea Triennale in Informatica
A.A. 2001 – 02	“Algoritmi e Strutture Dati 2”	Laurea Triennale in Informatica
A.A. 2000 – 01	“Informatica Generale 2 U.D.”	Laurea in Matematica
A.A. 2000 – 01	“Architettura degli Elaboratori 1”	Laurea Triennale in Informatica
A.A. 2000 – 01	“Architettura degli Elaboratori 2”	Laurea Triennale in Informatica

- In qualità di Ricercatore C.N.R.:

A.A. 1996 – 97	Prof. a contratto <i>ex art. 100 (382/80)</i> di “Algoritmi e Strutture Dati”, presso il corso di Diploma Universitario in Informatica, Università di Perugia
A.A. 1995 – 96	
A.A. 1994 – 95	Prof. a contratto <i>ex art. 25 (382/80)</i> di “Strutture Dati ed Algoritmi di Base”, presso il Corso di Laurea in Scienze dell'Informazione, Università di Pisa
A.A. 1993 – 94	
A.A. 1992 – 93	Prof. a contratto <i>ex art. 25 (382/80)</i> di “Complem. di Strutture Dati ed Algoritmi”, presso il Corso di Laurea in Scienze dell'Informazione, Università di Pisa
A.A. 1991 – 92	Prof. a contratto <i>ex art. 25 (382/80)</i> di “Attività Didattiche a carattere Tecnico-Pratico connesse all'insegnamento di Introduzione alla Programmazione”, presso la Scuola Diretta a Fini Speciali in Informatica, Università di Pisa
A.A. 1990 – 91	

Tesi

- Ha seguito la tesi di dottorato presso il Dipartimento di Informatica e delle Telecomunicazioni dell'Università di Trento della Dr.ssa Saxena Navrati, ora Research Professor at the School of Information and Communication Engineering, Sungkyunkwan University, Suwon, South Korea.

Relatore di numerose tesi di laurea, ne ha curato in vari casi il loro sviluppo fino alla pubblicazione su riviste o atti di congressi internazionali. In particolare:

Candidato	Titolo	Pubblicazione	Anno
R. Ciotti	"Flat: Algoritmi per la localizzazione . . ."	C2, T3	2007
Stefano Anticaglia	"Greedy: un'euristica efficiente . . ."	A3	2005
Lucio Iamele	"Euristiche per la trasmissione broadcast . . ."	A3	2005

6 Attività Professionale

Recentemente, ha svolto la seguente attività professionale:

Attività Editoriale

- E' editor dal Novembre 2004 della rivista **International Journal of Parallel, Emergent, and Distributed Systems**
http://www.informaworld.com/smpp/title_content=t713729127
- Editor di numeri speciali di riviste [S1,S2, S3, S4]

Responsabile dei Programmi Scientifici

- Algorithms and Applications Track, *The Fifth International Symposium on Parallel and Distributed Processing and Applications (ISPA07)*, August 29 – Friday, August 31, 2007 Niagara Falls, ON, CANADA
- Communication Networks Track, *12th Annual IEEE International Conference on High Performance Computing (HiPC05)*, Goa, India, December 18 – 21, 2005
- *1st Workshop on AlgorithmS for Wireless and Ad-hoc networks (A-SWAN)*, satellite di "Mobiquitous 2004" Boston, August 26, 2004
- *4rd Workshop on Wireless, Mobile, Ad Hoc and Sensor Networks (WMAN)*, satellite di "International Parallel and Distributed Processing Symposium" (IPDPS2004), Santa Fe, April 26-30, 2004

Partecipazione a Comitati di Programma di Conferenze/Workshop

- *IEEE International Symposium on Parallel and Distributed Processing and Applications (ISPA'08)*, Sydney, December 10-12, 2008
- *Second ACM Workshop on Sensor Actor Networks (SANET'08)*, Hong Kong, May 26-30, 2008

- *Fifth IEEE International Conference on Mobile Ad-hoc and Sensor Systems (IEEE MASS 2008)*, Atlanta (GA), USA, October 6-9, 2008
- *International Conference on Sensor Technologies and Applications SENSORCOMM 2008*, Cap Esterel, France, August 25-31, 2008 - Cap Esterel, France
- *IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS)*, (Publicity Chair) Santorini Island, Greece, June 11-14, 2008
- *Fourth International Workshop on Sensor Networks and Systems for Pervasive Computing*, to be held in conjunction with IEEE Percom 2008, Hong Kong, March 17-21, 2008
- *The 2007 IFIP International Conference on Embedded and Ubiquitous Computing (EUC 2007)*, Taipei, Taiwan, December 17-20, 2007
- *3rd International Conference on Mobile Ad-hoc and Sensor Networks (MSN'07)* (Awards Co-chairs), Beijing, China, December 12-14, 2007
- *International Conference on Sensor Technologies and Applications SENSORCOMM 2007*, Valencia, Spain, October 14-20, 2007.
- *Second International Conference on Grid and Pervasive Computing (GPC 2007)*, Paris, May 2-4, 2007.
- *First International Conference on Grid and Pervasive Computing (GPC 2006)*, Taichung City, Taiwan, May 3-5, 2006.
- *IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS)*, Marina del Rey, California, USA, June 30-July 1, 2005.
- *Symposium on Mobile Computing*, to be held in conjunction with *IEEE WirelessCom 2005*, Hawaii, USA, June 13-16, 2005
- *ACS/IEEE International Conference on Computer Systems and Applications (AICCSA-05)*, Egypt, January 3-6, 2005.
- *1st IEEE Int'l Conference on Mobile Ad-Hoc and Sensor Systems (IEEE MASS)*, Fort Lauderdale, Florida, October 25-27, 2004.
- *Workshop Online algorithms for mobile wireless computing and networking* held in conjunction with *IEEE GLOBECOM 2004*, Dallas, 2004.
- *Int'l 2004 Workshop on High Performance Switching and Routing (HPSR 2004)*, Phoenix, Arizona, USA, April 18-21, 2004.

Attività di Revisione:

Revisore per numerose riviste internazionali, tra cui: IEEE Transactions on Parallel and Distributed Systems, IEEE Transactions on Computers, IEEE Transactions on Circuits and Systems, Journal of Parallel and Distributed Computing.

Partecipazione a Progetti di Ricerca

- 2005 – 2006: Progetto ISTI-CNR “BREW-ERROR”, Responsabile: M.C. Pinotti

7 Elenco Pubblicazioni

S: Editore di Numeri Speciali di Riviste

1. Y-C. Tseng, W-C. Peng, V.C.M. Leung, W-T. Chen, & C.M. Pinotti, (Editor) “Information Processing and Data Management in Wireless Sensor Networks”, *Signal Processing*, (Elsevier), Vol. 87, No. 12, December 2007, 2859-2860.
2. A.A. Bertossi, A. Boukerche & M.C. Pinotti, (Editor) “Special Issue on WMAN04 Best Papers”, *Wireless Networks*, Vol. 12, No. 6, December 2006, 669-731.
3. A. A. Bertossi, S. Olariu, & M.C. Pinotti, (Editor) “Special Issue: Algorithms for wireless and ad-hoc networks”, *Journal of Parallel and Distributed Computing*, Vol. 66, No. 4, April 2006, 487-488.
4. A. Bar-Noy, A.A. Bertossi, M.C. Pinotti, & C.S. Raghavendra, (Editor) “Foreword: Special Issue on Algorithmic Solutions for Wireless, Mobile, Ad Hoc and Sensor Networks”, *Mobile Networks and Applications*, Vol. 10, No. 1-2, 2005.

A: Riviste Internazionali

1. A.A. Bertossi, S. Olariu, & C.M. Pinotti, “Efficient Corona Training Protocols for Sensor Networks”, *Theoretical Computer Science*, in stampa.
2. A. A. Bertossi & C.M. Pinotti, “Approximate $L(\delta_1, \delta_2, \dots, \delta_t)$ -Coloring of Trees and Interval Graphs”, *Networks*, Vol. 49, No. 3, 2007, 204-216.
3. S. Anticaglia, F. Barsi, A.A. Bertossi, L. Iamele & M.C. Pinotti, “Efficient Heuristics for Data Broadcasting on Multiple Channels”, *Wireless Networks*, on-line 2006, (printed *Wireless Networks* 14:2 April 2008).
4. N. Saxena, C.M. Pinotti, K. Basu & S.K. Das, “A Dynamic Hybrid Scheduling Algorithm for Heterogeneous Asymmetric Environments”, *The International Journal of Parallel, Emergent and Distributed Systems*, Vol. 20, No. 3-4, September-December 2005, 185-204.
5. A. Boukerche, T. Dash, & C.M. Pinotti “Performance analysis of a novel hybrid push-pull algorithm with QoS adaptations in wireless networks”, *Performance Evaluation*, Vol. 60, 2005, 201-221.
6. E. Ardizzoni, A.A. Bertossi, M.C. Pinotti, S. Ramaprasad, R. Rizzi, & M.V.S. Shashanka, “Optimal Skewed Data Allocation on Multiple Channels with Flat Broadcast per Channel”, *IEEE Transactions on Computers*, Vol. 54, No. 5, 2005, 558-572.
7. A.A. Bertossi, M.C. Pinotti, R. Rizzi, & A.M. Shende “Channel Assignment for Interference Avoidance in Honeycomb Wireless Networks”, *Journal of Parallel and Distributed Computing*, Vol. 64, No. 12, 2004, 1329-1344.
8. A.A. Bertossi, M.C. Pinotti, R. Rizzi, & P. Gupta, “Allocating Servers in Infostations for Bounded Simultaneous Requests”, *Journal of Parallel and Distributed Computing*, Vol. 64, No. 10, 2004, 1113-1126.

9. G. Lancia, M.C. Pinotti & R. Rizzi, "Haplotyping Populations by Pure Parsimony: Complexity, Exact and Approximation Algorithms", *INFORMS Journal on Computing*, Vol. 16, No. 4, 2004, 348-359.
10. A.A. Bertossi, S. Olariu, M.C. Pinotti & S.Q. Zheng, "Selection on Matrices Classifying Rows and Columns", *IEEE Transactions on Parallel and Distributed Systems*. Vol. 15, No. 7, 2004, 654-665.
11. A.A. Bertossi, M.C. Pinotti & R. Tan, "Channel Assignment with Separation for Interference Avoidance in Wireless Networks", *IEEE Transactions on Parallel and Distributed Systems*, Vol. 14, No. 3, 2003, 222-235.
12. S.K. Das and M.C. Pinotti, "Load Balanced and Optimal Disk Allocation Strategy for Partial Match Queries on Multi-dimensional Files", *IEEE Transactions on Parallel and Distributed Systems*, Vol. 13, No. 12, 2002, 1320-1332.
13. A.A. Bertossi and M.C. Pinotti, "Mappings for Conflict-Free Access of Paths in Bidimensional Arrays, Circular Lists, and Complete Trees", *Journal of Parallel and Distributed Computing*, Vol. 62, 2002, 1314-1333.
14. S. Olariu, M.C. Pinotti & L. Wilson, "Greedy Algorithms for Tracking Mobile Users in Special Mobility Graphs", *Discrete Applied Mathematics*, Vol 117/1-3, 2002, 215-227.
15. V. Auletta, S.K. Das, A. De Vivo, M.C. Pinotti, & V. Scarano, "Optimal Tree Access by Elementary and Composite Templates in Parallel Memory Systems," *IEEE Transactions on Parallel and Distributed Systems*, Vol. 13, No. 4, 2002, 399-412.
16. Y. Guo, S.K. Das & M.C. Pinotti, "A New Hybrid Broadcast scheduling Algorithm for Asymmetric Communication Systems: Push and Pull Data based on Optimal Cut-Off Point", *Mobile Computing and Communications Review (MC2R)*, Vol. 5, No. 4, 2001.
17. S.Q. Zheng, K.Li, Y. Pan, & M.C. Pinotti "Generalized Coincident Pulse Technique and New Addressing Schemes for Pipelined Time-Division Multiplexing Optical Buses", *Journal of Parallel and Distributed Computing*, Vol. 61, No. 8, 2001, 1033-1051.
18. S. Olariu, M.C. Pinotti & S.Q. Zheng, "An Optimal Hardware-Algorithm for Sorting Using a Fixed-Size Parallel Sorting Device", *IEEE Transactions on Computers*, Vol. 49, No. 12, 2000, 1310-1324.
19. S.K. Das & M.C. Pinotti, "Optimal Mappings of q -ary and Binomial Trees into Parallel Memory Modules for Fast and Conflict-Free Access to Path and Subtree Templates", *Journal of Parallel and Distributed Computing*, Vol. 60, No. 8, 2000, 998-1027.
20. R. Lin, S. Olariu, K. Nakano, M.C. Pinotti, J.L. Schwing, & A. Y. Zomaya, "Scalable Hardware-Algorithms for Binary Prefix Sums", *IEEE Transactions on Parallel and Distributed Systems*, Vol. 11, No. 8, 2000, 838-850.
21. S.K. Das & M.C. Pinotti, "Parallel Priority Queues Based on Binomial Heaps", *Parallel Computing*, Vol. 26, 2000, 1411-1428.

22. G. Brodal & M.C. Pinotti, "Comparator Networks for Binary Heap Construction", *Theoretical Computer Science*, Vol. 250/1-2, 2000, 235-245.
23. S. Olariu, M.C. Pinotti & S.Q. Zheng, "How to sort N Items Using a Network of Fixed I/O", *IEEE Transactions on Parallel and Distributed Systems*, Vol. 10, No. 5, 1999, 487-499.
24. S.K. Das & M.C. Pinotti, " $O(\log \log n)$ Time Algorithms for Hamiltonian-Suffix and Min-Max-Pair Heap Operations on the Hypercube", *Journal of Parallel and Distributed Computing*, Vol. 48, No. 2, 1998, 200-211.
25. S.K. Das & M.C. Pinotti, "Fast VLSI Circuits for CSD-Coding and GNAF-Coding", *Electronics Letters*, Vol. 32, No. 7, 1996, 632-634.
26. S.K. Das, M.C. Pinotti, & F. Sarkar, "Optimal and Load Balanced Mapping of Parallel Priority Queues in Hypercubes", *IEEE Transactions on Parallel and Distributed Systems*, Vol. 7, No. 6, 1996, 555-564.
27. F. Barsi & M.C. Pinotti, "Fast Base Extension and Precise Scaling in RNS for Look Up Table Implementations", *IEEE Transactions on Signal Processing*, Vol. 43, No. 10, 1995, 2427-2430.
28. M.C. Pinotti & G. Pucci, "Parallel Algorithms for Priority Queue Operations", *Theoretical Computer Science*, Vol. 148, 1995, 171-180.
29. F. Barsi & M.C. Pinotti, "Efficient Error Correcting Technique for Digital Equipment", *Electronic Letters*, Vol. 31, No. 3, 1995, 158-159.
30. F. Barsi & M.C. Pinotti, Addendum to "A Fully Parallel Residue-to-Binary Conversion", *Information Processing Letters*, Vol. 55, No. 1, 1995, 25-26.
31. F. Barsi & M.C. Pinotti, "Time Optimal Mixed Radix Conversion for Residue Number Applications", *The Computer Journal*, Vol. 37, No. 11, 1994, 907-916.
32. F. Barsi & M.C. Pinotti, "A Fully Parallel Residue-to-Binary Conversion", *Information Processing Letters*, Vol. 50, No. 1, 1994, 1-8.
33. C. Luchetti & M.C. Pinotti, "Some Comments on Building Heaps in Parallel", *Information Processing Letters*, Vol. 47, 1993, 145-148.
34. F. Barsi & M.C. Pinotti, "Adding Flexibility to Hybrid Number Systems", *The Computer Journal*, Vol. 35, No. 6, 1992, 630-635.
35. M.C. Pinotti & G. Pucci, "Parallel Priority Queues", *Information Processing Letters*, Vol. 40, 1991, 33-40.
36. F. Luccio & M.C. Pinotti, "Minimal Synthesis of Multivalued Functions with New Operators", *IEE Proceedings Part E: Computer and Digital Techniques*, Vol. 138, No. 6, 1991, pp. 419-423.
37. F. Luccio & M.C. Pinotti, "Suboptimal Solution for PLA Multiple Column Folding", *Computer Aided-Design*, Vol. 22, No. 8, 1990, 515-520.

I. Capitoli su Invito in Volumi Internazionali

1. P. Barsocchi, A.A. Bertossi, M.C. Pinotti, & F. Potortì, “Quality of Service of data broadcasting algorithms on erroneous wireless channels”, in: *Handbook of Research on Mobile Multimedia* (Ed. Ismail K. Ibrahim), IGI Publishing, in stampa.
2. P. Barsocchi, A.A. Bertossi, M.C. Pinotti, & F. Potortì, “Data broadcasting algorithms on error-prone wireless channels”, in: *NATO Security Through Science Series*, IOS Press, Amsterdam, in stampa.
3. Alan A. Bertossi & M.C. Pinotti, “Channel assignment with separation in wireless networks based on regular plane tessellations”, in: *NATO Security Through Science Series*, IOS Press, Amsterdam, in stampa.
4. Alan A. Bertossi & M.C. Pinotti “Channel Assignment in Wireless Local Networks”, in: *Wireless Ad Hoc Networking: Personal-Area, Local-Area, and Sensory-Area Networks* (Ed. Yu-Chee Tseng) Auerbach Publications, Taylor & Francis, June, 2007, pp. 277-299.
5. Alan A. Bertossi, M.C. Pinotti & R. Rizzi, “Scheduling Data Broadcasts on Wireless Channels: Exact Solutions and Heuristics”, *Handbook of Approximation Algorithms and Metaheuristics* (Ed. Teofilo Gonzalez) Taylor & Francis Books (CRC Press), May, 2007, pp. 73.1-73.16.
6. S.K. Das & M.C. Pinotti, “Distributed Data Access in Tree-Like Structures and Multidimensional Vector Spaces – A Survey”, *Distributed Data and Structures*, (Eds. N. Santoro & P. Widmayer), Carleton Scientific Pub., 1999, 21-42.
7. S.K. Das & M.C. Pinotti, “Efficient Schemes for Distributing Data on Parallel Memory Systems”, *AAMS-DIMACS Series on Discrete Mathematics and Theoretical Computer Science, External Memory Algorithms*, (Eds. J. M. Abello & J. S. Vitter), Vol. 50, 1999, 233–245.

C. Atti di Congressi Internazionali

1. F. Barsi, F. Betti Sorbelli, R. Ciotti, M.C. Pinotti, A.A. Bertossi, & S.Olariu, “Asynchronous Training in SANET”, *First ACM Workshop on Sensor Actor Networks*, Montreal, Canada, September 10, 2007.
2. F. Barsi, A.A. Bertossi, F. Betti Sorbelli, R. Ciotti, S. Olariu, & M.C. Pinotti, “Asynchronous Training in Wireless Sensor Networks”, *3rd International Workshop on Algorithmic Aspects of Wireless Sensor Networks (ALGOSENSORS 2007)*, July 14, 2007, Wroclaw, Poland, in stampa in LNCS.
3. A. A. Bertossi, S. Olariu, & M.C.Pinotti, “Efficient Training of Sensor Networks”, *2nd International Workshop on Algorithmic Aspects of Wireless Sensor Networks (ALGOSENSORS 2006)*, July 15, 2006, Venice, Italy, in LNCS 4240, December 2006, 1-12.
4. A. A. Bertossi & M.C. Pinotti, “Skewed Allocation of Non-Uniform Data for Broadcasting over Multiple Channels”, *IEEE Int’l Parallel and Distributed Processing Symposium (IPDPS)*, April 26-28, 2006, Rhodes, Greece, 8 pp.

5. N. Saxena, K. Basu, S. Das, and C.M. Pinotti, "A New Service Classification Strategy in Hybrid Scheduling to Support Differentiated QoS in Wireless Data Networks", *International Conference on Parallel Processing (ICPP-05)*, 2005, Oslo, Norway, June 2005, 389-396.
6. N. Saxena and M. C. Pinotti, "On-line Balanced K-Channel Data Allocation with Hybrid Schedule per Channel", *IEEE Intl. Conf. in Mobile Data Management (MDM)*, 2005, Ayia Napa, Cyprus, May 2005, 239-246.
7. N. Saxena, K. Basu, S.K. Das, & M.C. Pinotti, "A Dynamic Hybrid Scheduling Algorithm with Clients' Departure for Impatient Clients in Heterogeneous Environments", *5th IEEE International Workshop on Algorithms for Wireless, Mobile, Ad Hoc and Sensor Networks (WMAN), WMAN-IPDPS*, 2005, 7 pp.
8. N. Saxena, K. Basu, S.K. Das & C.M. Pinotti, "A New Hybrid Scheduling Framework for Asymmetric Wireless Environments with Request Repetition", *3rd IEEE Intl. Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, April 3-7, 2005, 368-376.
9. N. Saxena, C.M. Pinotti, & S.K. Das "A Probabilistic Push-Pull Hybrid Scheduling Algorithm for Asymmetric Wireless Environment", *IEEE GLOBECOM Wireless Ad hoc and Sensor Networks*, Dallas, TX, December 2004, 5-9.
10. E. Ardizzoni, A. Bertossi, M.C.Pinotti, & R.Rizzi, "Comparing Algorithms for Data Broadcasting over Multiple Channels", *AlgorithmS for Wirelss and Ad-hoc networks (A-SWAN)*, August 26, 2004, Boston, USA.
11. A. Boukerche, T. Dash & M.C. Pinotti, "Performance Analysis of a Hybrid Push-Pull Algorithm with QoS Adaptions in Wireless Networks", *The Ninth IEEE Symposium on Computers and Communications (ISCC04)*, June 28 - July 01, 2004, Cairo, Egypt, 512-524.
12. A.A. Bertossi, M.C. Pinotti, S. Ramaprasad, R. Rizzi, & M.V.S. Shashanka, "Optimal multi-channel data allocation with flat broadcast per channel", *IEEE Int'l Parallel and Distributed Processing Symposium (IPDPS)*, April 26-30, 2004, Santa Fe, USA, 8 pp.
13. A.A. Bertossi, M.C. Pinotti, R. Rizzi, & A.M. Shende "Channel Assignment in Honeycomb Networks", *3rd ICTCS*, October 13-15, Bertinoro, Italy, 2003.
14. A.A. Bertossi, M.C. Pinotti & R. Rizzi, *Channel Assignment with Separation on Trees and Interval Graphs*, *3rd Int'l Workshop on Wireless, Mobile and Ad Hoc Networks*, (satellite workshop of IEEE IPDPS 2003), April 26, 2003, 7 pp.
15. A.A. Bertossi, M.C. Pinotti, R. Rizzi, & P. Gupta, *Allocating Servers in Infostations for Bounded Simultaneous Requests*, *IEEE Int'l Parallel and Distributed Processing Symposium (IPDPS)*, April 22-26, 2003, Nice, France, 8 pp.
16. M.C. Pinotti, N. Saxena, "Push less and pull the current highest demanded data item to decrease the waiting time in asymmetric communication environments", *4th Int'l Workshop on Distributed Computing, Special Day on Wireless Networks*, in LNCS 2571, December 28-31, 2002, Calcutta, India.

17. A.A. Bertossi, M.C. Pinotti, & R. Tan, "Channel Assignment with Separation for Special Classes of Wireless Networks : Grids and Rings", *2nd Int'l Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing*, (satellite workshop of IEEE IPDPS 2002), April 15-19, 2002, Fort Lauderdale, Florida, 8 pp.
18. Y. Guo, S.K. Das & M.C. Pinotti, "A New Hybrid Broadcast scheduling Algorithm for Asymmetric Communication Systems: Push and Pull Data based on Optimal Cut-Off Point", *ACM Int'l Workshop on Modeling Analysis and Simulation of Wireless and Mobile Systems (MSWim 2001)*, Rome, July 2001, 123-130.
19. V. Auletta, S.K. Das, A. De Vivo, M.C. Pinotti & V. Scarano, "Optimal Tree Access by Elementary and Composite templates in Parallel Memory Systems", *Proc. Int'l Parallel and Distributed Processing Symposium* (sponsored by the IEEE Computer Society), San Francisco, April, 2001.
20. A.A. Bertossi, M.C. Pinotti & R. Tan, "Efficient Use of Radio Spectrum in Wireless Networks with Channel Separation between Close Stations", *DIAL M for Mobility; Int'l ACM Workshop on Discrete Algorithms and Methods for Mobile Computing*, Boston, August 11, 2000.
21. A.A. Bertossi & M.C. Pinotti, "Mappings for Conflict-Free Access of Paths in Elementary Data Structures", *Sixth Annual Int'l Computing and Combinatorics Conference*, Sydney, July 26-28, 2000 (atti pubblicati su LNCS).
22. S. Olariu, M.C. Pinotti & S.Q. Zheng, "An Optimal Hardware-Algorithm for Selection Using a Fixed-Size Parallel Classifier Device", *6th Int'l Conference on High Performance Computing*, Calcutta, India, December 17-20, 1999, 284-288.
23. S.Q. Zheng, K. Li, Y. Pan, & M.C. Pinotti, "Generalized Coincident Pulse Technique and New Addressing Schemes for Pipelined Time-Division Multiplexing Optical Buses", *6th (IEEE) International Conference on Parallel Interconnects (PI'99)*, Anchorage, Alaska, USA, October 17-19, 1999.
24. S.K. Das & M.C. Pinotti, "A Strictly-Optimal Strategy to Access Multi-Dimensional Data on Parallel Disk Systems", *29th Int'l Conference on Parallel Processing*, Aizu-Wakamatsu City, Japan, September 21-24, 1999, 120-127.
25. M.C. Pinotti & S.Q. Zheng, "Efficient Parallel Computation on a Processor Array with Pipelined TDM Optical Buses", *12th ISCA-PDCS Int'l Conference on Parallel and Distributed Computing Systems*, August 1999, Florida.
26. R. Lin, S. Olariu, K. Nakano, M.C. Pinotti, J.L. Schwing, & A. Y. Zomaya, "Scalable Hardware-Algorithms for Binary Prefix Sums", *Proc. Reconfigurable Architecture Workshop 99 Int'l Parallel Processing Symposium and Symposium on Parallel and Distributed Processing* (sponsored by the IEEE Computer Society), Puerto Rico, April 1999, 500-504.
27. S.K. Das & M.C. Pinotti, "An Optimal Disk Allocation Strategy for Partial Match Queries on Non-Uniform Cartesian Product Files", *Proc. Int'l Parallel Processing Symposium and Symposium on Parallel and Distributed Processing* (sponsored by the IEEE Computer Society), Puerto Rico, April 1999, 550-554.

28. S. Olariu, M.C. Pinotti & S.Q. Zheng, “An Optimal Hardware-Algorithm for Sorting Using a Fixed-Size Parallel Sorting Device”, *10th Int’l IASTED Conf. Parallel and Distributed Computing and Systems*, Las Vegas, Nevada, October 28-31, 1998, 38-44.
29. G. Brodal & M.C. Pinotti, “Comparator Networks for Binary Heap Construction”, *Sixth Scandinavian Workshop on Algorithm Theory*, July 1998, in LNCS 1432, Stockholm, Sweden, 158-168.
30. R. Lin, S. Olariu, K. Nakano, M.C. Pinotti, J.L. Schwing, & A. Y. Zomaya, “A Scalable VLSI Architecture for Binary Prefix Sums”, *Proc. Int’l Parallel Processing Symposium and Symposium on Parallel and Distributed Processing* (sponsored by the IEEE Computer Society), Orlando, April 1998, 333-337.
31. V. Auletta, S.K. Das, A. De Vivo, M.C. Pinotti & V. Scarano, “Toward a Universal Mapping for Accessing Trees in Parallel Memory Systems”, *Proc. Int’l Parallel Processing Symposium and Symposium on Parallel and Distributed Processing* (sponsored by the IEEE Computer Society), Orlando, April 1998, 447-454.
32. M.C. Pinotti & L. Wilson, “On the Problem of Tracking Mobile Users in Wireless Communications Networks”, *Wireless Networks and Mobile Computing Minitrack of the Thirty-First Hawaii International Conference on System Sciences (HICSS-31)*, January 6-9 1998, 666-671.
33. S.Q. Zheng, S. Olariu & M.C. Pinotti, “A Systolic Architecture for Sorting an Arbitrary Number of Elements”, *IEEE 3rd International Conf. on Algorithms and Architectures for Parallel Processing*, Melbourne, Australia, December 1997, 113-126.
34. S.K. Das, M.C. Pinotti & F. Sarkar, “Conflict-Free Data Access in Parallel Memory Systems: Algorithms and Experimental Study”, invited paper at *World Multiconference in Systemics, Cybernetics and Informatics (ISAS’97)* Caracas, Venezuela, July 7-11, 1997, 467-474.
35. G. Bilardi, B. Codenotti, G. Del Corso, M.C. Pinotti & G. Resta, “Broadcast and Other Primitive Operations on Fat-Trees”, *EuroPar*, Passau, Germany, August 26-29, 1997, in LNCS 1300, 196-207.
36. S.K. Das & M.C. Pinotti, “Load Balanced Mapping of Data Structures in Parallel Memory Modules for Fast and Conflict-Free Templates Access”, *Proc. 5th Int. Workshop on Algorithms and Data Structures (WADS’97)* Halifax NS, August 1997, in LNCS 1272, 272-281.
37. S.K. Das & M.C. Pinotti, “Conflict-Free Access to Templates of Trees and Hypercubes in Parallel Memory Systems”, *3rd Annual Int’l Conference on Computing and Combinatorics (Cocoon)*, Shanghai, Cina, August 20-22, 1997 in LNCS 1276, 1-10.
38. S.K. Das, M.C. Pinotti & F. Sarkar, “Conflict-Free Template Access in k -ary and Binomial Trees”, *Proc. ACM-Int’l Conference on Supercomputing 1997*, Wien, July 7-11, 1997, 237-244.

39. S.K. Das & M.C. Pinotti, “ $O(\log \log N)$ Time Algorithms for Hamiltonian-Suffix and Min-Max-Pair Heap Operations on the Hypercube”, *Proc. Int’l Parallel Processing Symposium* (sponsored by the IEEE Computer Society), Geneve, April 1-5, 1997, 507-511.
40. V. Crupi, S.K. Das & M.C. Pinotti, “Parallel and Distributed Meldable Priority Queues Based on Binomial Heaps”, *Proc. Int’l Conf. on Parallel Processing, Indian Lakes Resort*, August 12-16, 1996, 255-262.
41. S.K. Das, M.C. Pinotti & F. Sarkar, “Distributed Priority Queues on Hypercube Architectures”, *Proc. 16th IEEE Int’l Conf. on Distributed Computing Systems*, Hong Kong, May 27-30, 1996, 620-627.
42. V. Crupi, S.K. Das & M.C. Pinotti, “A Parallel Solution to the Extended Set-Union Problem With Unlimited Backtracking,” *Proc. IEEE Int’l Parallel Processing Symposium* (sponsored by the IEEE Computer Society), Hawaii, April 15-19, 1996, 182-186.
43. S.K. Das & M.C. Pinotti, ”Parallel CSD-Coding and Its Generalization”, *Proc. Int’l Conf. on High Performance Computing* (sponsored by IEEE Computer Society), New Delhi, India, December 1995, 730-733.
44. S.K. Das, M.C. Pinotti & F. Sarkar, “Conflict-Free Path Access of Trees in Parallel Memory Systems and Its Generalization with Applications to Distributed Heap Implementation”, *Proc. Int’l Conf. on Parallel Processing, Wisconsin (Oconomowoc)*, August 1995, Vol. III, 164-167.
45. G. Pucci & M.C. Pinotti “Parallel Algorithms for Priority Queue Operations”, *Scandinavian Workshop on Algorithm Theory SWAT 1992*, Helsinki, Finland, July 1992, in LNCS 621, 130-139.
46. G. Pucci & M.C. Pinotti “Parallel Priority Queues”, *Twenty-Eighth Annual Allerton Conference on Communication, Control and Computing*, Urbana Champaign (IL), October 1990, pp. 926-935.

T: Rapporti Scientifici

1. P. Barsocchi, A.A. Bertossi, M.C. Pinotti, & F. Potortì, “Allocating data for broadcasting over wireless channels subject to transmission errors”, **Technical Reports 2006/9**, Dipartimento di Matematica e Informatica, Università di Perugia, 2006.
2. F. Barsi & M.C. Pinotti, “Error Control by Product Codes in Arithmetic Units”, **Technical Reports 2007**, Dipartimento di Matematica e Informatica, Università di Perugia, 2007.
3. F. Barsi, A.A. Bertossi, F. Betti Sorbelli, R. Ciotti, S. Olariu & M.C. Pinotti, “Asynchronous Corona Training Protocols in Wireless Sensor and Actor Networks”, **Technical Reports 2007**, Dipartimento di Matematica e Informatica, Università di Perugia, 2007.

D: Altre Pubblicazioni

1. A.A. Bertossi, L. Mancini, & M.C. Pinotti, *Esercizi di Algoritmi e Strutture Dati (Seconda Edizione)*, Servizio Editoriale Universitario, Pisa, 1993, 111 pp.

O: Comunicazioni a Congressi (con revisione, ma senza atti)

1. M.C. Pinotti, N. Saxena, “Research in Broadcasting and Disseminating Information Goods On the Air”, *Sixth International Conference on Information Technology*, December 22-25, 2003, Bhubaneswar, India
2. A.A. Bertossi, M.C. Pinotti, S. Ramaprasad, R. Rizzi, & M.V.S. Shashanka, “Optimal Data Broadcast over Multiple Channels”, *Algorithmic Solutions for Wireless Networks*, Bologna, Ottobre 2003
3. A.A. Bertossi, M.C. Pinotti, & R. Tan, “Channel Assignment with Separation for Interference Avoidance in Wireless Networks”, *WSDAAL 01*, September 2001, Como.
4. A.A. Bertossi & M.C. Pinotti, “Efficient use of radio spectrum in wireless networks”, *ARDENT 01*, Gennaio 2001, Madonna di Campiglio.
5. S.K. Das & M.C. Pinotti, “Conflict-Free Parallel Access to Templates in Trees and Hypercubes”, Research Announcements, *9th Annual ACM Symposium on Parallel Algorithms and Architectures*, June 22-25, 1997, Newport, Rhode Island.