Curriculum Scientifico e Didattico
Maria Cristina Pinotti

`Qualifica:`	Professore Associato INF/01 (ex K05B)
	Idonea Professore Ordinario INF/01 (22 Maggio 2003): G.U. n. 79 del 4/10/2002
`Affiliazione:`	Dipartimento di Matematica e Informatica, Università di Perugia
`Sede lavoro:`	Via Vanvitelli, 1, 06123 Perugia
	Tel. (+39) 335 8448245
	E-mail: pinotti@unipg.it, pinotti@iei.pi.cnr.it
	Home page: http://www1.isti.cnr.it/~pinotti
`Residenza:`	Via delle Medaglie d'oro, 5, 56127 Pisa, Telefono: 050-550278
`Nascita:`	Mantova, 11/08/1962

Studi e Posizioni in Italia

`Laurea:`	Scienze dell'Informazione, Università di Pisa, 19/06/1986, *cum laude*
`Posizioni:`	Dal 30/12/1987 al 31/08/2000, Ricercatore C.N.R. Istituto di Elaborazione dell'Informazione (Pisa) (Idonea profilo di Primo Ricercatore C.N.R., Settore Disciplinare Informatica Fisica e Cibernetica, 4/1997)
	Dal 1/09/2000 al 31/10/2003, Professore Associato, Dipartimento di Informatica e Telecomunicazioni, Università di Trento
	Dal 1/11/2003, Professore Associato, Dipartimento di Informatica e Matematica, Università di Perugia

Periodi di Ricerca all'Estero

`08/1994 - 12/1994`	Visiting Researcher, Department of Computer Science, University of North Texas, Denton (TX), U.S.A.;
`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*
`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*,
`10/1998 - 11/1998`	Visiting Researcher, Department of Computer Science, Old Dominion University, Norfolk (VA), USA.

Attività Scientifica

Interessi di ricerca: Aspetti computazionali dei sistemi ad elevate prestazioni; progetto e analisi di strutture dati; reti wireless e cellulari; organizzazione di basi di dati; biologia computazionale.

Classificazione delle pubblicazioni:

1. **Aspetti computazionali dei sistemi ad elevate prestazioni**
   (a) sistemi paralleli VLSI
   - aritmetica per sistemi numerici residui [A4, A6, A7, A8, A9, A11, A13, C4]
   - ordinamento [A15, A20, C14, C19]
   - mediana [C25, A28]
   - reti logiche [A1, A2]
   - reti di interconnessione [C12]

(b) sistemi paralleli con bus riconfigurabili
   - somma dei prefissi [A18, C17, C21]

(c) sistemi paralleli con bus ottici
   - somma dei prefissi [C22]
   - tecniche di indirizzamento [A21, C24]

2. **progetto e analisi di strutture dati**

   (a) disegno di strutture dati per il modello computazionale PRAM
      - code di priorità [A3, A5, A10, A17, C1, C2, C7]
      - collezioni di insiemi [C5]

   (b) implementazioni di strutture dati in reti
      - reti a grado limitato [A12, A14, C6, C8]
      - reti di confrontatori [A16, C18]

   (c) implementazione di strutture dati in memorie a banchi
      - allocazioni di array, alberi, anelli, [L1, L2, A19, A23, A25, C3, C9, C10, C11, C13, C16, C26, C28]

3. **reti wireless**
   - localizzazione di utenti in reti cellulari [A24, C15]
   - assegnamento di canali in reti senza fili [A27, C27, C30, C33, C34]
   - tecniche di disseminazione dei dati [A22, C29, C31]
   - infostations [C32]

4. **organizzazione di basi di dati**
   - accesso a collezioni di records multidimensionali [A26, C20, C23]

5. **biologia computazionale**
   - single nucleotide polymorphism [R1]

Sommario delle Pubblicazioni

Aspetti computazionali dei sistemi ad elevate prestazioni

Sistemi paralleli VLSI. 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 [A8]. Essa calcola l'espressione polinomiale del CRT nel sistema a radici miste associato al sistema residuo evitando perciò computazioni a pieno campo. In [A6], 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 [A7] si prova che ha complessità in area e tempo, misurata dalla funzione AT^2, ottima.

Infine, in [A11], 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 [A4] è proposta e valutata in area e tempo un'architettura che rende flessibile un sistema ibrido.

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 [A9], 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.

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 [A13, C4], 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).

In [C14, A15], 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. La soluzione definitiva a questo problema, cioè una architettura VLSI per l'ordinamento di grandi agglomerati di dati che richiede ottimo $O(\frac{N\log N}{p\log p})$ tempo, è discussa in [C19, A20]. 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.

In [C25], si propone un algoritmo hardware per estrarre il k-esimo elemento da un insieme di N dati applicando ripetutamente un classificatore che dati p elementi, $p \ll N$, restituisce i $p/2$ elementi più piccoli seguiti dai $p/2$ elementi più grandi. L'algoritmo, che per la fase di sampling simula il funzionamento dell'algoritmo di sorting in [C19], è ottimo in tempo.

Infine, in [A28], si studia un algoritmo hardware per il calcolo della mediana di un insieme di N elementi che applica ripetutamente una rete di confrontatori capace di classificare p elementi, con $p < N$, in due gruppi ciascuno di $p/2$ elementi e tali che gli elementi del primo gruppo sono minori o uguali agli elementi del secondo gruppo. L'algoritmo proposto richiede $O(N/p + \log p)$ tempo ed è ottimo quando $N = p \log p$.

Sempre nell'ambito della progettazione VLSI, si sono affrontati i problemi di ridurre l'area occupata dall'interconnessioni fra chips [A2] e quello di minimizzare l'area occupata da un *programmable logic array* (PLA) [A1]. In [A2], 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 $f : p \to p$. 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 variabile 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 [A1] è proposta un'euristica che seleziona la riga corrente 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.

In [C12], 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.

Sistemi paralleli con bus riconfigurabili. In [C17], è considerata un'architettura VLSI fornita di bus molto corti, capaci anche di modificare un segnale (in accordo a particolari registri di stato) mentre lo propagano. Per tale architettura, si propone un' implementazione ottima, nel numero di broadcast richiesti, del calcolo dei prefissi della somma di una stringa di bits. Soluzioni più efficienti per lo stesso problema sono presentate in [C21, A18]. 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.

Sistemi paralleli con bus ottici. Il problema del calcolo dei prefissi della somma di una stringa di bits è affrontato anche 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 [C22].

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 [C24, A21] 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.

Progetto e Analisi di Strutture Dati

Strutture dati per il modello computazionale 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 [A3, C1], 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, Cuncurrent-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 [A10, C2] 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 [A5], 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 [C7, A17] 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 [C5], 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.

Implementazione di strutture dati in reti. Lo studio di realizzazioni di strutture dati in reti di processori a memoria distribuita, come è discusso in [C6], 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 [A12], 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 [A14, C8]. Per ottenere tale risultato, si è risolto in modo ottimo, senza permutazioni globali dei dati, il 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 [C5], 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 [A16, C18], 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 [A16, C18], 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 $c(n) \in (\Omega(n), o(n \log n))$.

Implementazione di strutture 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 256 processori e 2^{15} moduli. Anche un fattore di espansione pari ad 1, non esclude che in uno stesso ciclo di CPU occorrano *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 [L1, A19, A23, L2]. In particolare, è formalizzato in [C2] 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:

- sottoalberi di alberi binari completi [C10],
- sottoalberi di alberi k-ari completi [C9],
- sottoalberi di alberi binomiali [C11],
- cammini dalle foglie alla radice di alberi completi [C3],
- sottocubi di ipercubi e di ipercubi generalizzati [C10],
- cammini di lunghezza k (k è un parametro) di matrici, anelli e alberi completi [C26, A25],
- cammini e sottoalberi di alberi completi [C16],
- sottografi connessi di grafi arbitrari [C13].

Oltre a studiare caso per caso il lower bound al numero di banchi 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 [C28, A23], si studiano mapping *versatili* che minimizzano il numero di conflitti sia per cammini che per sottoalberi. Le tecniche usate in [C26] per derivare lower and upper bounds per il numero di banchi di memoria necessari per l'accesso senza conflitti sono di particolare interesse. In [A19], sono riportati la maggior parte dei risultati in [C10, C9, C11, C3].

Reti Wireless

Localizzazione di utenti in reti cellulari. L'obbiettivo 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 cossiddette 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 [C15, A24], 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.

Assegnamento di canali in reti senza fili. È studiato in [C27] il problema di allocare risorse per la comunicazioni (per esempio, canali di frequenze e time-slots) alle stazioni base delle reti senza fili (WLAN, Wireless Local Loop, reti cellulari) in modo tale che stazioni vicine usino frequenze molto diverse per evitare interferenze nelle trasmissioni e le stesse frequenze siano riutilizzabili solo a distanza superiore di una soglia prestabilita. In particolare, sono considerate due versioni molto realistiche del problema di assegnamento, equivalenti a varianti di problemi di colorazione di grafi, dove le frequenze sono rappresentate con numeri interi. La differenza fra le frequenze assegnate a due stazioni adiacenti deve essere di almeno 2, mentre la stessa frequenza può essere riutilizzata in stazioni a distanza almeno 3 o 4. Entrambe queste versioni del problema, in cui si vuole 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 piú logaritmico la frequenza assegnabile a ciascuna stazione.

Inoltre, in [C30, A27], osservato che il problema dell'interferenze sarà sempre più esteso dato l'uso di una rete sempre più capillare, 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 $[d(1), d(2), ..., d(s-1)]$, il problema 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 $d(i)$ ogniqualvolta 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 $[1, 1, \ldots, 1]$ e $[2, 1, \ldots, 1]$. Per la rete ad albero, a grafo di intervalli e la rete a griglia esagonale (Honeycomb grid), rispettivamente in [C33] e in [C34], sono progettati algoritmi ottimi o approssimati di complessità polinomiale per vettori di separazione $[1, 1, \ldots, 1]$ e $[\delta_1, \delta_2]$.

Tecniche di disseminazione dei dati. L' architettura di rete wireless che si va delineando consiste in uno o più livelli di insfrastrutture. Alla rete fisica, si sovrappone una infrastruttura per l'accesso, una infrastruttura per la distribuzione dei dati, ed infine è auspicabile avere una infrastruttura dei servizi personalizzati, che dipendono cioè sia dal contesto che dall'utente. Ogni infrastruttura è concepita in modo distribuito e pertanto sarà ben lontana dall'essere monolitica. Essa piuttosto sarà il frutto dell'integrazione di soluzioni assai diverse fra loro, dettate da criteri di opportunità, quali le opportunità tecnologiche, geografiche, e finanziarie. Nell'ambito della infrastruttura di distribuzione dei dati, si studia un algoritmo per l'ottimizzazione dell'organizzazione dei dati quando la comunicazione avviene in un ambiente di comunicazione asimmetrica, cioè quando il server che distribuisce i dati ha molto più banda di trasmissione dell'utente che desidera ricevere qualche informazione. In questo contesto, si è progettato un algoritmo di scheduling ibrido che diffonde (broadcast) i dati più richiesti e distribuisce su richiesta i dati meno popolari. Il punto di rottura tra i dati piú popolari e quelli meno popolari è scelto in modo che sia minimizzato il tempo che l'utente deve rimanere in ascolto prima di ricevere il dato a cui è interessato. L'algoritmo determina analiticamente il punto di rottura tra i dati più popolari e quelli meno popolari. Una simulazione sperimentale del sistema rafforza i risultati ottenuti dall'analisi teorica del problema [C29, A22]. In [C31], si studia una variante del problema in [A22], dove il sistema gestisce una coda di richieste per ciascun dato.

In [R2] si studia il problema del broadcast su canali multipli sotto l'ipotesi che i dati assegnati ad un canale siano distributi in a round-robin fashion (scheduling flat). Il problema si riduce a trovare un assegnamento dei dati ai canali tale che minimizzi il tempo medio di attesa degli utenti misurato da una funzione quadratica che dipende sia dalla cardinalita' che dalla somma delle probabilita' dei dati assegnati ad un canale.

Infostations. In [C32], è studiato il problema di minimizzare il numero di processori necessari in una *infostation* per servire on-line e senza ritardo le richieste degli utenti che l'attraversano. Ogni richiesta è caratterizzata da una durata e da un tipo. Ogni processore può soddisfare simultaneamente al più k richieste di cui al più h_c dello stesso tipo c. Si mostra che il problema è NP-hard, e si danno algoritmi approssimati di complessità polinomiale.

Organizzazione di Basi di Dati

Accesso a collezioni di records multidimensionali. In [A26, C20, C23], 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 basato sui codici residui e risolve ottimamente ogni partial match query.

Biologia Computazionale

Single nucleotide polymorphism. Si è intrappreso lo studio di algoritmi per la biologia computazionale. In particolare, in [R1], nell'ambito dei problemi relativi alle variazioni dei singoli nucleotidi (Single Nucleotide Polymorphism SNP) si è studiata la complessità computazionale del problema di trovare un insieme di haplotype di cardinalità minima capace di spiegare una famiglia data di genotype.

Attività Didattica

Insegnamenti:

- In qualità di Ricercatore C.N.R.

A.A. 1990 – 91 A.A. 1991 – 92	Professore a contratto *ex. art. 25 (382/80)* 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. 1992 – 93	Professore a contratto *ex. art. 25 (382/80)* di "Complementi di Strutture Dati ed Algoritmi", presso il Corso di Laurea in Scienze dell'Informazione, Università di Pisa
A.A. 1993 – 94 A.A. 1994 – 95	Professore a contratto *ex. art. 25 (382/80)* di "Strutture Dati ed Algoritmi di Base", presso il Corso di Laurea in Scienze dell'Informazione, Università di Pisa
A.A. 1995 – 96 A.A. 1996 – 97	Professore a contratto *ex art. 100 (382/80)* di "Algoritmi e Strutture Dati", presso il corso di Diploma Universitario in Informatica, Università di Perugia

- In qualità di docente dell'Università di Trento

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

Tesi:

- Ha seguito diverse tesi, in qualità di ricercatore C.N.R. In particolare dalla tesi "Costruzione di Heaps su Connection Machine" di Carlo Luchetti e "Algoritmi PRAM per la soluzione dell'Union-Find" di Vincenzo Crupi sono stati ricavati gli articoli A5, C5 e C7.
- Sta seguendo tesisti presso l'Università di Trento, ed in particolare la tesi di dottorato di Navrati Saxena "Scheduling data on the air",da cui al momento è stato tratto l'articolo C31. Inoltre, ha seguito due stagisti nell'ambito della collaborazione BITS Pilani, India – DIT. Sono in preparazione due rapporti scientifici su "Broadcast on the air" e "Reporting Center Problems on Honeycomb Grids".

Altre Attività

Partecipazione a Progetti:

- 1988 – 1990: Progetto Finalizzato "Materiali e dispositivi per l' elettronica a stato solido", Responsabile Unità Operativa: E. Martinelli;
- 1989 – 1993: Progetto Finalizzato "Sistemi informatici e calcolo parallelo", sottoprogetto "Processori dedicati", Responsabile Unità Operativa: L. Lopriore;
- 1993 – 96: 60% Università di Perugia "Algoritmi per elaborazione numerica ed applicazioni calcolo simbolico", Responsabile Unità Operativa: F. Barsi;
- 1995 – 96 e 1996 – 97: 40% Progetto Nazionale "Efficienza di algoritmi e di strutture informative", Responsabile Nazionale: Giorgio Ausiello, Responsabile unità operativa di Perugia: F. Barsi;
- 1998: Progetto CNR "Sistemi Mobili", Responsabile unità operativa IEI: M. C. Pinotti, Responsabile Nazionale: M. Bonuccelli;
- 2001 – 2003: Progetto CoFin RE-AL-WI-NE, Responsabile Nazionale: Alan A. Bertossi,
- 2002 – 2004: Progetto UNITN-PAT "WILMA", Responsabile Uni-TN: R. Battiti;
- 2002 – 2003: Progetto FIRB "ADONIS", Responsabile Uni-TN: R. Battiti;

Partecipazione a Comitato di Programma di Conferenze/Workshop:

- *Ninth IASTED Int'l Conference on Parallel and Distributed Computing and Systems*, Washington, D.C., 1997
- *Workshop in Mobile Computing*, ISPAN'99, Perth, Western Australian, 1999.
- *Eleventh IASTED Int'l Conference on Parallel and Distributed Computing and Systems*, Cambridge, MA, 1999.
- *Sistx Int'l Conference on High-Performance Computing*, Calcutta, 1999.
- *Fifth Int'l Conference on Computer Science and Informatics*, Atlantic City, 2000.
- *Int'l Conference on Parallel Processing*, 2001
- *First Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing*, svoltosi in cooperazione con "International Parallel and Distributed Processing Symposium" (IPDPS2001)"
- *2nd Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing*, svoltosi in cooperazione con "International Parallel and Distributed Processing Symposium" (IPDPS2002)
- *Int'l Conference Parallel And Distributed Systems*, Taiwan, 2002.
- *Int'l Workshop on Distributed Computing*, Calcutta, December 28 – 31, 2002.
- *Int'l 2004 Workshop on High Performance Switching and Routing (HPSR 2004)*, Phoenix, Arizona, USA, April 18 – 21, 2004.

Organizzazione Workshop:

- *3rd Workshop on Wireless, Mobile and Ad Hoc Networks* (WMAN), satellite di "International Parallel and Distributed Processing Symposium" (IPDPS2003), Nice, April 22-26, 2003.
- *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.

Editor Special Issue:

E' uno degli editor dello Special Issue *Algorithmic Solutions for Wireless, Mobile, Ad Hoc and Sensor Networks* in corso di preparazione per la rivista **MObile NETworks and applications (MONET)**
`http://www.kluweronline.com/issn/1383-469X`

Revisore di articoli per riviste:

Revisore per numerose riviste, tra cui: IEEE Transactions on Parallel and Distributed Systems, IEEE Transactions on Circuits and Systems, Journal of Parallel and Distributed Computing, The VLSI Journal, IEE Proceeding, Electronic Letters, Information Processing Letters, Parallel Processing Letters.

Collaborazioni:

- Ha organizzato lo stage *Efficient Implementation of Dynamic Data Structures on Parallel Computers*, presso l'IEI-CNR (Pisa), nell'ambito del programma CNR, Short Term Mobility. Nell'ambito dello stesso programma, ha organizzato le visite dei Prof. S. Olariu, Prof. S.Q. Zheng, e Dr. G. Brodal nel 1995-96.
- Ha organizzato la visita del Prof. Stephan Olariu nel Dicembre 2000 e del Prof. Phalguni Gupta, IIT Kanpur, India nei mesi di Giugno e Luglio 2001 presso l'Università di Trento.
- Ha organizzato la visita del Prof. Anil Shende, BITS Pilani, India nel mese di Luglio 2002 presso l'Università di Trento.

Invited Talk:

- "$O(\log \log N)$ Time Algorithms for Hamiltonian-Suffix and Min-Max-Pair Heap Operations on the Hypercube" Dagstuhl Seminar on Data Structures, March 15, 1998;
- "Efficient Schemes for Distributing Data on Parallel Memory Systems" *Proceedings of the DIMACS Workshop on External Memory Algorithms and/or Visualization*, May 22, 1998;
- "Efficient Schemes for Distributing Data on Parallel Memory Systems", Old Dominion University, Norfolk (VA), U.S.A., October 14, 1998;
- "An Optimal Disk Allocation Strategy for Partial Match Queries on Non-Uniform Cartesian Product Files" DIMACS Workshop on Parallel Data Structures, May 10-11, 1999.
- "Channel Assignment with Separation for Interference Avoidance in Wireless Networks", University of Texas at Arlington, Arlington, TX, USA, April 11, 2001.

Elenco Pubblicazioni

A: Riviste Internazionali:

1. F. Luccio & M.C. Pinotti, “Suboptimal Solution for PLA Multiple Column Folding”, *Computer Aided-Design*, Vol. 22, No. 8, 1990, pp. 515-520.

2. F. Luccio & M.C. Pinotti, “Minimal Synthesis of Multivalued Functions with New Operators”, *IEE Proceedings Part E: Computer and Digital Tecniques*, Vol. 138, No. 6, 1991, pp. 419-423.

3. M.C. Pinotti & G. Pucci, “Parallel Priority Queues”, *Information Processing Letters*, Vol. 40, 1991, pp. 33-40.

4. F. Barsi & M.C. Pinotti, “Adding Flexibility to Hybrid Number Systems”, *The Computer Journal*, Vol. 35, No. 6, 1992, pp. 630-635.

5. C. Luchetti & M.C. Pinotti, “Some Comments on Building Heaps in Parallel”, *Information Processing Letters*, Vol. 47, 1993, pp. 145-148.

6. F. Barsi & M.C. Pinotti, “A Fully Parallel Residue-to-Binary Conversion”, *Information Processing Letters*, Vol. 50, No. 1, 1994, pp. 1-8.

7. F. Barsi & M.C. Pinotti, Addendum to “A Fully Parallel Residue-to-Binary Conversion”, *Information Processing Letters*, Vol. 55, No. 1, 1995, pp. 25-26.

8. F. Barsi & M.C. Pinotti, “Time Optimal Mixed Radix Conversion for Residue Number Applications”, *The Computer Journal*, Vol. 37, No. 11, 1994, pp. 907-916.

9. F. Barsi & M.C. Pinotti, “Efficient Error Correcting Technique for Digital Equipment”, *Electronic Letters*, Vol. 31, No. 3, 1995, pp. 158-159.

10. M.C. Pinotti & G. Pucci, “Parallel Algorithms for Priority Queue Operations”, *Theoretical Computer Science*, Vol. 148, Agust 1995, pp. 171-180.

11. 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, pp. 2427-2430.

12. 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, pp. 555-564.

13. S.K. Das & M.C. Pinotti, “Fast VLSI Circuits for CSD-Coding and GNAF-Coding”, *Electronics Letters*, Vol. 32, No. 7, 1996, pp. 632-634.

14. 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, pp. 200-211.

15. 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, pp. 487-499.

16. G. Brodal & M.C. Pinotti, “Comparator Networks for Binary Heap Construction”, *Theoretical Computer Science*, Vol. 250/1-2, November 2000, pp. 235-245.

17. S.K. Das & M.C. Pinotti, “Parallel Priority Queues Based on Binomial Heaps”, *Parallel Computing*, Vol. 26, pp. 1411-1428, 2000.

18. 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, pp. 838-850.

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, pp. 998–1027.

20. 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, pp. 1310–1324.

21. 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, pp. 1033–1051.

22. 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.

23. 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, pp. 399-412.

24. 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, pp. 215-227.

25. 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, pp. 1314-1333.

26. 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, pp. 1320-1332.

27. 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, pp. 222-235.

28. A.A. Bertossi, S. Olariu, M.C. Pinotti & S.Q. Zheng, *Selection on Matrices Classifying Rows and Columns*, accepted for publication on *IEEE Transactions on Parallel and Distributed Systems.*

C. Atti di Congressi Internazionali:

1. 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.

2. 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, pp. 130-139.

3. 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, pp. 164-167.

4. 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, pp. 730-733.

5. 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, pp. 182-186.

6. 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, pp. 620-627.

7. 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, pp. 255-262.

8. 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, pp. 507-511.

9. 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, pp. 237-244.

10. 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, pp. 1-10.

11. 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, pp 272-281.

12. 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, pp. 196-207.

13. 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, pp. 467-474.

14. 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, pp. 113-126.

15. 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, pp. 666-671.

16. 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, pp. 447-454.

17. 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, pp. 333-337.

18. G. Brodal & M.C. Pinotti, “Comparator Networks for Binary Heap Construction”, *Sixth Scandinavian Workshop on Algorithm Theory* , July 1998, in LNCS 1432, Stockholm, Sweden, pp. 158-168.

19. 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, pp. 38-44.

20. 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, pp. 550-554.

21. 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), pp. 500-504, Puerto Rico, April 1999.

22. 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.

23. 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, pp. 120-127.

24. 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.

25. 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, pp. 284 –288.

26. 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).

27. 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.

28. 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.

29. 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, pp. 123–130.

30. 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.

31. 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.

32. 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

33. 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

34. A.A. Bertossi, M.C. Pinotti, R. Rizzi, & A.M. Shende *Channel Assignment in Honeycomb Networks*, *3rd ICTCS*, October 13-15, 2003

L: Capitoli di Libri:

1. 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, pp. 233–245, 1999.

2. 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., pp. 21-42, 1999.

R: Rapporti scientifici Recenti:

1. G. Lancia, M.C. Pinotti & R. Rizzi, *Haplotyping Populations: Complexity and Approximations*, Dipartimento di Informatica e Telecomunicazioni, Università di Trento, DIT-02-0079, 2002.

2. A.A. Bertossi, M.C. Pinotti, S. Ramaprasad, R. Rizzi, & M.V.S. Shashanka, *Optimal multi-channel data allocation with flat broadcast per channel*, Dipartimento di Informatica e Telecomunicazioni, Università di Trento.

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

1. 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 Achitectures*, Newport, Rhode Island, June 22-25, 1997.

2. A.A. Bertossi, M.C. Pinotti, & R. Tan, "Channel Assignment with Separation for Interference Avoidance in Wireless Networks", *WSDAAL 01*, Como, September 2001.

D: Altre Pubblicazioni:

1. "Esercizi di Algoritmi e Strutture Dati (Seconda Edizione)", (con A. A. Bertossi e L. Mancini), *Servizio Editoriale Universitario*, Pisa, 1993.