

# INTEGRAZIONE GA-FEM BASATA SU UNA TECNICA DI CALCOLO DISTRIBUITO PER L'OTTIMIZZAZIONE DI COMPONENTI PASSIVI A MICROONDE

S. Caorsi<sup>1</sup>, M. Donelli<sup>2</sup>, A. Massa<sup>3</sup>, M. Raffetto<sup>2</sup>

<sup>1</sup> Dipartimento di Eletttronica, Università di Pavia, Via Ferrata 1, I-27100 Pavia  
(caorsi@ele.unipv.it)

<sup>2</sup> Dipartimento di Ingegneria Biofisica ed Eletttronica, Università di Genova, Via Opera Pia 11a, I-16145, Genova (donelli@dibe.unige.it, raffetto@dibe.unige.it)

<sup>3</sup> Dipartimento di Informatica e Telecomunicazioni, Università di Trento, Via Sommarive 14, I-38050, Trento, (andrea.massa@ing.unitn.it)

## Abstract

*Powerful automatic electromagnetic CAD tools can be devised by integrating in a single software package a genetic algorithm and a finite element simulator. A distributed computing technique exploiting PVM is exploited to limit the computational time required by the design of passive microwave components even when low-end personal computers are exploited. In order to provide an example of practical exploitation of this distributed computing CAD tool, the parallel efficiency and the speed up obtained in the design of a particular three-port rectangular waveguide junction are considered.*

## INTRODUZIONE

L'integrazione in un singolo strumento CAD di un programma di ottimizzazione ed un programma di analisi di un dispositivo rappresenta una delle tematiche di maggiore interesse applicativo nel campo della cosiddetta sintesi di componenti passivi a microonde. Sono infatti del tutto evidenti i vantaggi offerti da una progettazione non supervisionata, sia in termini di efficienza che di riduzione di costi.

Tuttavia, a fronte di tali innegabili vantaggi, risultano altrettanto significativi i possibili ostacoli ad una sua realizzazione funzionale. A tale proposito ed a titolo di esempio, si consideri il fatto che qualsiasi algoritmo di ottimizzazione richiede la valutazione di innumerevoli configurazioni parametriche del dispositivo in esame. Ed inoltre che ciascuna valutazione richiede l'utilizzo di un simulatore numerico generalmente oneroso da un punto di vista computazionale. Ne consegue che, complessivamente, il carico computazionale, richiesto da una progettazione automatica di un dispositivo a microonde, risulta essere eccessivo (per non dire impraticabile) per una "workstation" dalle prestazioni "standard". Tuttavia, recentemente, si è assistito allo sviluppo di numerosi pacchetti software (P4, Express, MPI, Linda, PVM, etc. [1]) che consentono lo sfruttamento di una rete eterogenea di calcolatori al fine di emulare una singola risorsa concorrente dalle prestazioni elevatissime (notevolmente superiori a quelle offerte da una risorsa di calcolo normalmente disponibile). D'altra parte, lo sviluppo di tecniche di ottimizzazione globale (come ad esempio i cosiddetti algoritmi genetici (GA), recentemente utilizzati per la soluzione di problemi di natura elettromagnetica [2]) intrinsecamente ad elevato parallelismo, rende possibile l'implementazione di strumenti CAD automatici pur disponendo di una semplice rete di "personal computers".

In questo lavoro viene presentata la descrizione di uno strumento CAD per la progettazione non-supervisionata di circuiti passivi a microonde e viene fornito un esempio di applicazione pratica.

## DESCRIZIONE DELLO STRUMENTO CAD

Per descrivere l'architettura dello strumento CAD proposto, si consideri il problema di progettare un generico dispositivo passivo a microonde. Senza perdere di generalità, si può assumere che l'obiettivo della sintesi sia quello di determinare i valori ottimi di un insieme di parametri e che il criterio di ottimalità sia definito per mezzo di una opportuna funzione di costo da minimizzare. In genere questa funzione di costo risulta essere non lineare. Inoltre spesso presenta un numero elevato di minimi locali che impediscono il corretto funzionamento delle procedure classiche (deterministiche) di ottimizzazione. Per superare questi problemi viene quindi utilizzato un algoritmo genetico [2].

Gli algoritmi genetici sono delle procedure “multi-agent” e presentano un parallelismo intrinseco. Per ogni generazione vengono considerati contemporaneamente  $M$  diversi punti dello spazio delle soluzioni e questo consente la valutazione di  $M$  diversi valori della funzione di costo in parallelo. Inoltre, così facendo, ogni singola valutazione della funzione di costo può ancora essere considerata come una singola operazione indivisibile. Quest'ultima considerazione implica che non è necessario modificare il simulatore numerico e che l'implementazione del parallelismo è semplice se i due blocchi costituenti (l'algoritmo genetico e il simulatore a elementi finiti) sono già stati sviluppati.

Questo semplice approccio alla programmazione parallela è chiamato “master-slave model” [1]. Esso è il più comune dei modelli supportati da PVM [1]. L'algoritmo genetico è il “master”, che controlla la generazione dei processi “slave” che eseguono le simulazioni con il metodo degli elementi finiti.

Il codice dell'algoritmo genetico viene modificato con istruzioni PVM solo all'interno del sottoprogramma che “calcola il valore della funzione di costo per tutti i cromosomi [2] di una singola popolazione”. In particolare, in questo sottoprogramma, si troveranno le chiamate alle “routines” PVM che consentono di generare dei processi “slave”, trasmettere a questi ultimi i parametri corrispondenti a un dato cromosoma e ricevere il valore della funzione di costo da ogni singolo processo generato.

Per quanto riguarda i singoli processi “slave”, è sufficiente sviluppare un semplice programma che deve essere in grado di ricevere i valori dei parametri di progetto corrispondenti ad un dato cromosoma, lanciare una simulazione che calcola il valore della funzione di costo per la configurazione di progetto appena ricevuta e trasmettere al master il valore della funzione di costo calcolato.

## ESEMPIO NUMERICO

Come primo test si consideri il progetto di una giunzione a 3 porte in guida d'onda rettangolare WR-28 (7.112 mm  $\times$  3.556 mm). Si deve dimensionare il diaframma presente nella zona centrale della giunzione. In particolare è necessario dimensionare la posizione  $p$ , la larghezza  $l$  e l'altezza  $a$ . In Figura 1 è riportata una vista dall'alto di una possibile soluzione del problema. La frequenza di funzionamento è di 38.598 GHz. Si richiede che i parametri  $S$  della giunzione siano tali che  $0.578 \leq |S_{13}| \leq 0.638$ ,  $0.620 \leq |S_{23}| \leq 0.690$  e  $0.40 \leq |S_{11}| \leq 0.498$ , dove, facendo nuovamente riferimento alla Figura 1, la porta 1 è quella centrale, la porta 2 è quella di destra e la porta 3 è quella di sinistra.

La funzione di costo considerata è data da:

$$f(p, l, a) = \sqrt{(|S_{11}|(p, l, a) - 0.449)^2 + (|S_{13}|(p, l, a) - 0.608)^2 + (|S_{23}|(p, l, a) - 0.655)^2}.$$

In questo primo test, per valutare le prestazioni dell'algoritmo, soprattutto in termini di “speed-up” e “parallel efficiency”, si è supposto che i parametri del diaframma potessero assumere dei valori pari ad un numero intero di elementi “rettangolari” mostrati in Figura 1.

Di conseguenza, è stato possibile scegliere una codifica intera per l'algoritmo genetico. Inoltre si è stabilito di usare popolazioni composte da 10 individui. La rete di calcolatori utilizzata è composta da 24 calcolatori identici.

In Figura 2 viene mostrato l'andamento del valore della funzione di costo corrispondente al miglior individuo di ciascuna delle popolazioni generate dall'algoritmo genetico durante la sua evoluzione. Il valore ottenuto alla quinta e ultima iterazione è pari a  $f = 0.039706$ . La configurazione ottima corrisponde ai seguenti valori:  $p = -0.7112$  mm ( $p$  individua la coordinata  $x$  degli spigoli di sinistra del diaframma (si veda la Figura 1)),  $l = 3.556$  mm e  $a = 2.1336$  mm.

Infine in Figura 3 vengono riportati gli andamenti dello "speed-up", della "parallel efficiency" e del tempo di calcolo normalizzato al tempo di calcolo ottenuto nel caso seriale (utilizzando solo uno dei 24 calcolatori disponibili) in funzione del numero di calcolatori utilizzati. Si noti che utilizzando un algoritmo genetico con popolazioni composte da 10 individui non si ottiene alcun miglioramento in termini di prestazioni quando il numero di calcolatori utilizzati è superiore a 10. Al contrario, per un numero di calcolatori minore o uguale a 10 l'incremento delle prestazioni risulta essere significativo.

## Riferimenti bibliografici

- [1] A. Geist, A. Beguelin, J. Dongarra, W. Jiang, R. Manchek, and V. Sunderam, *PVM: Parallel Virtual Machine. A users' guide and tutorial for networked parallel computing*. Cambridge, Massachusetts: The MIT Press, 1994.
- [2] Y. Rahmat-Samii and E. Michielssen, *Electromagnetic Optimization by Genetic Algorithms*. New York: Wiley and Sons, 1999.

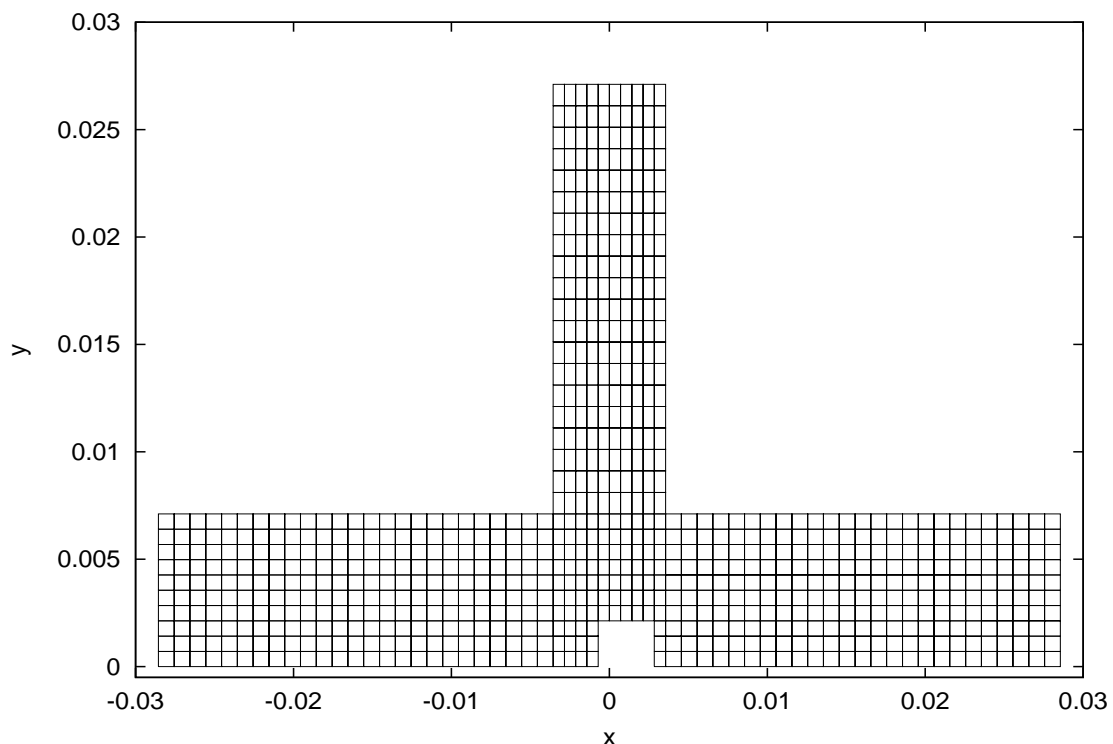


Figura 1: Configurazione ottima del dispositivo di interesse.

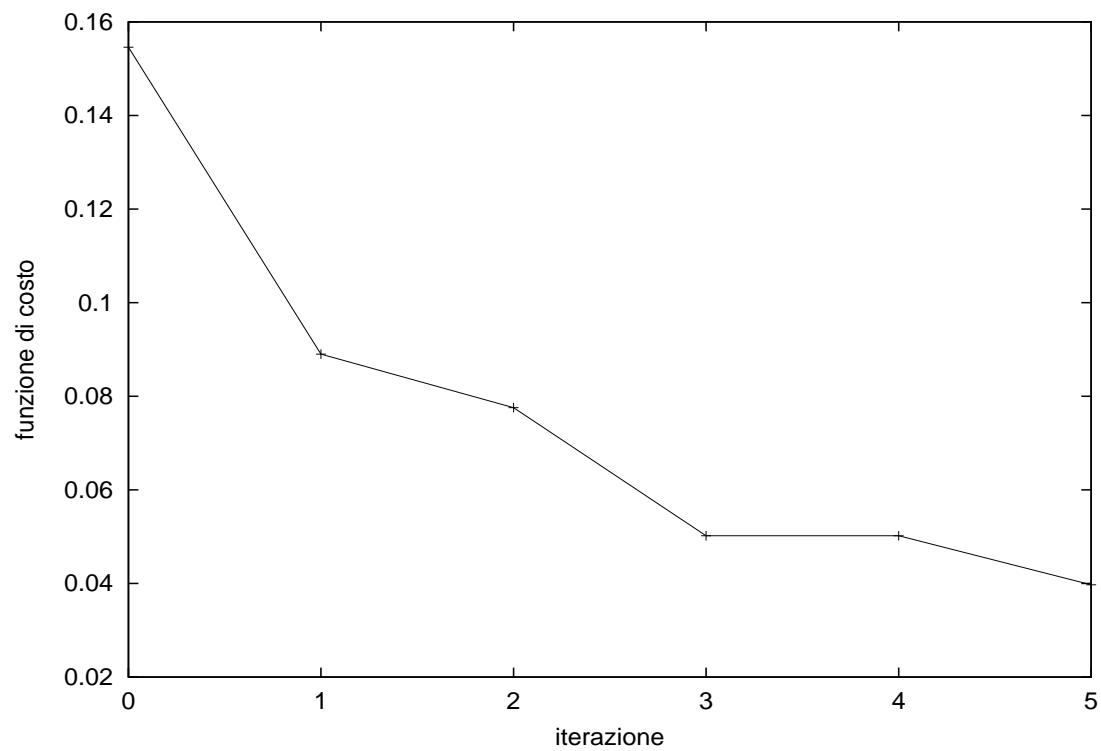


Figura 2: Valori della funzione di costo corrispondenti ai migliori individui di ciascuna popolazione.

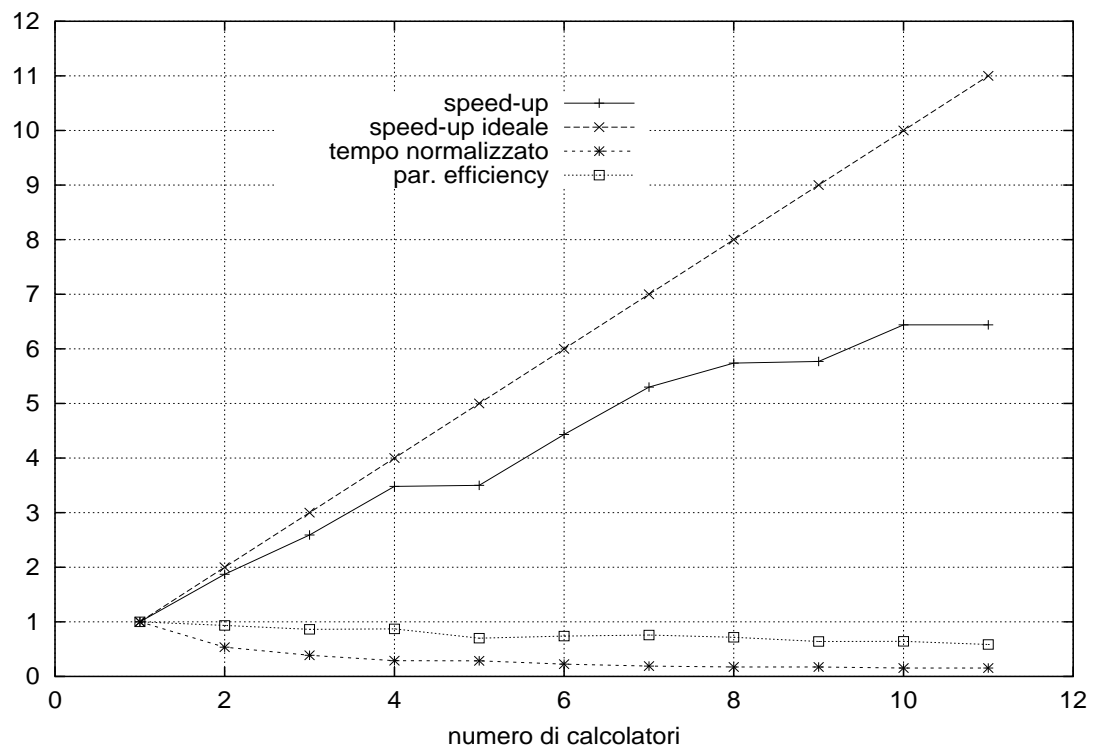


Figura 3: Andamento dello “speed-up” e della “parallel efficiency” in funzione del numero di calcolatori utilizzati.