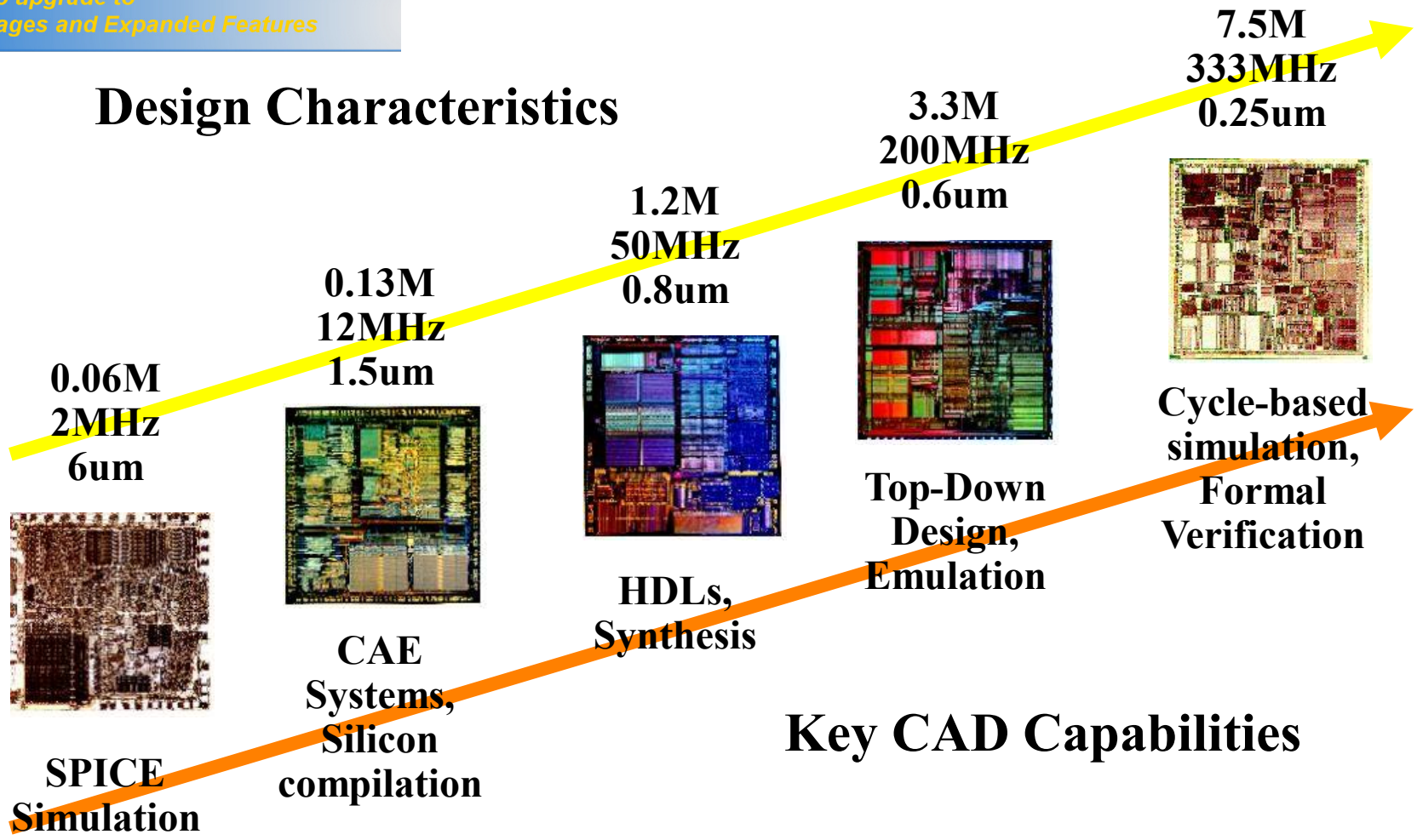
General Iterative Heuristics for VLSI Multiobjective Partitioning

by
Dr. Sadiq M. Sait
Dr. Aiman El-Maleh
Mr. Raslan Al Abaji
King Fahd University
Computer Engineering Department

Outline é .

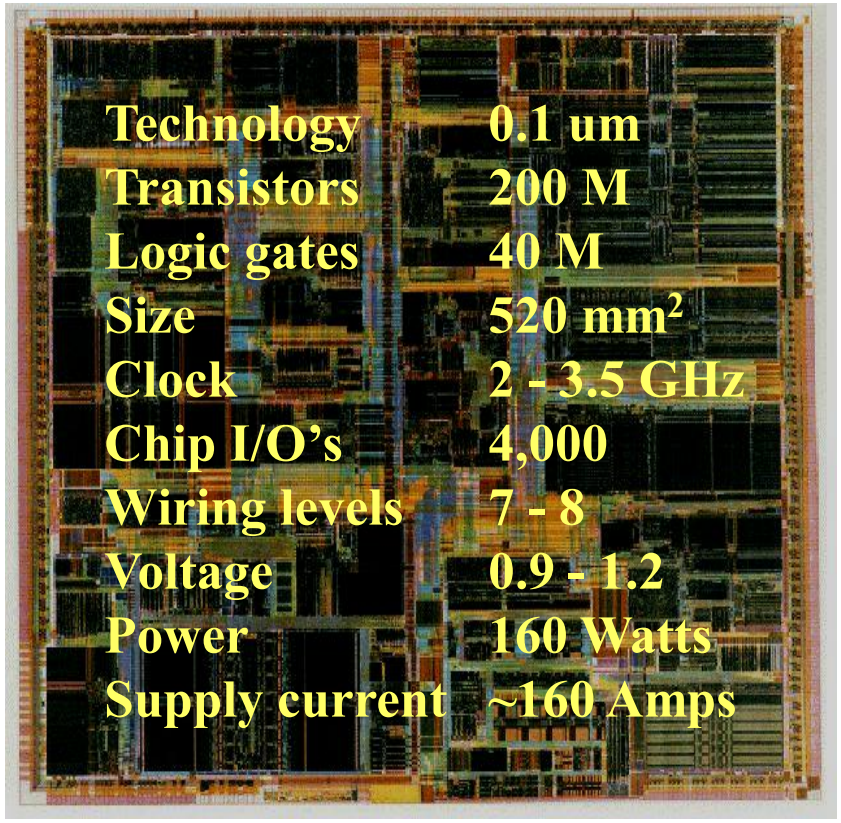
- Introduction
- Problem Formulation
- Cost Functions
- Proposed Approaches
- Experimental results
- Conclusion

Technology Trend



The Challenges to sustain such an exponential growth to achieve gigascale integration have shifted in a large degree, from the process of manufacturing technologies to the design technology.

The VLSI Chip in 2006



Performance
Power consumption
Noise immunity
Area
Cost
Time-to-market

Tradeoffs!!!

VLSI Design Cycle

ÅVLSI design process is carried out at a number of levels.

1. System Specification
2. Functional Design
3. Logic Design
4. Circuit Design
5. **Physical Design**
6. Design Verification
7. Fabrication
8. Packaging Testing and Debugging

Physical Design

The physical design cycle consists of:

1. **Partitioning**
2. Floorplanning and Placement
3. Routing
4. Compaction

Physical Design converts a circuit description into a geometric description. This description is used to manufacture a chip.

ve need Partitioning ?

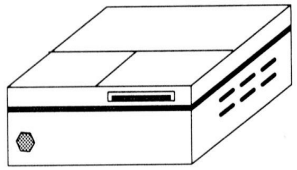
- Decomposition of a complex system into smaller subsystems.
- Each subsystem can be designed independently speeding up the design process (divide-and conquer-approach).
- Decompose a complex IC into a number of functional blocks, each of them designed by one or a team of engineers.
- Decomposition scheme has to minimize the interconnections between subsystems.

els of Partitioning

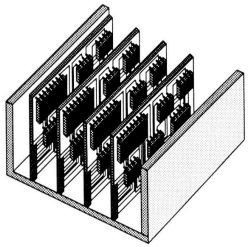
System Level Partitioning

Board Level Partitioning

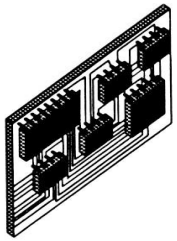
Chip Level Partitioning



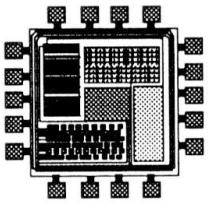
System



PCBs



Chips



Subcircuits / Blocks

Motivation

Need for Power optimization

- Portable devices.
- Power consumption is a hindrance in further integration.
- Increasing clock frequency.

Need for delay optimization

- In current sub micron design wire delay tend to dominate gate delay. Larger die size imply long on-chip global routes, which affect performance.
- Optimizing delay due to off-chip capacitance.

Objective

- Design a class of iterative algorithms for VLSI multi objective partitioning.
- Explore partitioning from a wider angle and consider circuit delay , power dissipation and interconnect in the same time, under balance constraint.

Problem formulation

Objectives :

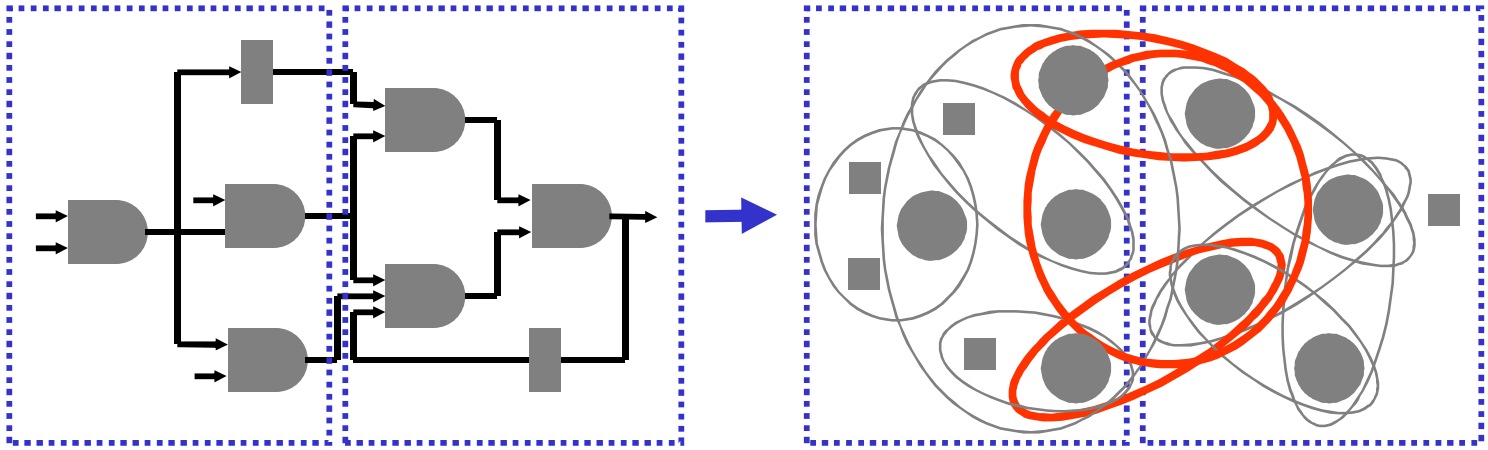
- Power cost is optimized AND
- Delay cost is optimized AND
- Cutset cost is optimized

Constraint

- Balanced partitions to a certain tolerance degree. (10%)

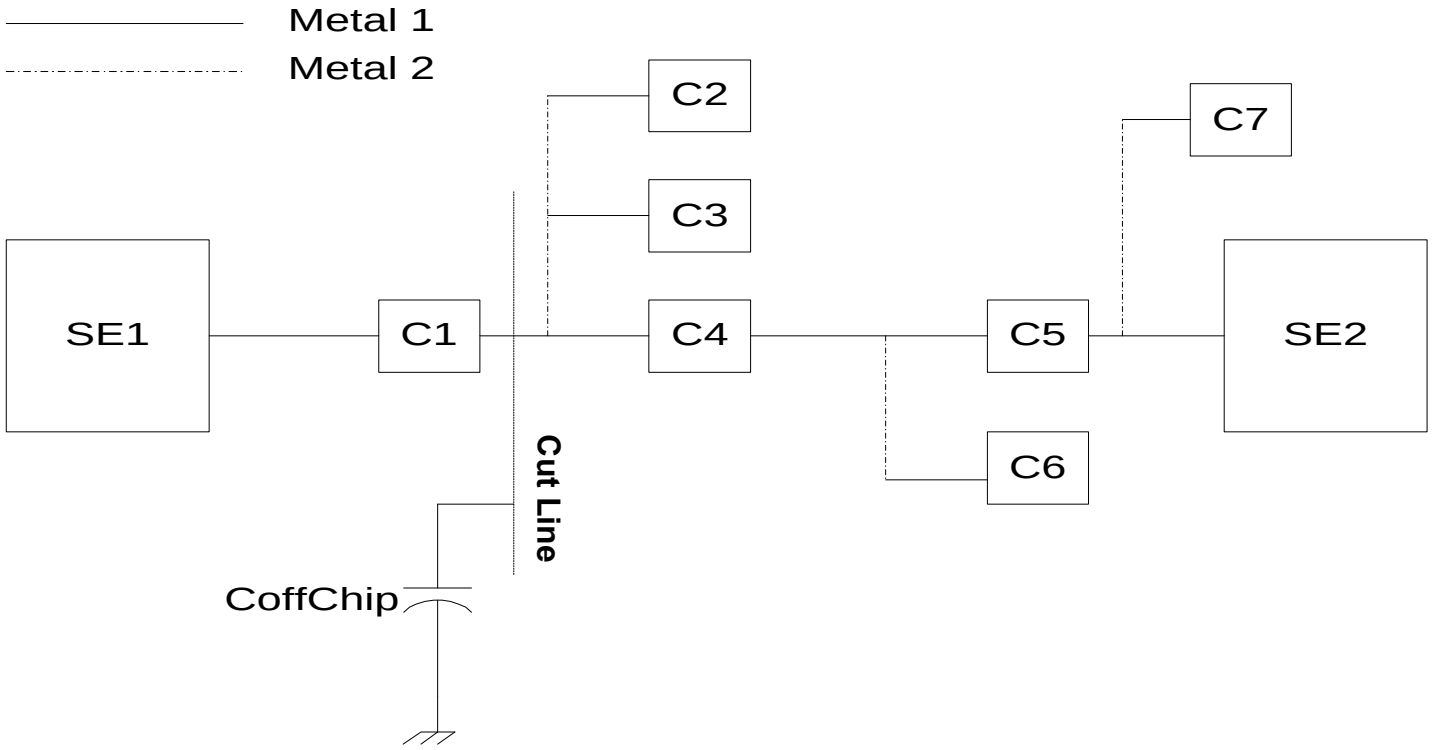
Cutset

- Based on hypergraph model H = (V, E)
- Cost 1: c(e) = 1 if e spans more than 1 block
- Cutset = sum of hyperedge costs
- Efficient gain computation and update



cutset = 3

Delay Model



path π: $SE_1 \rightarrow C_1 \rightarrow C_4 \rightarrow C_5 \rightarrow SE_2$.

$$Delay\pi = CD_{SE1} + CD_{C1} + CD_{C4} + CD_{C5} + CD_{SE2}$$

$$CD_{C1} = BD_{C1} + LF_{C1} * (Coffchip + CINP_{C2} + CINP_{C3} + CINP_{C4})$$

Delay

$$\textbf{Delay(Pi)} = \sum_{cell \in Pi} Delay(cell) + \sum_{net \in Pi} Delay(net)$$

$$\textbf{Delay(Pi)} = \sum_{cell \in Pi} Delay\ (cell\ )$$

$$Objective \quad : \underset{Pi \in P}{Max} \left(Delay\ \ (\ Pi\ ) \right)$$

Pi: is any path Between 2 cells or nodes

P : set of all paths of the circuit.

Power

The average dynamic power consumed by CMOS logic gate in a synchronous circuit is given by:

$$P_i^{average} = 0.5 \cdot \frac{V_{dd}^2}{T_{cycle}} \cdot C_i^{Load} \cdot N_i$$

Ni : is the number of output gate transition per cycle(switching Probability)

C_i^{Load} : Is the Load Capacitance

Power

$$C_i^{Load} = C_i^{basic} + C_i^{extra}$$

C_i^{basic} : Load Capacitances driven by a cell before Partitioning

C_i^{extra} : additional Load due to off chip capacitance.(cut net)

Total Power dissipation of a Circuit:

$$P_\zeta = \beta \cdot \frac{V_{dd}^2}{T_{cycle}} \cdot \sum_{i \in \zeta} \left(C_i^{basic} + C_i^{extra} \right) \cdot N_i$$

Power

$$C_i^{extra} >> C_i^{basic}$$

C_i^{extra} : Can be assumed identical for all nets

$$objective : \sum_{i \in \zeta v} N_i$$

ζv :Set of Visible gates Driving a load outside the partition.

Balance

The Balance as constraint is expressed as follows:

$$cells / blocks - Tol \leq Block_i \leq cells / blocks + Tol$$

$$Tol = cells / block * PercentTol$$

However balance as a constraint is not appealing because it may prohibits lots of good moves.

Objective : |Cells(block1) ï Cells(block2)|

Fuzzy logic for cost function

- Imprecise values of the objectives
  - best represented by linguistic terms that are basis of fuzzy algebra
- Conflicting objectives
- Operators for aggregating function

fuzzy logic for Multi-objective cost function

1. The cost to membership mapping.
2. Linguistic fuzzy rule for combining the membership values in an aggregating function.
3. Translation of the linguistic rule in form of appropriate fuzzy operators.

Some fuzzy operators

And-like operators

- Min operator $\mu = \min(\mu_1, \mu_2)$
- And-like OWA

$$\mu = \beta * \min(\mu_1, \mu_2) + \tfrac{1}{2}(1-\beta)(\mu_1 + \mu_2)$$

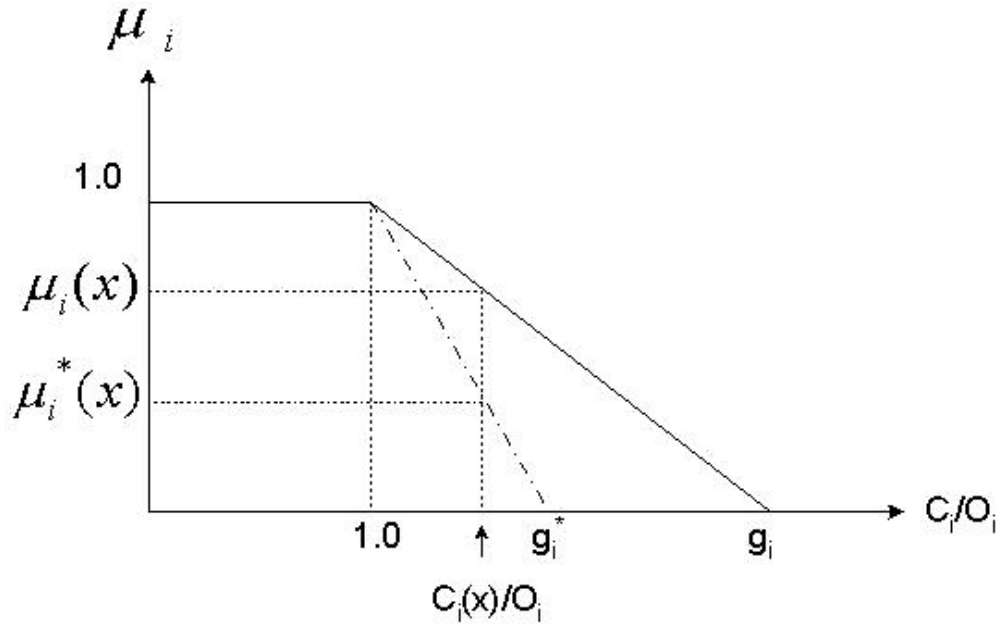
Or-like operators

- Max operator $\mu = \max(\mu_1, \mu_2)$
- Or-like OWA

$$\mu = \beta * \max(\mu_1, \mu_2) + \tfrac{1}{2}(1-\beta)(\mu_1 + \mu_2)$$

Where β is a constant in range [0,1]

mbership functions



Where O_i and C_i are lower bound and actual cost of objective ñiò

$\mu_i(x)$ is the membership of solution x in set ñgood áiôò

g_i is the relative acceptance limit for each objective.

zzy linguistic rule

Å A good partitioning can be described by the following fuzzy rule

IF solution has

small cutset **AND**

low power **AND**

short delay **AND**

good Balance.

THEN it is a good solution

Fuzzy cost function

The above rule is translated to AND-like OWA

$$\mu(x) = \beta \times \min\left(\mu_C, \mu_P, \mu_D, \mu_B\right)$$

$$+(1-\beta)\times \frac{1}{4}\left(\mu_C + \mu_P + \mu_D + \mu_B\right)$$

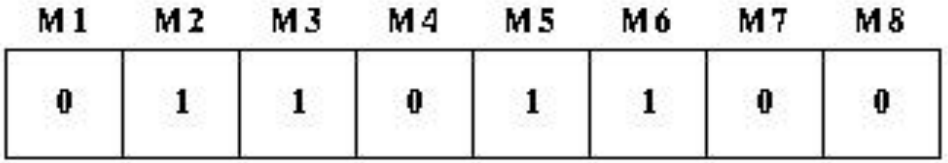
$\mu(x)$ Represent the total Fuzzy fitness of the solution, our aim is to Maximize this fitness.

$\mu_C, \mu_P, \mu_D, \mu_B$ Respectively (Cutset, Power, Delay , Balance) Fitness.

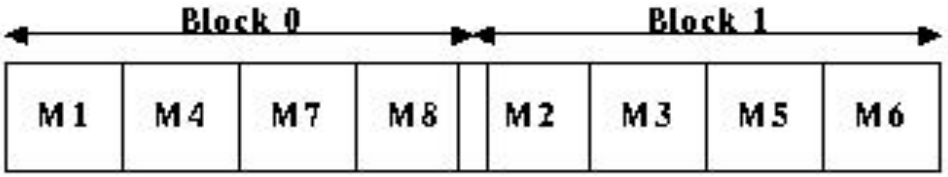
ultiobjective Partitioning

Algorithm (Genetic_Algorithm)
Construct_Population(N_p);
  For j = 1 to N_p
    Evaluate_Fitness (Population[j])
  End For;
  For i = 1 to N_g
    For j = 1 to N_o
      (x,y) ← Choose_parents;
      offspring[j] ← Crossover(x,y)
    EndFor;
    Population ← Select (Population, offspring, N_p)
    For k = 1 to N_p
      Apply Mutation (Population[k])
    EndFor;
  EndFor;

Solution representation



(a) Group Number Encoding



(b) Permutation with Seperator Encoding.

A implementation

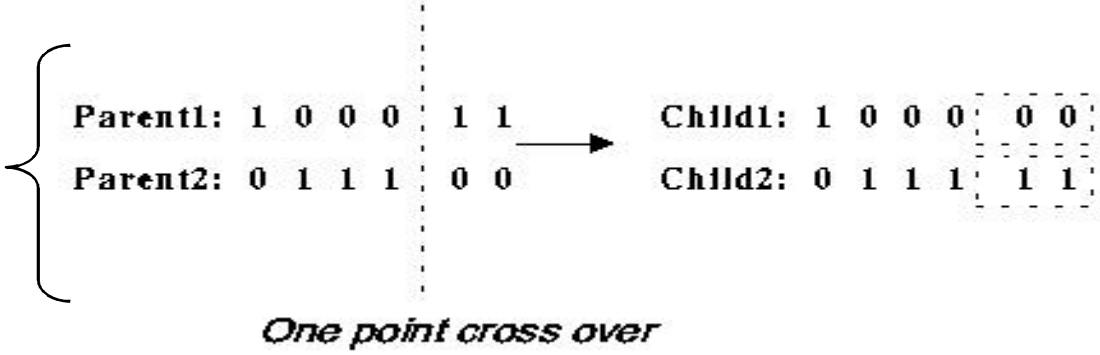
- Different population sizes
- Parent selection: Roulette wheel
  - The probability of selecting a chromosome for crossover is

$$P_{choice} = \frac{\varepsilon(x)}{\sum_{i=1}^{Np} \varepsilon(i)}$$

N_p is the population size

GA implementation

- Simple single point crossover:



One point cross over

- Selection before mutation
- Roulette wheel (rlt)
- Elitism random (ernd)

Tabu Search

Algorithm Tabu_Search
Start with an initial feasible solution $S \in \Omega$
Initialize tabu list and aspiration level
For fixed number of iterations **Do**
    Generate neighbor solutions $V^* \subset N(S)$
    Find best $S^* \in V^*$
    If move S to S* is not in T **Then**
        Accept move and update best solution
        Update T and AL
    Else If Cost(S*) < AL **Then**
        Accept move and update best solution
    Update T and AL
End If
End For

TS implementation

- Neighbor solution
  - Change the block of a randomly selected cells.
- The Tabu list size depends on the circuit size.

TS implementation

Tabu list

- Store index of one of the swapped cell.
- Various sizes for tabu list.

Aspiration Level

- The best neighbor is better than the global best.

perimental Results

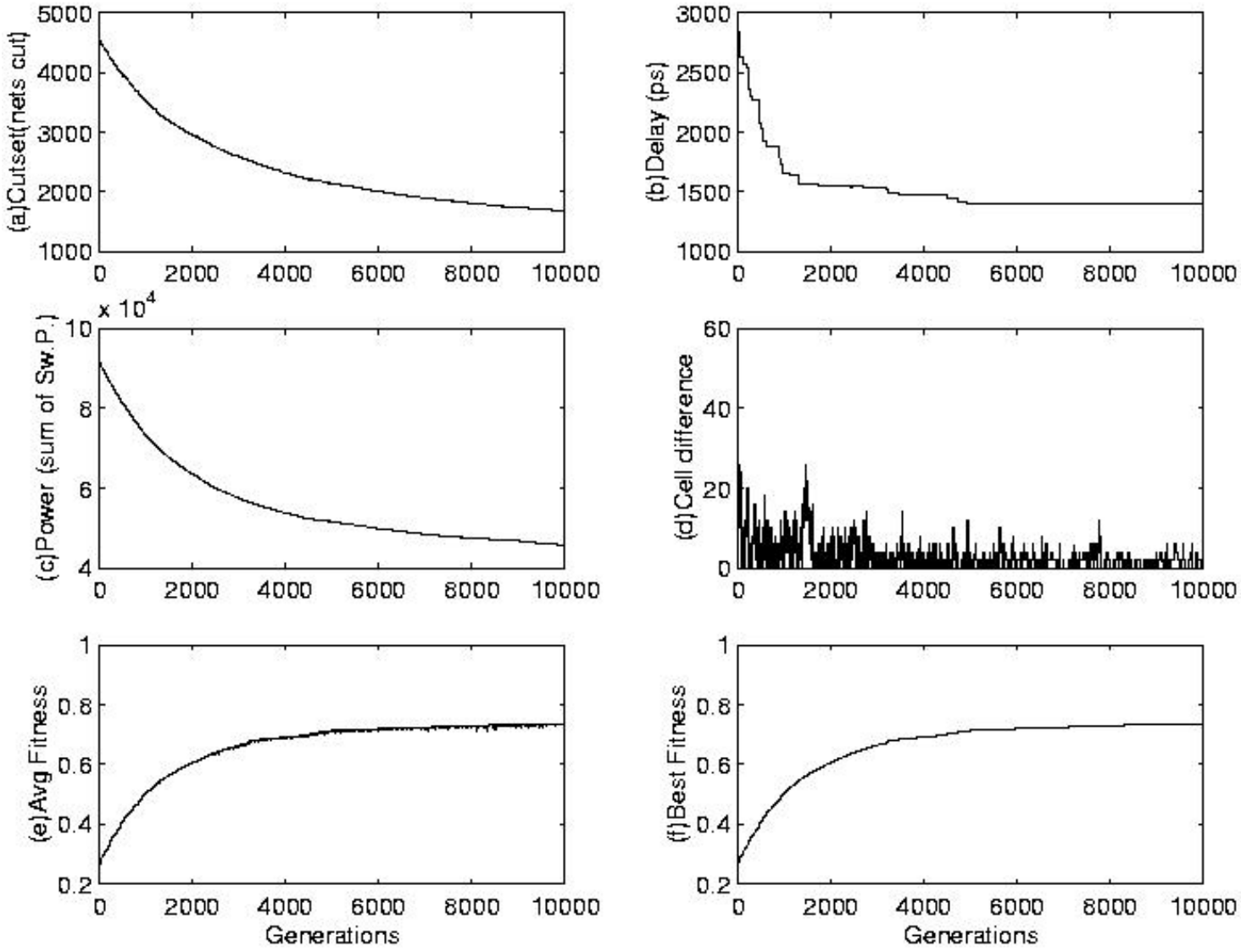
Name	Number of cells	number of nets
s298	136	130
s386	172	165
s641	433	410
s832	310	291
s953	440	417
s1196	561	547
s1238	540	526
s1488	667	648
s1494	661	642
s2081	122	121
s3330	1962	1888
s5378	2994	2944
s9234	5845	5822
s13207	8652	8530
s15850	10384	10296

ISCAS 85-89 Benchmark Circuits

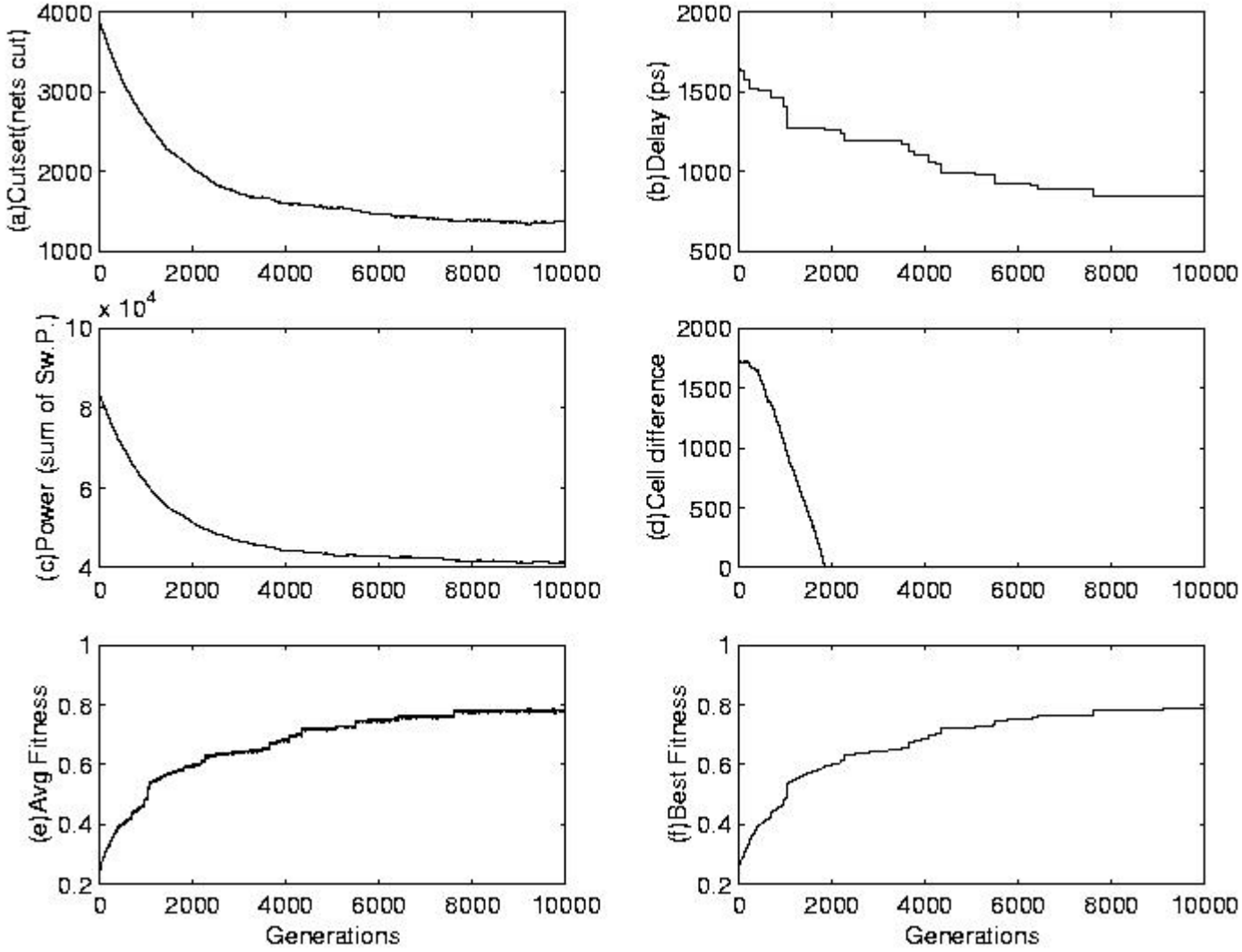
GA Vs Tabu Multi-objective

	GA						TS					
Circuit	D (ps)	Cut	P(sp)	$\mu(x)$	T(s)	Best(s)	D (ps)	Cut	P(sp)	$\mu(x)$	T(s)	Best(s)
S298	233	19	1013	0.79	123	43	197	24	926	0.81	62.00	21.00
S386	356	36	1529	0.75	163	151	366	30	1426	0.76	82.00	77.00
S641	1043	45	2355	0.83	1868	1540	889	59	2281	0.85	939.00	818.00
S832	444	45	3034	0.68	289	276	446	50	2731	0.68	148.00	80.00
S953	526	96	2916	0.69	618	182	466	99	2518	0.73	313.00	225.00
S1196	396	123	5443	0.76	375	373	301	106	4920	0.80	184.00	134.00
S1238	475	127	5713	0.72	397	365	408	79	4597	0.75	187.00	160.00
S1488	571	104	5648	0.71	1238	1183	528	98	5529	0.72	616.00	405.00
S1494	614	102	5474	0.70	1228	1040	585	101	5339	0.71	616.00	427.00
S2081	302	26	787	0.73	94	32	225	17	770	0.79	47.00	16.00
S3330	571	299	10358	0.79	2096	2074	533	295	10298	0.80	1078.00	994.00
S5378	587	573	18437	0.79	2687	2686	590	430	16527	0.80	1338.00	1100.00
S9234	1313	1090	38149	0.72	5963	5949	1052	918	34055	0.77	2992.00	2821.00
S13207	1399	1683	45611	0.74	8098	8097	843	1332	41114	0.79	4001.00	3690.00
S15850	1820	2183	51747	0.74	10214	10206	1411	1671	47480	0.79	5131.00	5130.00

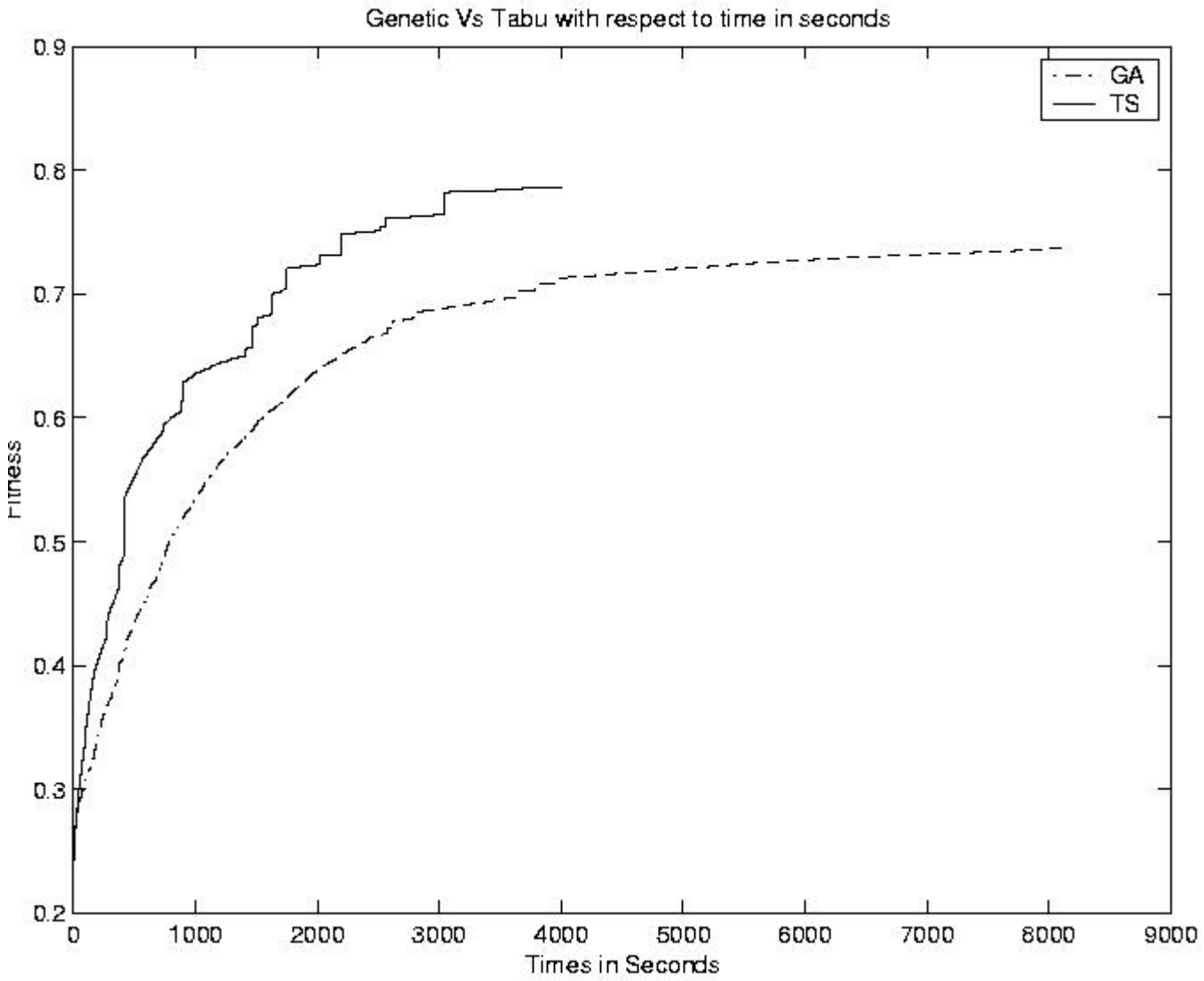
rcuit S13207 GA



ircuit S13207 TS



S13207 GA Vs TS time



Conclusion

- The present work successfully addressed the important issue of reducing power and delay consumption in VLSI circuits.
- The present work successfully formulate and provide solutions to the problem of multiobjective VLSI partitioning
- TS partitioning algorithm outperformed GA in terms of quality of solution and execution time

Thank you.