A Comparative Study on Optimization Algorithms in PowerSynth 2

Mehran Sanjabiasasi[1], Imam Al Razi[2], H. Alan Mantooth[1], Yarui Peng[1]

[1]EECS, University of Arkansas, Fayetteville, AR

[2]Intel Corporation, Hillsboro, OR

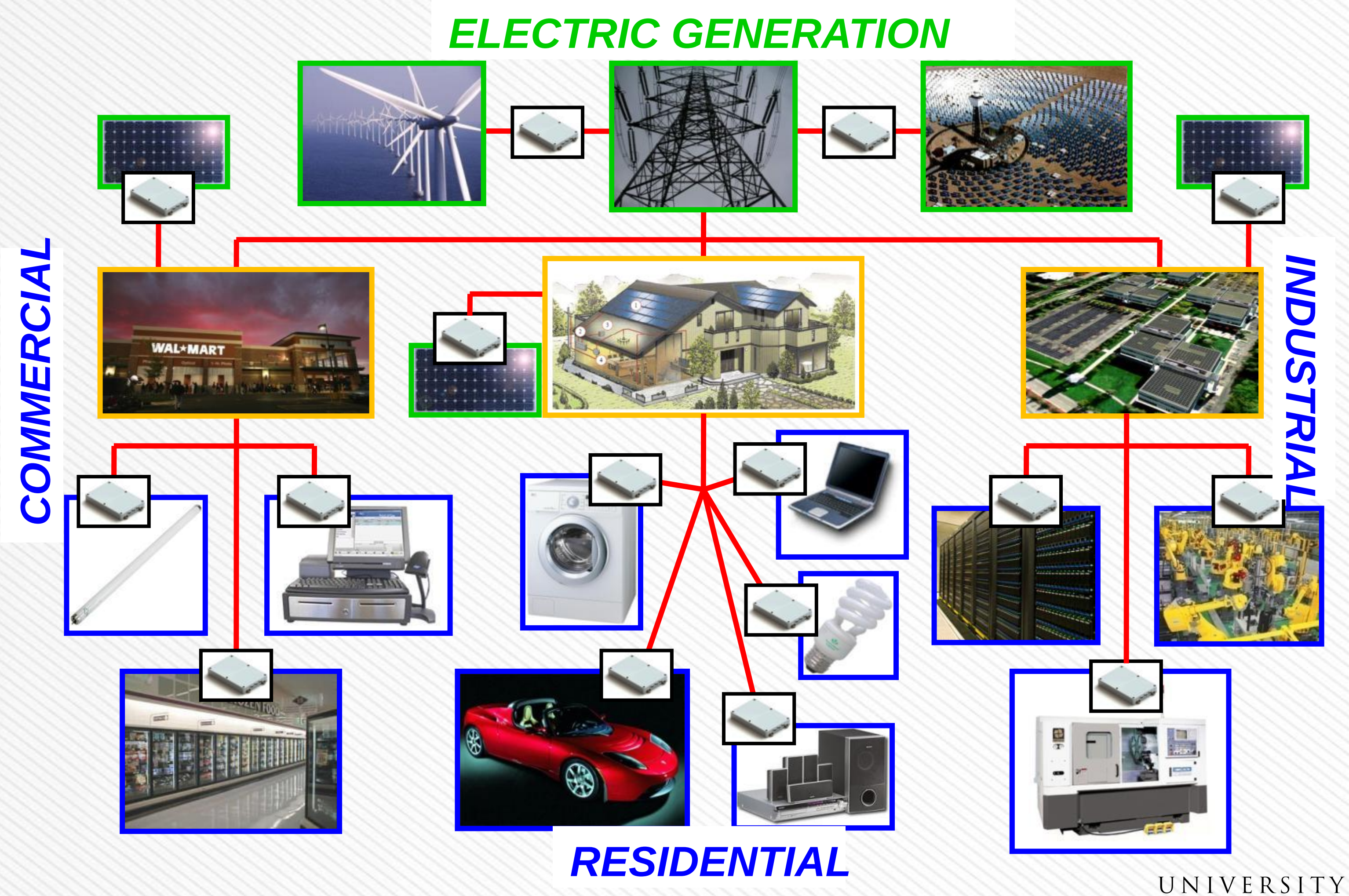
UNIVERSITY OF ARKANSAS

https://e3da.csce.uark.edu yrpeng@uark.edu ☎ +1 (479) 575-6043

Power Electronics is Everywhere

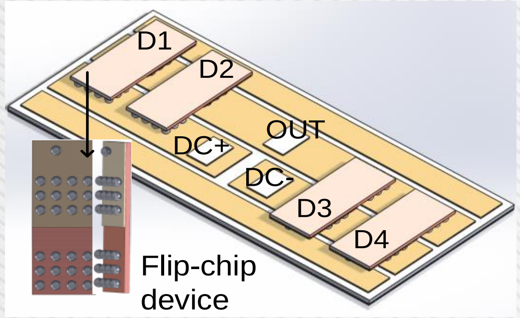
Power converters are essential parts of power systems

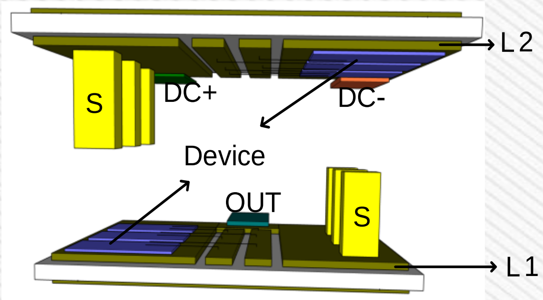
- ❑ Electric automobiles
- ❑ Aircrafts
- ❑ Smart grid
- ❑ Consumer electronics
- ❑ ....

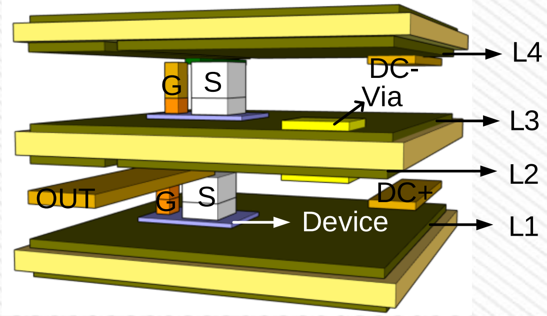


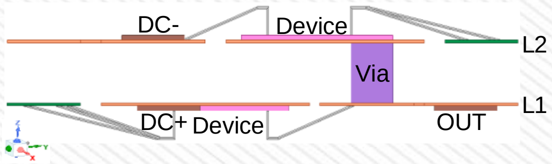
UNIVERSITY OF ARKANSAS

- **MCPM is a critical component in power conversion applications.**
- **Module design, mostly focusing on three parts:**
  - layout generation
  - modeling
  - optimization



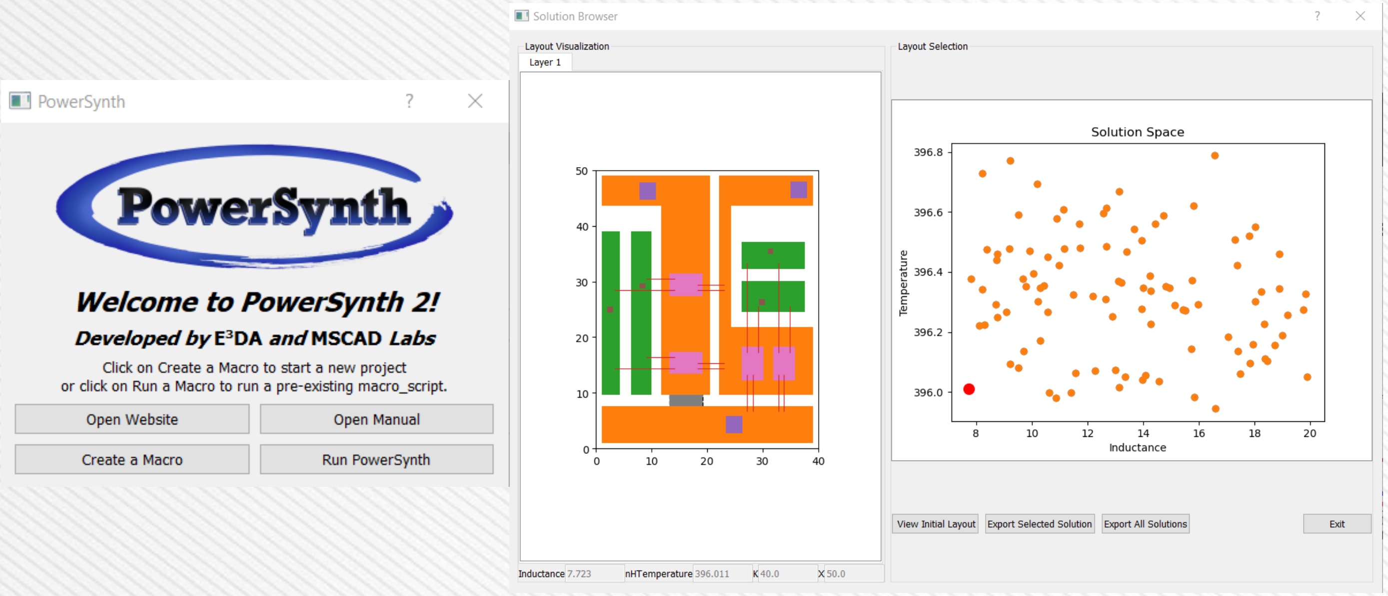






[1] I. Al Razi, Q. Le, T. Evans, H. A. Mantooth, and Y. Peng, “PowerSynth 2: Physical Design Automation for High-Density 3D Multi-Chip Power Modules,” IEEE Transactions on Power Electronics, 2023.

- A software for MCPM layout optimization.
- Hierarchical corner stitching data structure with a constraint graph.
- PowerSynth 2: (2D/2.5D/3D) layout, lab validation

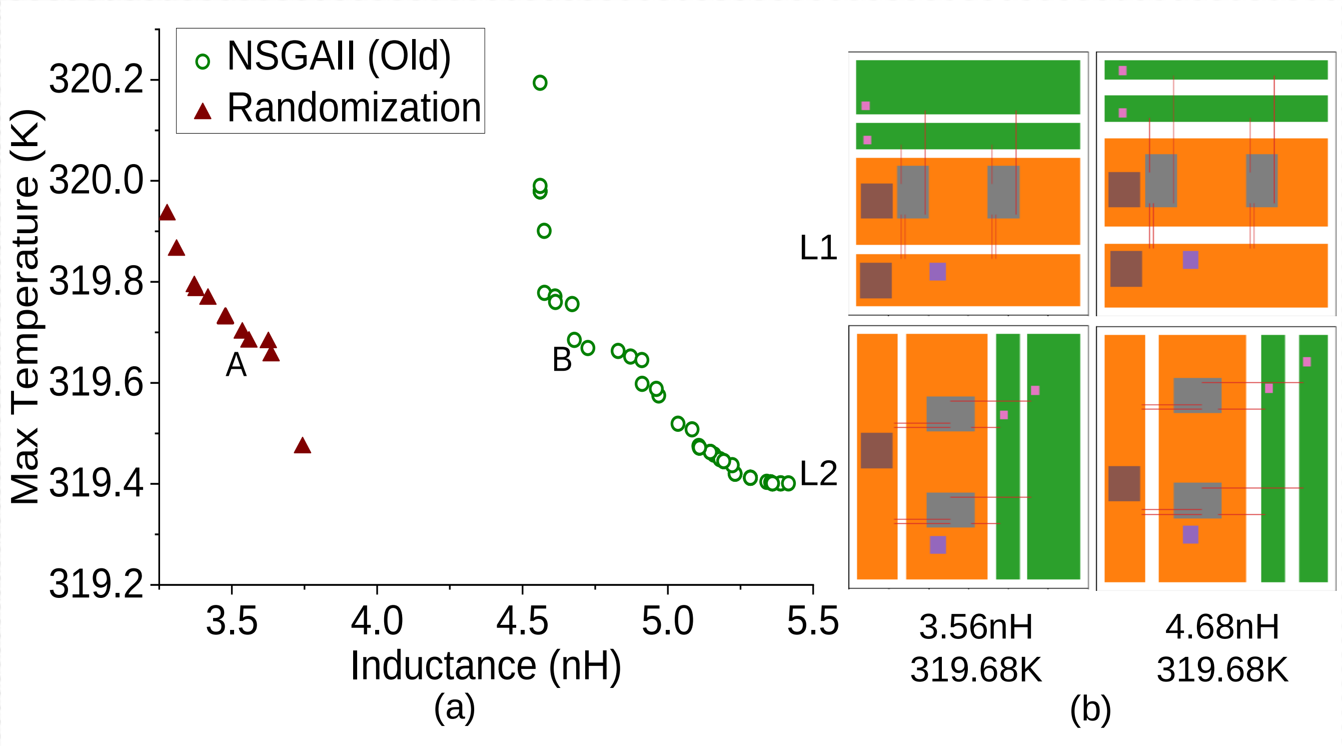


PowerSynth 2 GUI

[1] I. Al Razi, Q. Le, T. Evans, H. A. Mantooth, and Y. Peng, "PowerSynth 2: Physical Design Automation for High-Density 3D Multi-Chip Power Modules," IEEE Transactions on Power Electronics, 2023.

PowerSynth 2 Existing Optimization Algorithms:

- **Randomization (RAND)**
  - + build-in algorithm
  - + follows the hierarchy structure
  - - blind search
- **NSGA-II**
  - + fitness evaluation guided
  - - unaware of the layout hierarchy



(a) Parato Front comparison and (b) two selected layouts.

Our Contributions

1. A **hierarchical** optimization framework for power module layout optimization
2. A **new** layout optimization algorithm based on Multi-Objective Particle Swarm Optimization (**MOPSO**)
3. A **qualitative comparative study** using various performance indicators on the latest optimization Algorithms

UNIVERSITY OF ARKANSAS

Hierarchical Optimization

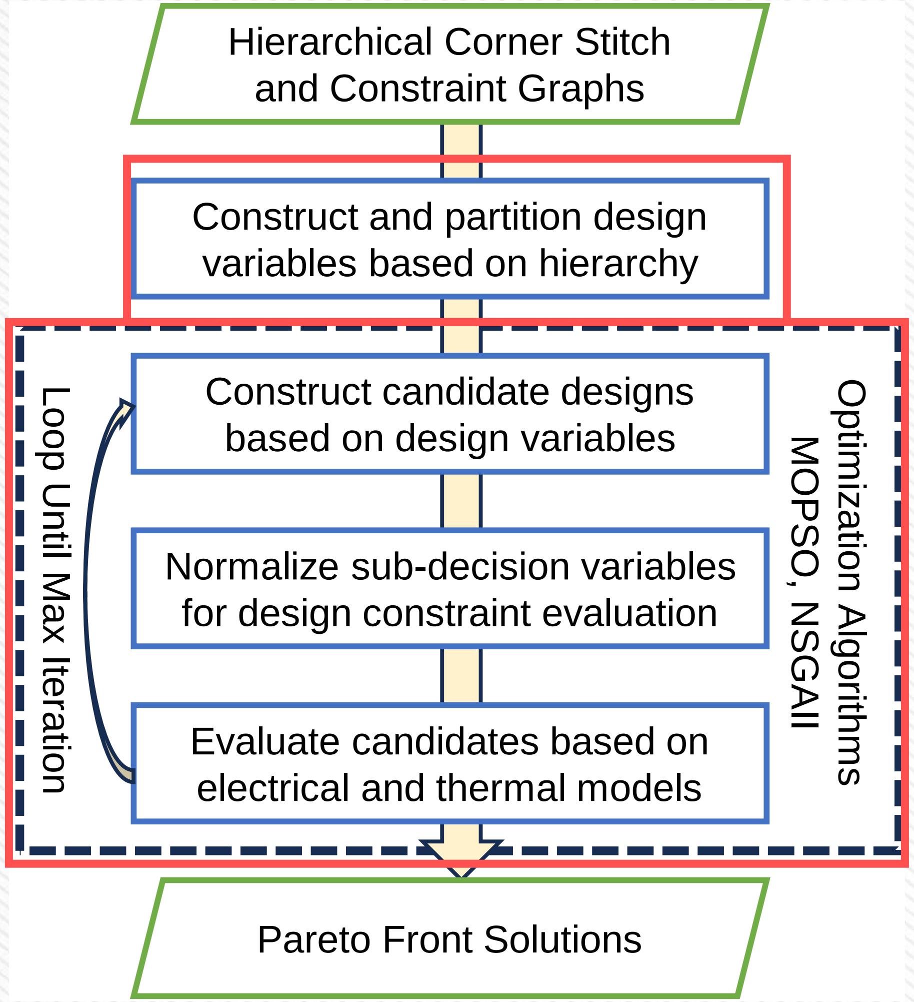
Design String

- Decision Variables are grouped into a list of design strings.

$$\{[H_1, H_2, H_3, \ldots, H_n][h_{11}, h_{12}, h_{13}, \ldots, h_{1m}][h_{21}, h_{22}, h_{23}, \ldots, h_{2m}] \ldots [h_{n1}, h_{n2}, h_{n3}, \ldots, h_{nm}]\}$$

$$\text{subject to} \sum_{n=1}^{N} H_n = Room \quad \text{and} \quad \sum_{m=1}^{M \le N} h_{nm} = H_n$$

An example horizontal design string



Hierarchical Optimization

❑ MOPSO

- MOPSO was proposed by [1] in 2004.
- Stochastic search, population-based, evolutionary computation technique
- Like PSO, particles are sharing information and moving towards the global best particles and their own personal (local) best memory.
- The inertia weight α equals 0.9
- b and c are random numbers [0,1]
- Mutation probability: 1/DV#
  - applied on 1/3 population

Algorithm 1: MOPSO Workflow

```
Initialize External Repository (ER)
for each particle do
    Initialize the particle's position & velocity randomly
    Evaluate particle
    Update the best personal value and archive it in ER
while Maximum Iteration is not met do
    for each particle do
        Select a leader from the ER
        V = a * V + b * (Pbest − POP) + c * (BestER(h) − POP)
        POP = POP + V
        Perform Mutation with the Swarm
        Evaluate particle
        Update the best personal value and archive it in ER
    update leader in ER
Return ER as a Pareto Front Solutions
```

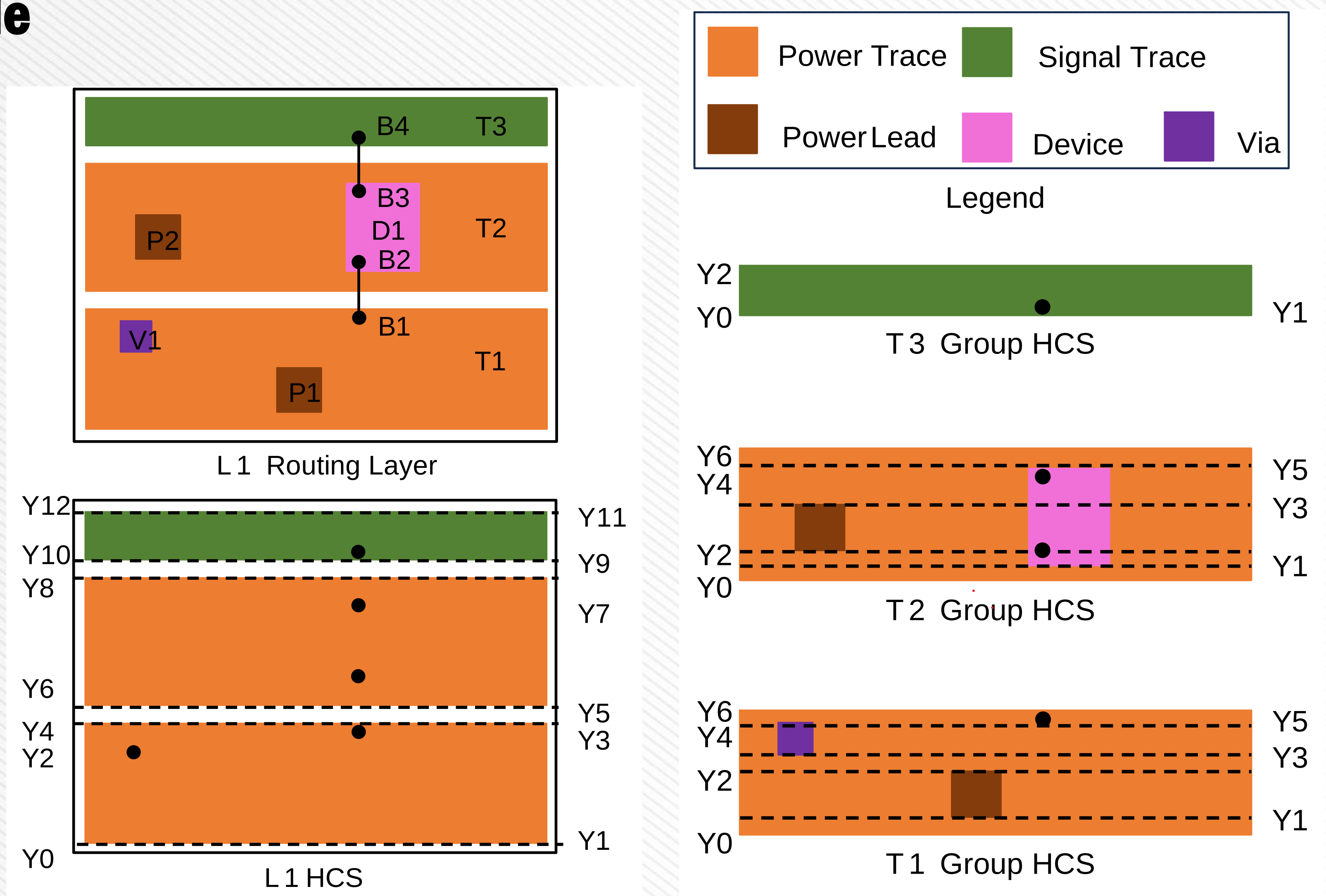
[1] C. Coello, G. Pulido, and M. Lechuga, "Handling Multiple Objectives with Particle Swarm Optimization," IEEE Transactions on Evolutionary Computation, 2004.

UNIVERSITY OF ARKANSAS

Example Layout

❑ Illustration of an Example

```
L1 Z-
+ T1 power 7 7 30 9
   +B1 power 25 15
   +L1 power_lead 19 7.5
    +V1 Via 8 12
+ T2 power 7 17 30 10.5
   +L2 power_lead 9 19
   +D1 MOS 24.5 18
      +B2 power 25 19
      +B3 power 25 23
+ T3 signal 7 29 30 2.5
   +B4 signal 25 30
L2 Z+
+ T1 power 7 24 30 10
   +B5 power 21 25
   +L3 power_lead 9 26
+ T2 power 7 11.5 30 11.5
   +V1 Via 8 12
   +D2 MOS 20 13.5R180
      +B6 power 21 14
      +B7 power 21 18
+ T3 signal 7 7 30 3
   +B8 power 21 8
```

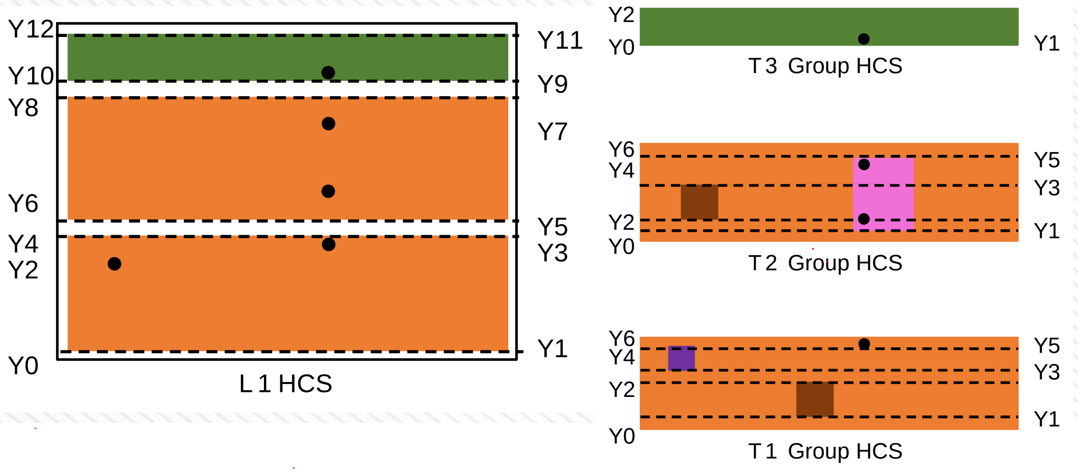


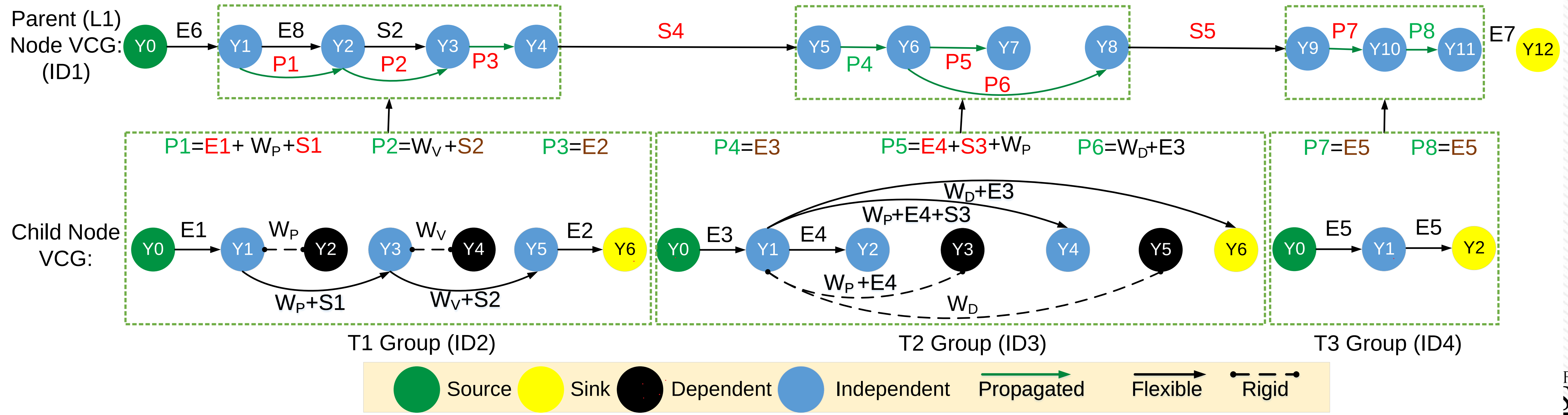
(a) Geometry script, (b) Horizontal corner Stitch (HCS) of L1, (c) HCS of each group

UNIVERSITY OF ARKANSAS

❑ Constraint Graph and Design String

ID1 = [P1, P2, P3, S4, P5, P6, S5, P7]
ID2 = [E1, S1, S2, E2]
ID3 = [E3, E4, S3]
ID4 = [E5]
VDV = [[P1, P2, P3, S4, P5, P6, S5, P7], [E1, S1], [E4, S3]]





Hierarchical Optimization

Algorithm 2: Construction of Decision Variables (DV)

```
Inputs: HCG, VCG, Layout hierarchy, User constraints
Outputs: HDV, VDV
for each tree from leaf to root do
    Perform bottom-up constraint propagation
Evaluate the root node and determine the ID and
  numbers of flexible edge weights
Append them to the HDV list
for each sub-tree from root to leaf do
    Determine the ID and Numbers of flexible edge
      weights
    Append them to the HDV list
Perform Optimization Algorithms
```

Bottom-up propagation

Top-Down propagation

VDV = [[P1, P2, P3, S4, P5, P6, S5, P7], [E1, S1], [E4, S3]]

UNIVERSITY OF ARKANSAS

Hierarchical Optimization

Algorithm 3: Optimization Algorithms Workflow

```
Generate initial population for each sub-decision
  variables randomly
for each population do
    Normalized each sub-decision variables
    Evaluate and find the best solution
while Maximum Iteration is not met do
    Generate new solutions based on the existing ones
    for new solutions do
        Normalized each sub-decision variables
        Evaluate and find the best
    update the best solution
Find out the Pareto Front from the best solutions
```

Individuals
Position, Velocity

Electrical and Thermal

Crossover, Mutation
Velocity, Position

Non-dominated

Normalized Pop1 = [[0.12, 0.26, 0.05, 0.03, 0.17, 0.11, 0.14, 0.12], [0.34, 0.66], [0.92, 0.08]]

UNIVERSITY OF ARKANSAS

Hierarchical Optimization

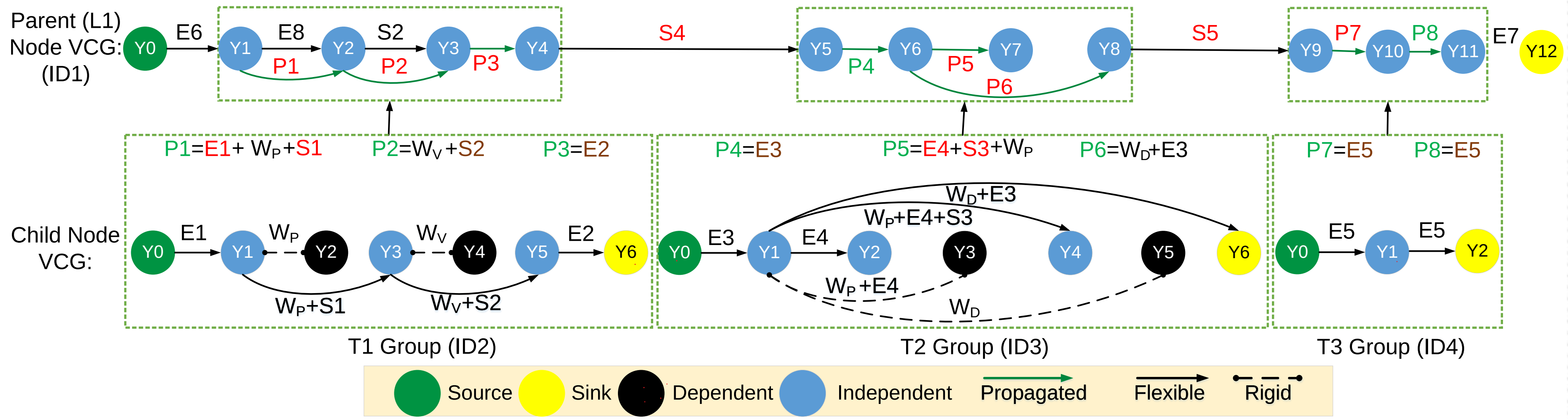
- **E.g., ID1 has a room of 10:**
  - The edge weights are calculated by each ID list multiplied by the available room:

Normalized Pop1 = [[0.12, 0.26, 0.05, 0.03, 0.17, 0.11, 0.14, 0.12], [0.34, 0.66], [0.92, 0.08]]

ID1 : [**1.2**, 2.6, 0.5, 0.3, **1.7**, 1.1, 1.4, 1.2]

ID2 : [0.408, 0.792]

ID3 : [1.564, 0.136]



Performance Indicator

Comparison between indicators [1]

- **Modified Generational Distance (GD^{+}) & Inverted Generational Distance (IGD^{+}):**
  - Average distance between given Pareto to reference Pareto
- **Epsilon (ϵ):**
  - Maximum distance between the reference and the given Pareto
- **Error Ratio (ER):**
  - Number of non-dominated solutions in reference Pareto
- **Hypervolume (HV):**
  - Size of the space covered by Pareto

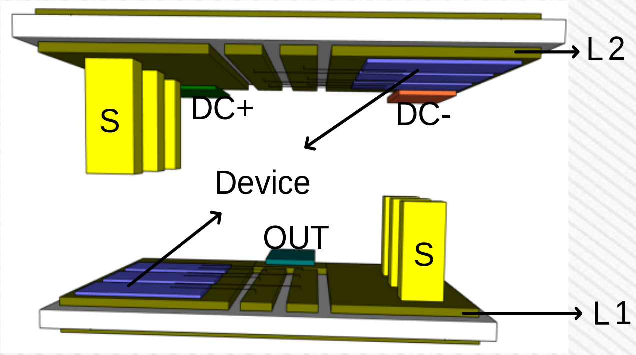
Indicators	GD+	IGD+	ϵ	HV	ER
Convergence	+	+	+		
Cardinality				+	+
Distribution		+		+	
Spread			+	+	
Preference	Lower	Lower	Lower	Higher	Lower

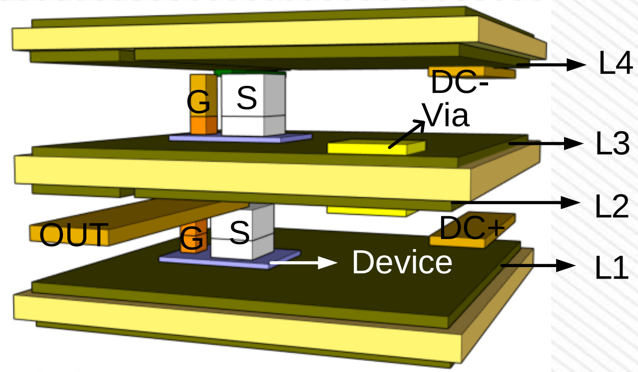
[1] C.Audet, J. Bigeon, D. Cartier, S. Le Digabel, and L. Salomon, "Performance Indicators in Multiobjective Optimization," European Journal of Operational Research, 2021.

Case Study Summary

❑ Summary of test case designs

- Fitness Function = Minimizing (power loop inductance, maximum temperature)
- The reference Pareto Front is obtained by combining the results from all algorithms
- The number of layout Generations:
  - 2D cases: all algorithms: 400
  - 3D cases: RAND:400, NSGAII(Old, New) , MOPSO : 200





Case Study Summary

❑ Summary of test case designs [1]

Design	2D/3D	Packaging	Paral. Sw.	Cooling	Final Size (mm^2)
Case 1	2D SiC	Wire-bonded	2	Single-sided	40 × 50
Case 2	2D SiC	Wire-bonded	2	Single-sided	40 × 50
Case 3	3D SiC	Metallic post	3	Double-sided	30 × 15
Case 4	3D SiC	Wire-bonded	2	Double-sided	30 × 30

- **Case 1 and Case 2 are 2D Half-bridge SiC modules with two switches in parallel.**
- **Case 3: four-layer 3D Half-bridge SiC modules, metallic post-type vias.**
- **Case 4: two-layer 3D Half-bridge SiC modules, bonding wires.**

[1] PowerSynth source code: https://github.com/e3da/

UNIVERSITY OF ARKANSAS

Analysis Result Summary

❑ Result Comparison

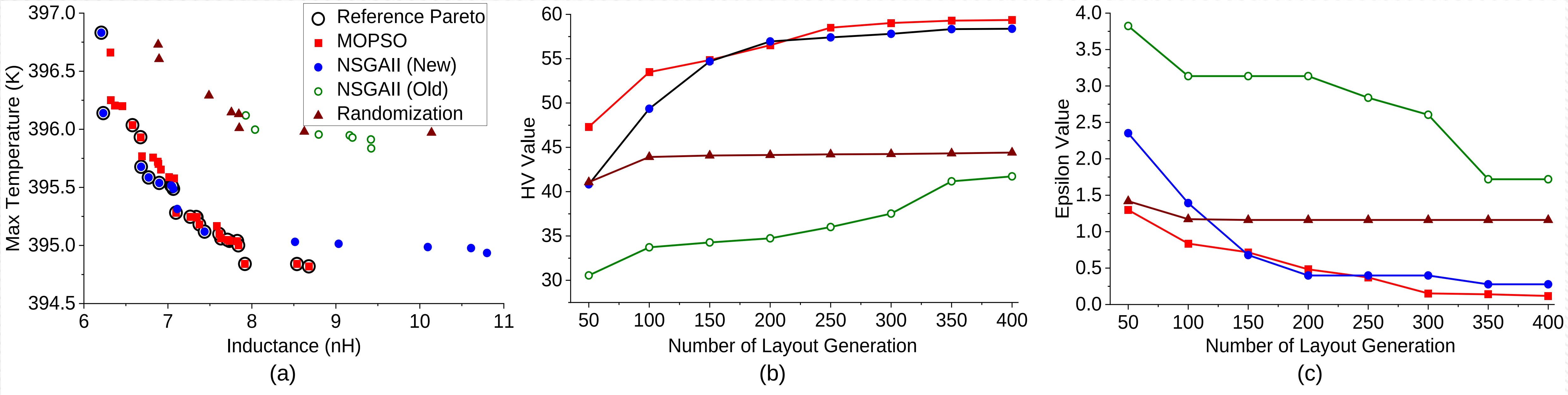
- MOPSO outperforms NSGA-II in GD+ and IGD+ convergence indicators for all Cases.
- MOPSO and NSGA-II obtain similar HV in Case 1-3.
- In Case 4, MOPSO clearly outperforms NSGA-II in all indicators.
- Overall, MOPSO is faster to converge to the Pareto and outperforms NSGA-II in general.

Case	Indicator	Current		Proposed	
		RAND	NSGAII	MOPSO	NSGAII
Case 1	DG+	1.172	2.149	**0.049**	0.360
	IGD+	1.165	0.948	**0.042**	**0.056**
	Epsilon	1.719	1.160	**0.118**	0.278
	HV	41.72	44.41	**59.37**	**57.38**
	ER	1.000	1.000	**0.407**	0.593
Case 2	DG+	0.310	1.295	**0.044**	0.197
	IGD+	0.977	2.007	**0.032**	0.173
	Epsilon	3.731	5.214	**0.419**	**0.516**
	HV	42.71	25.83	**47.95**	**45.61**
	ER	0.977	1.000	**0.295**	0.727
Case 3	DG+	0.129	0.104	**0.023**	0.049
	IGD+	0.205	0.111	**0.032**	0.043
	Epsilon	0.771	0.380	**0.156**	0.221
	HV	5.252	5.743	**6.267**	**6.206**
	ER	0.893	0.893	**0.571**	0.714
Case 4	DG+	0.354	0.078	**0.000**	0.303
	IGD+	0.598	0.080	**0.004**	0.284
	Epsilon	1.163	0.137	**0.024**	0.626
	HV	3.411	6.161	**6.617**	4.886
	ER	0.607	0.893	**0.571**	1.000

Analysis Results: Case 1

❑ Case 1

- **The proposed method performs better than the current method.**
- **MOPSO obtained a better Pareto Front solution than NSGA-II.**
- **MOSPO generates optimal results faster than NSGA-II for this case.**

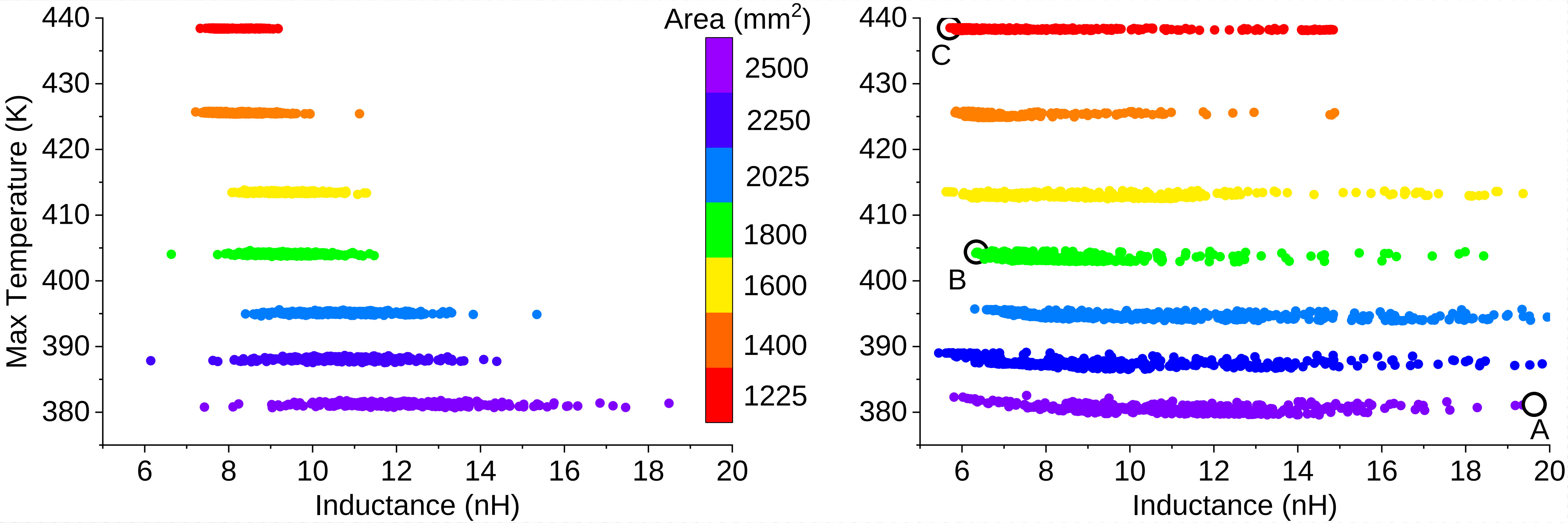


Comparison of different algorithms in Case 1: (a) Pareto Front, (b) Hypervolume, (c) Epsilon

UNIVERSITY OF ARKANSAS

❑ Case 1

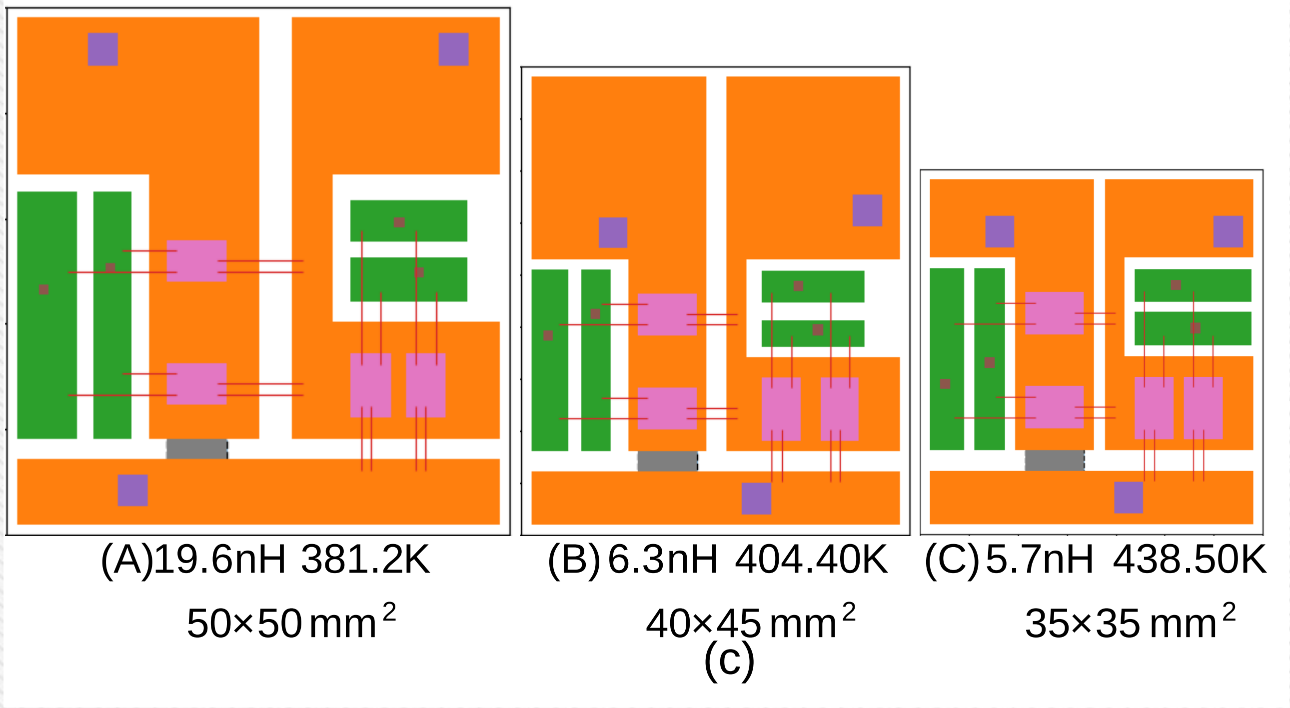
- **Seven distinct floorplan sizes, ranging from 1225 mm to 2225 mm.**
- **MOPSO solution space is more spread and distribution than that of RAND.**



Solution space of seven different floorplan sizes: (a) RAND, (b) MOPSO

Analysis Results: Case 1

- **Case 1**
  - **Layout A: highest inductance value, lowest temperature.**
  - **Layout C: worse thermal results, better electrical performance.**
  - **Layout B: balanced tradeoff between the two extreme choices.**

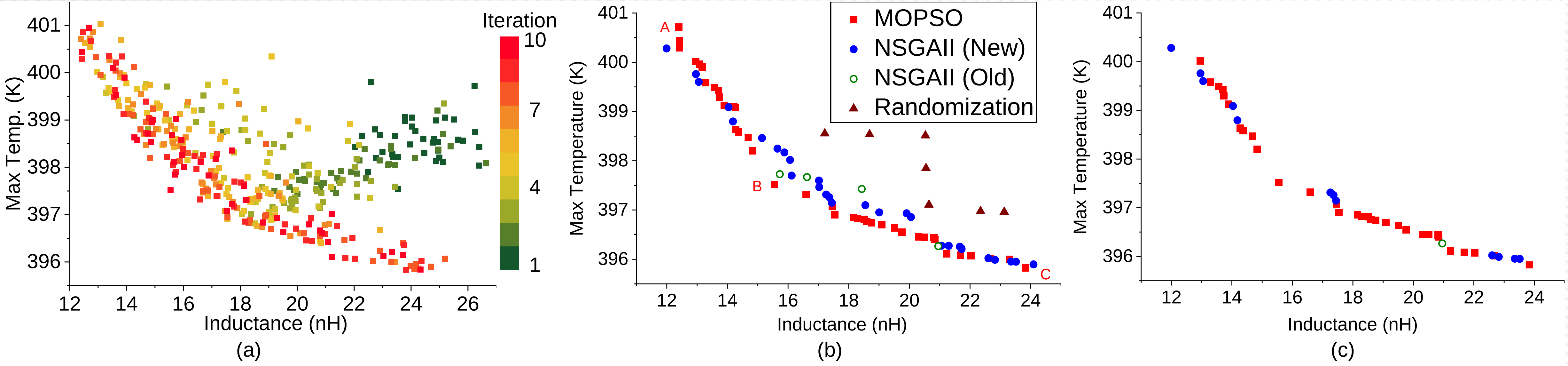


Three selected solutions by MOPSO

Analysis Results: Case 2

❑ Case 2

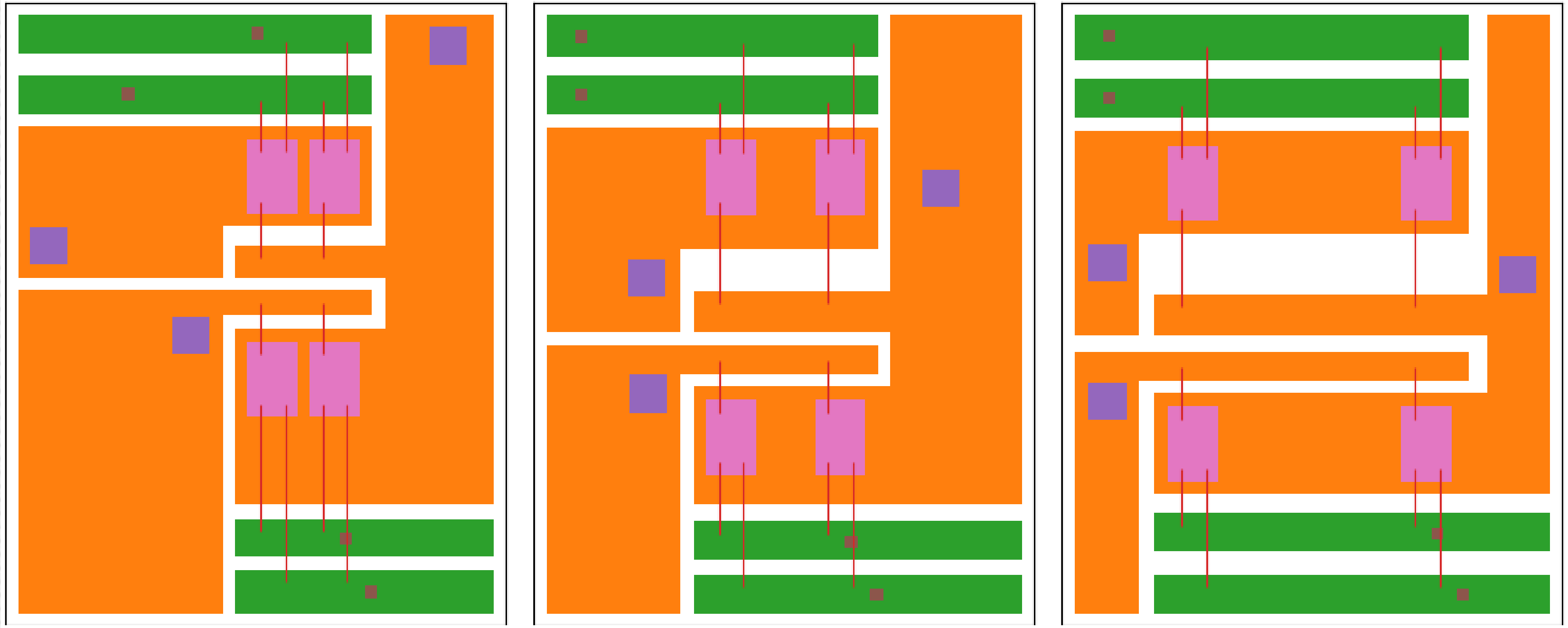
- The proposed method performs better than the current method.
- MOPSO obtains a comparable Pareto Front solution to NSGA-II.
- The solution space expands and concentrates towards the Pareto Front over iterations.
- MOPSO demonstrates a higher convergence speed.



Comparison of different algorithms: (a) MOPSO solution space over iteration, (b) Pareto Front (c) Global best solutions

Analysis Results: Case 2

- Case 2



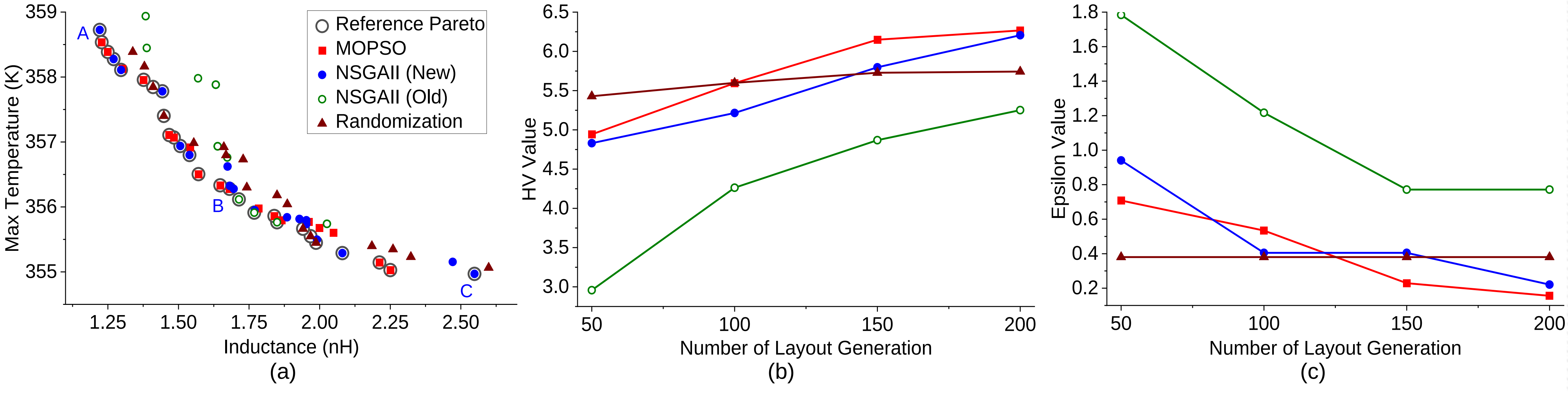
(A) 12.4nH 400.7K (B) 15.5nH 397.5K (C) 23.8nH 395.8K

Three solutions were generated by MOPSO. All layouts are 40×50 mm^2

UNIVERSITY OF ARKANSAS

❑ Case 3

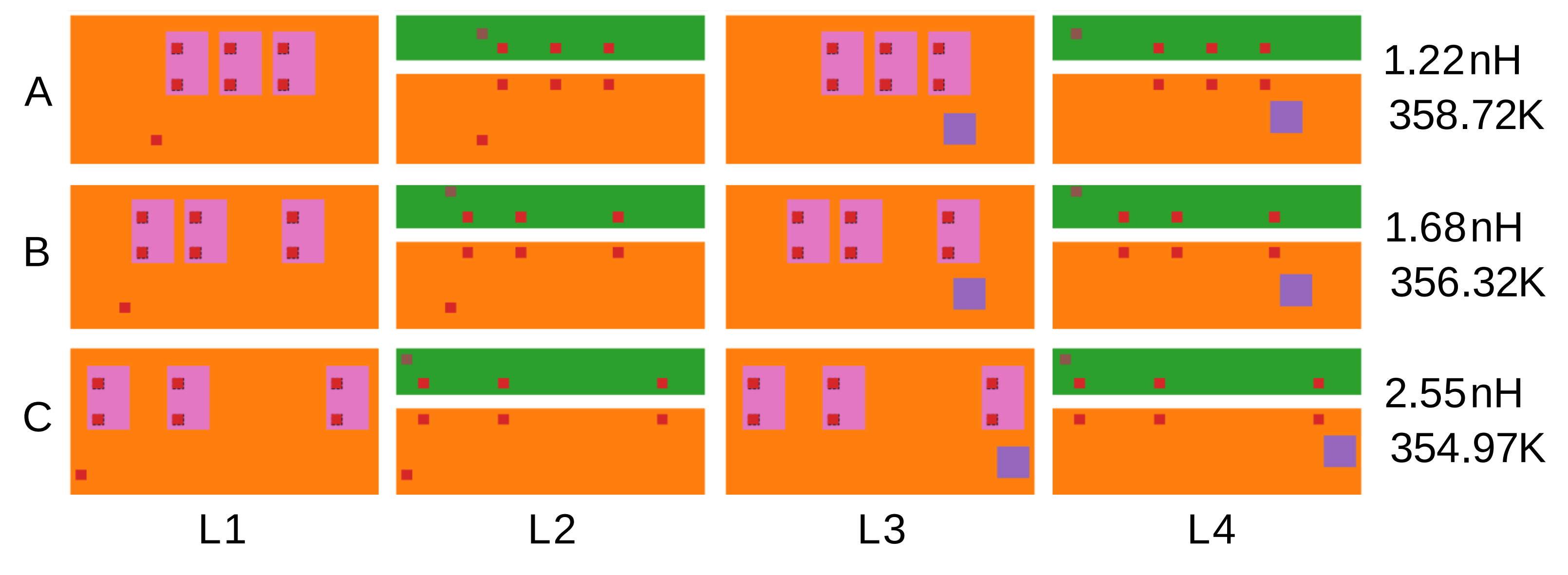
- The proposed methodology outperforms the current method.
- MOPSO obtained a comparable Pareto Front solution to NSGA-II.



Comparison of different algorithms in Case 3: (a) Pareto Front, (b) Hypervolume, (c) Epsilon

Analysis Results: Case 3

- Case 3
  - Layout A: lowest inductance value, highest temperature.
  - Layout C: better thermal performance, worse electrical results.
  - Layout B: balanced Electro-thermal solution.



Three solutions were generated by NSGAII. All layouts are 30×15 mm^2

Analysis Results: Runtime

❑ Runtime Comparison

- The proposed algorithms have comparable runtime to NSGA-II, but RAND is 5-7 times faster due to parallelization.
- NSGA-II and MOPSO can also be accelerated with parallel computing in the future.
- Hierarchical optimization does not introduce much runtime overhead

Case	RAND	NSGAII(Old)	NSGAII(New)	MOPSO
Case 1	15.5	16.0	16.0	16.0
Case 2	15.4	14.8	15.0	15.0
Case 3	20.0	100	103	103
Case 4	9.0	63.0	64.0	65.0

Conclusions and Future Work

Conclusions

- PowerSynth 2 optimization algorithms have been updated from planar to hierarchical.
- A new MOPSO is proposed as a faster alternative to existing NSGAII and RAND.
- Five indicators and runtime are considered to evaluate other aspects of the algorithms.
- Hierarchical optimizations significantly improve result quality and solution space size with minimum runtime overhead
- MOPSO is comparable to NSGA-II in terms of distribution and spread but achieves a faster convergence speed.

Future Work

- Proposed method will be open-sourced and released as PowerSynth v2.1 for testing.
- Implement Parallel computing to accelerate the runtime of these algorithms further.

UNIVERSITY OF ARKANSAS

Thank you

UNIVERSITY OF ARKANSAS

https://e3da.csce.uark.edu yrpeng@uark.edu +1 (479) 575-6043