

# New Heuristic for Simple Offset Assignment (SOA) Problem

Authors:

Hesham S. Ali\*

Hatem M. EL-Boghdadi\*

Samir S. Shaheen\*

*\*Cairo University, Egypt*

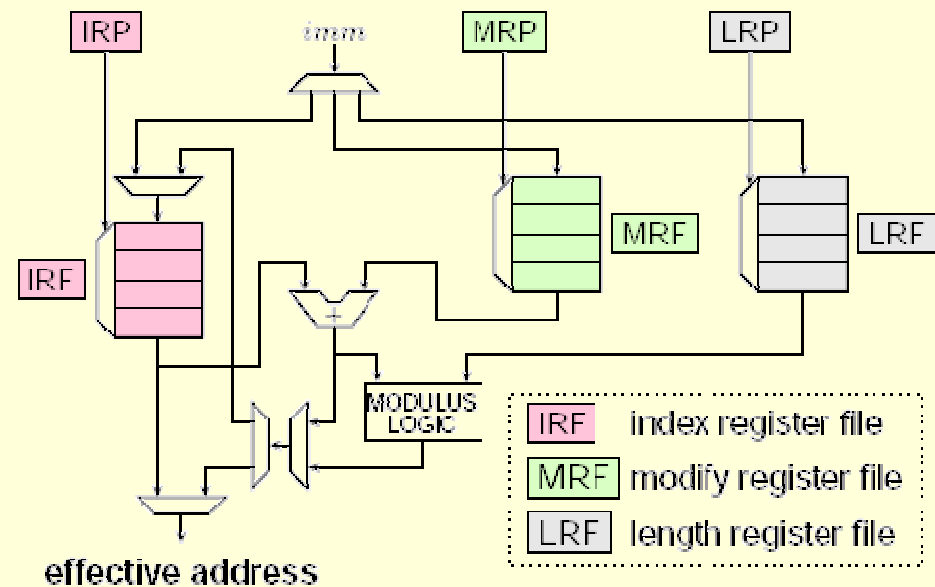
# Outline

- **Background**
  - Address Generation Unit (AGU)
  - SOA Problem
- **Approaches to SOA**
  - Basic heuristic by Liao
  - Tie Break Function by Leupers and Marwedel
  - Other approaches.
- **New Heuristic for SOA**
  - Introduction of the Effective Tie Break Function
  - Motivational example
- **Experiments & Results**
- **Summary**
- **Thanks**

# Background – AGU

- AGU is a common H/W feature in modern embedded DSP processors.
- It has autoinc/autodec operations: executed in || with program instr. Operations save address arithmetic instructions, provided that proper offset assignment of RAM variables is made.
- IR is loaded with an initial address, then, through || autoinc/autodec ops, IR is loaded with the effective addr., instead of having a separate instruction to reload its address.
- The SOA problem addresses an AGU model with only one IR and without MRs.

## Address Generation Unit (AGU)

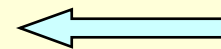
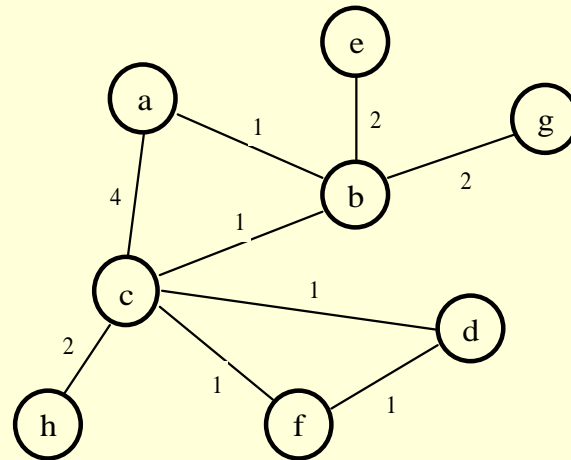


# Background – SOA Problem

A **Basic Block** is a sequence of program code statements with no branching.

The **access sequence** is the set of variables ordered by their appearance order in the code.

```
c = c + d + f
a = h - c
b = b + e
c = g - b
a = a - c
```



*c d f c h c a b e b g b c a c a*



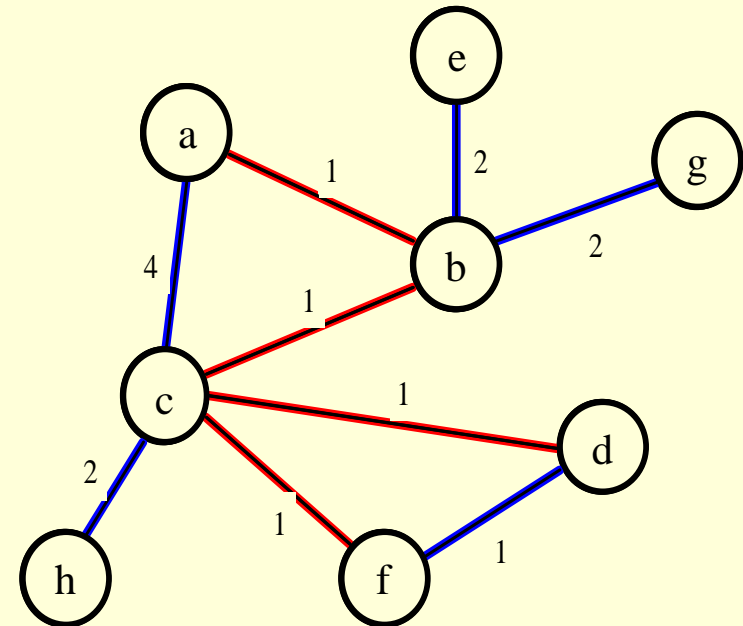
- **Access Graph:** nodes represent program variables, edges (along with their weights) represent the number of access transitions between pairs of variables.
- **Objective:** placement of these variables in consecutive places in memory such that **maximum number of the transitions are implemented by autoinc/autodec operations**

# Background – SOA Problem (cont.)

Liao [1] modeled the SOA problem as finding the Max. Weighted Path Cover (MWPC).

Example, the blue edges in the figure represent the MWPC of the graph, and the variable assignment will be a,c,h,e,b,g,f,d

Edges not included in the path cover (a-b, b-c, c-d, c-f) represent the Assignment Cost. The cost is the sum of weights of these edges. In the shown graph, the cost is 4.



# Approaches to SOA: Liao

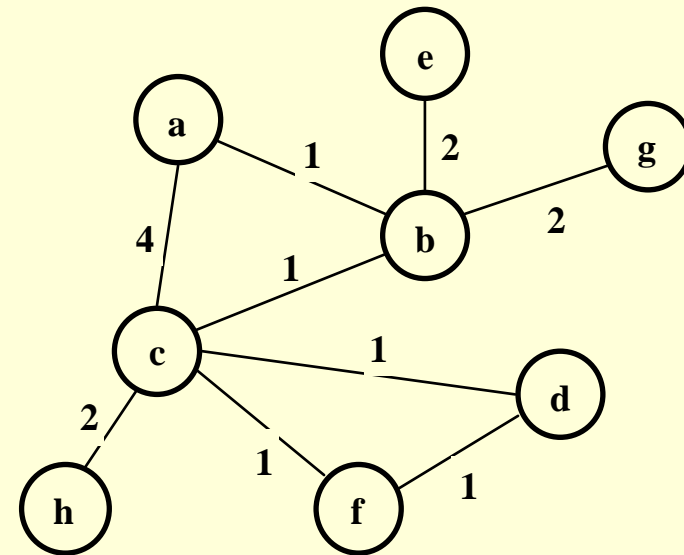
- Greedy algorithm to construct the MWPC, by arranging the graph edges according to their weights and repeatedly trying to include the highest-weight and valid edge in the path cover.
- Jump in cost reduction (about 20%) over the naive assignment (variables are placed in memory based on their access sequence)
- Arbitrarily selecting edges when there are multiple edges with the same weight.

# Approaches to SOA: Leupers and Marwedel (LM)

- LM proposed a *Tie Break Function* (TBF\_LM) of an edge and used it to select between multiple equally-weighted edges. They selected the edge with the least TBF\_LM to be included in the path cover.
- Noticeable improvement with average saving of 3% of cost over Liao's algorithm.

# Approaches to SOA: Leupers and Marwedel (LM) (2)

- $TBF\_LM(e)$  is the total sum of weights of the neighboring edges of  $e$ .
- Example:  $TBF\_LM(b-c) = 13$  and  $TBF\_LM(d-f) = 2$




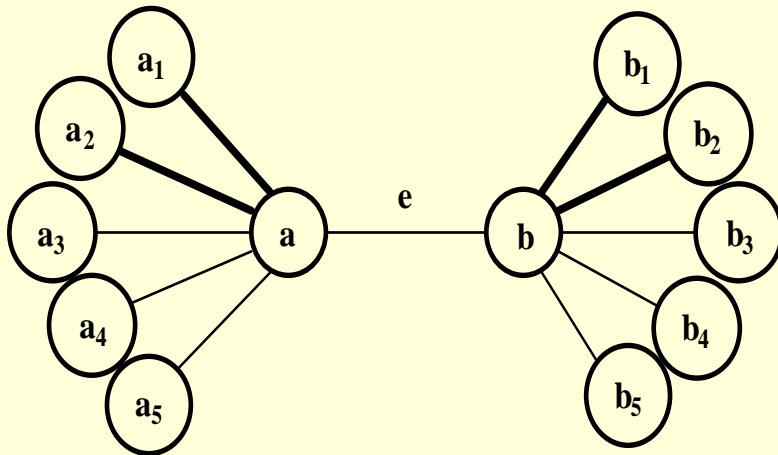
- LM prefer to select the edge with least  $TBF\_LM$ .  
Hypothesis: selecting an edge with high weight would decrease the opportunity of neighboring edges to be selected (later) in the path cover.  
So, an edge with *higher*  $TBF\_LM$  is given *lower selection priority* in order to leave a selection opportunity for edges around it.

# Other approaches

Other approaches [3], [4] used algebraic transformations and instruction re-ordering, which is not the direction of the work presented in this paper.

# New Heuristic – Effective TBF

- Any node in the graph will have at most 2 of its incident edges included in the path cover.
-  it is more effective to consider these edge pairs rather than counting on all the neighboring edges.
- For a node  $n$ , let the highest-weighted pair of edges incident on that node to be referred to as  $MWP(n)$ . See below:




Assume

$w(a-a_1) > w(a-a_2) \dots > > w(a-a_5)$  and  
 $w(b-b_1) > w(b-b_2) \dots > > w(b-b_5)$

$MWP(a) = \text{pair of edges } a-a_1 \text{ and } a-a_2$

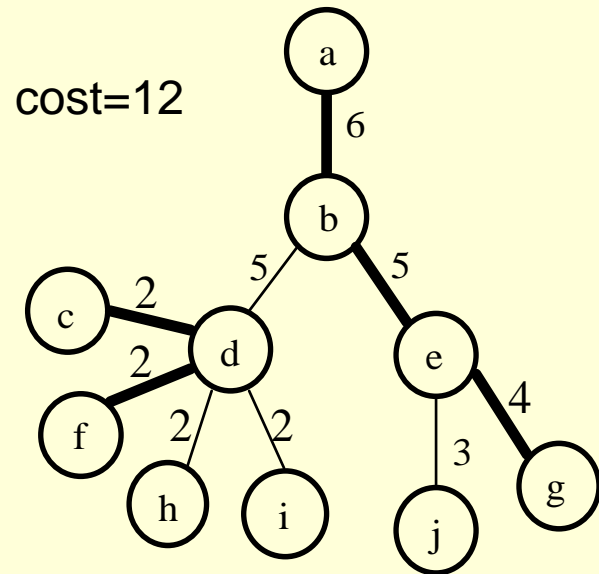
$MWP(b) = \text{pair of edges } b-b_1 \text{ and } b-b_2$

 *Effective Tie-Break Function (ETBF)* to be used rather than TBF\_LM.

$$ETBF(e(a,b)) = MWP(a) + MWP(b).$$

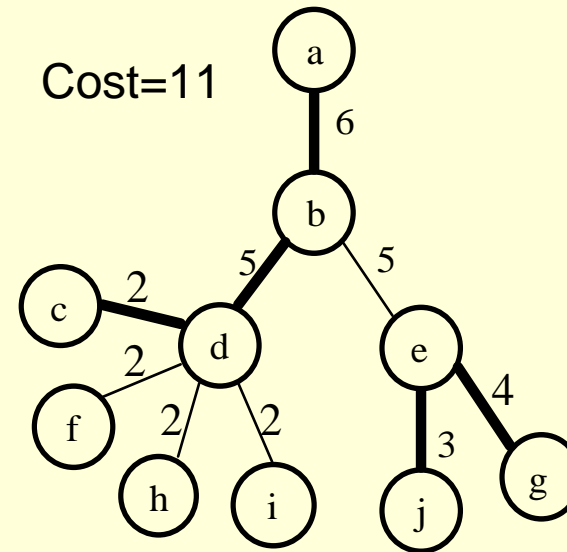
# New Heuristic: Motivational Example

1st edge selected is a-b; then: 2 equally weighted edges b-d and b-e:



**Fig (a) - *TBF\_LM* used**

$TBF\_LM(b-d) = 8 + 11 = 19$   
 $TBF\_LM(b-e) = 7 + 11 = 18,$   
 👉 b-e is selected



**Fig. (b) - *ETBF* used**

$ETBF(b-d) = 4, ETBF(b-e) = 7,$   
 👉 b-d is selected  
 👉 lower cost

The algorithm is simple and in the worst case it defaults to Liao's algorithm.

# Experiments & Results

- Random access sequences:
  - Same methodology as in previous work
  - Do not reflecting the real world code, but represent sparse and dense graphs with a wide ranges of weight values.
- Graphs sizes ranging from 10 to 100 nodes and 20 to 600 edges.  
For each size, 500 random graphs are used and the average value is recorded.
- The new algorithms achieved up to 7% and an average of 4.5% improvement in cost reduction over the LM algorithm.

Graph size G(V, E)	Liao	LM	New algorithm
(10, 20)	459.7	406.97	382.84
(10, 40)	491.1	448.82	434.14
(20, 40)	449.88	426.94	401.62
(20, 190)	1278.46	1130.16	1089.7
(40, 60)	440.07	402.89	389.91
(40, 500)	8307.47	7346.29	7017.18
(60, 100 )	467.47	417.78	397.98
(60, 550 )	8967.5	8570.24	8025.17
(80, 100 )	427.93	392.37	372.94
(80, 570 )	9023.89	8018.63	7550.34
(100, 200)	786.89	744	709.71
(100, 590)	9399.89	8666.7	8131.1

# Future Work

In future work, Ali et al.

- Will apply this work to the more GOA problem
- Use the work with other AGU Configurations  
(with Modify Registers)

# Summary

In their paper, Ali et al. present:

- The Simple Offset Assignment (SOA) problem
- Liao's model: Max Weighted Path Cover of the access graph
- Summary of approaches proposed to solve it.
- Problems resulting from the definition of the Tie-Break Function by Leupers and Marwedel
- A new metric called Effective Tie-Break Function (ETBF)
- An example demonstrating that ETBF gives a correct edge selection decision when TBF\_LM fails to do.
- Experimentation methodology and results of applying Liao, LM and our algorithm to a range of access sequences and compared the results. ETBF results in up to 7% and average of 4.5% improvement over TBF\_LM

**Thank you  
for your attention!**