Clustering Solutions of Multiobjective Function Inlining Problem

Kateryna Muts, Heiko Falk
k.muts@tuhh.de, heiko.falk@tuhh.de
Hamburg University of Technology

Workshop on Worst-Case Execution Time Analysis (WCET 2023)
July 11, 2023

Hard real-time systems have several design criteria.

Worst-Case Execution Time (WCET)

code size

energy consumption

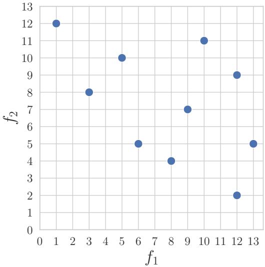
Hard real-time systems have several design criteria.

Worst-Case Execution Time (WCET)	code size	energy consumption
		

Several contradicting objectives $\longrightarrow$ Multiobjective problem

What is a solution of a multiobjective problem?

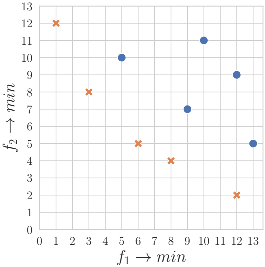
A set of trade-off solutions

Multiobjective problem $\longrightarrow$ 

What is a solution of a multiobjective problem?

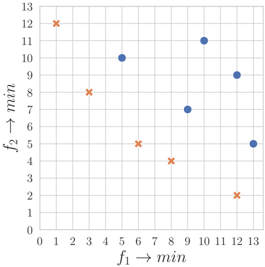
Multiobjective problem $\longrightarrow$

A set of trade-off solutions



What is a solution of a multiobjective problem?

Multiobjective problem ⟶ A set of trade-off solutions



How to choose the best solution?

How to choose the best solution?

System designer's preferences

known

before solution process

unknown

before solution process

How to choose the best solution?

System designer's preferences

known

before solution process

unknown

before solution process

- all but one of the objectives are placed into constraints

How to choose the best solution?

System designer's preferences

known

before solution process

unknown

before solution process

- all but one of the objectives are placed into constraints
- all objectives are combined into a single objective

How to choose the best solution?

System designer's preferences

known *before solution process*

- all but one of the objectives are placed into constraints
- all objectives are combined into a single objective
- a decision maker conducts in direction of the desired solution

unknown *before solution process*

How to choose the best solution?

System designer's preferences

known

before solution process

- all but one of the objectives are placed into constraints
- all objectives are combined into a single objective
- a decision maker conducts in direction of the desired solution

one solution

unknown

before solution process

How to choose the best solution?

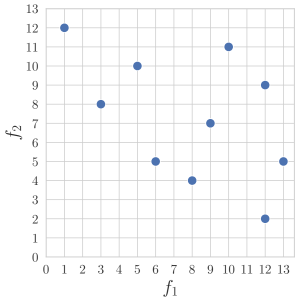
System designer's preferences

known *before solution process*

unknown *before solution process*

- all but one of the objectives are placed into constraints
- all objectives are combined into a single objective
- a decision maker conducts in direction of the desired solution

one solution



How to choose the best solution?

System designer's preferences

known

before solution process

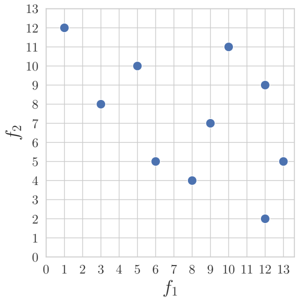
unknown

before solution process

- all but one of the objectives are placed into constraints
- all objectives are combined into a single objective
- a decision maker conducts in direction of the desired solution

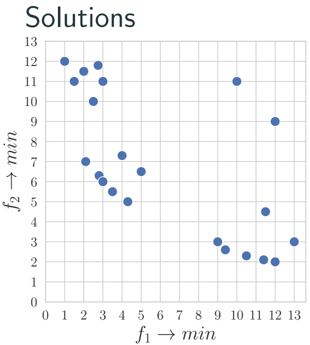
$\downarrow$

one solution



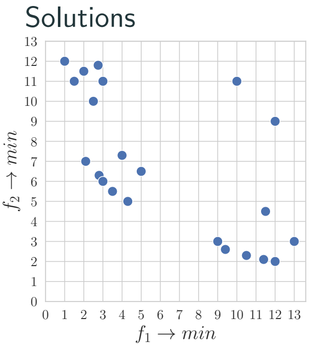
$\longrightarrow$ **one solution**

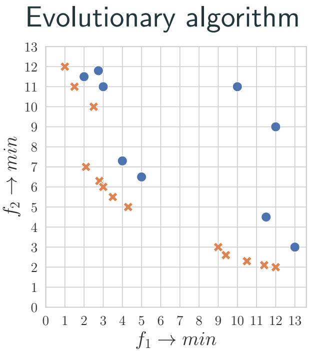
"The magical number seven, plus or minus two" effect[1]: Humans can handle only a limited amount of information simultaneously.



[1] George A. Miller. "The Magical Number Seven, plus or Minus Two: Some Limits on Our Capacity for Processing Information". In: *Psychological Review* 63 (1956), pp. 81–97.

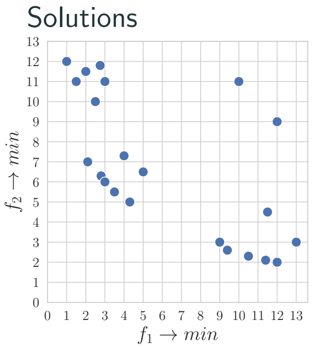
"The magical number seven, plus or minus two" effect[1]: Humans can handle only a limited amount of information simultaneously.

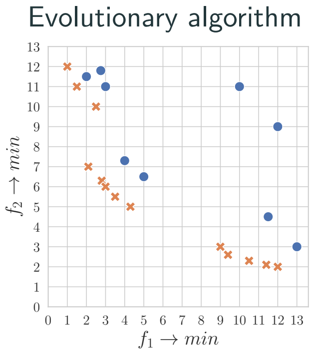


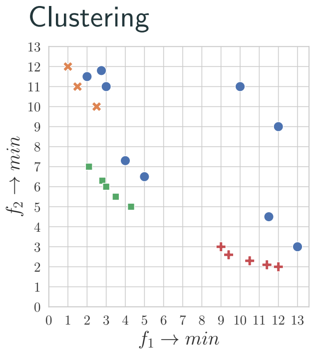


[1] Miller, "The Magical Number Seven, plus or Minus Two: Some Limits on Our Capacity for Processing Information".

"The magical number seven, plus or minus two" effect[1]: Humans can handle only a limited amount of information simultaneously.





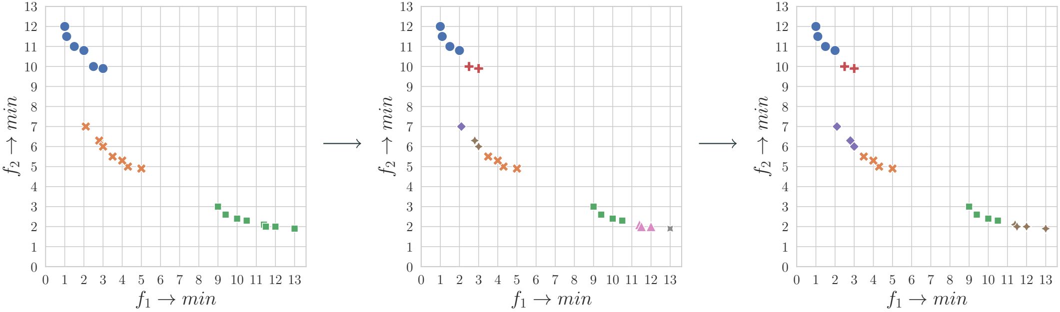


[1] Miller, "The Magical Number Seven, plus or Minus Two: Some Limits on Our Capacity for Processing Information".

How to guarantee that the sizes of clusters are less than a predefined size?

Goal: the size of each cluster is less than or equal to 5

Original clustering | Refine large clusters | Merge small clusters

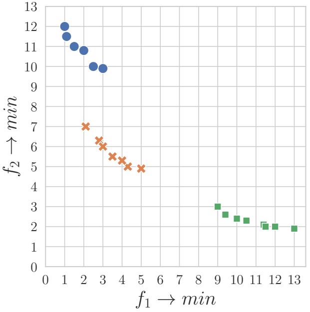


3 clusters | 8 clusters | 6 clusters

How to guarantee that the sizes of clusters are less than a predefined size?

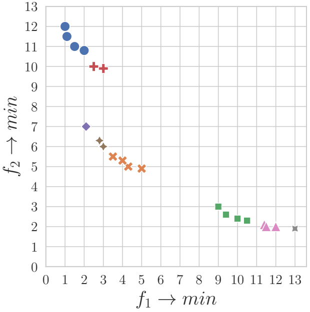
Goal: the size of each cluster is less than or equal to 5

Original clustering



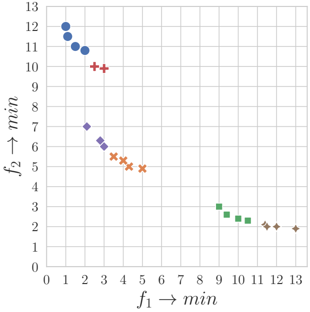
3 clusters

Refine large clusters



8 clusters

Merge small clusters

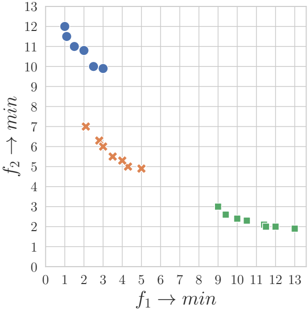


6 clusters

How to guarantee that the sizes of clusters are less than a predefined size?

Goal: the size of each cluster is less than or equal to 5

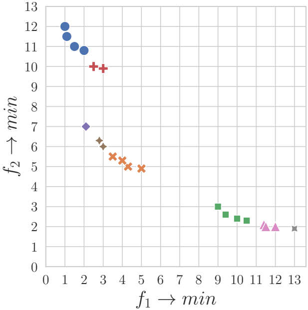
Original clustering



3 clusters

$\longrightarrow$

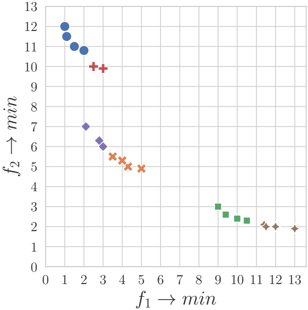
Refine large clusters



8 clusters

$\longrightarrow$

Merge small clusters

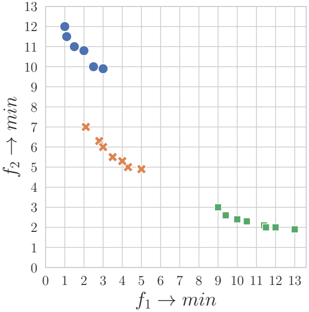


6 clusters

How to guarantee that the sizes of clusters are less than a predefined size?

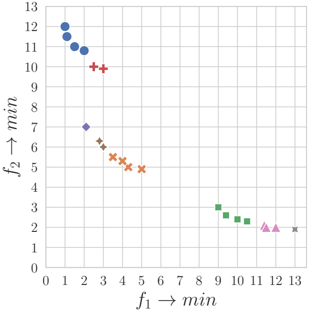
Goal: the size of each cluster is less than or equal to 5

Original clustering



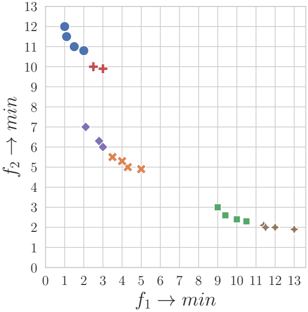
3 clusters

Refine large clusters



8 clusters

Merge small clusters



6 clusters

Original clustering $\longrightarrow$ **Refine large clusters** $\longrightarrow$ **Merge small clusters**

Given:

- set S to be clustered
- maximum cluster size τ

$\Longrightarrow$ Divide S into $n = \left\lceil \frac{|S|}{\tau} \right\rceil$ clusters by using an existing clustering method[2]

[2] Fabian Pedregosa et al. "Scikit-Learn: Machine Learning in Python". In: *Journal of Machine Learning Research* (Jan. 2012). arXiv: `1201.0490v4`.

Original clustering $\longrightarrow$ **Refine large clusters** $\longrightarrow$ **Merge small clusters**

Given:

– set S to be clustered

– maximum cluster size τ

$\Longrightarrow$ Divide S into $n = \left\lceil \frac{|S|}{\tau} \right\rceil$ clusters by using an existing clustering method[2]

- K-Means clustering
- Agglomerative clustering
- Spectral clustering

[2] Pedregosa et al., "Scikit-Learn: Machine Learning in Python".

$$\textbf{Original clustering} \xrightarrow{\text{Clusters}} \textbf{Refine large clusters} \longrightarrow \textbf{Merge small clusters}$$

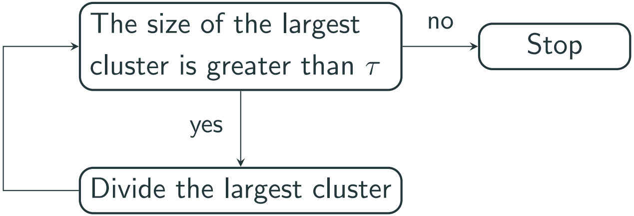
Given:

- clusters
- maximum cluster size τ

Original clustering $\xrightarrow{\text{Clusters}}$ **Refine large clusters** $\longrightarrow$ **Merge small clusters**

Given:

- clusters
- maximum cluster size τ



Original clustering $\xrightarrow{\text{Clusters}}$ **Refine large clusters** $\xrightarrow{\text{Clusters}}$ **Merge small clusters**

Given:

- Clusters
- maximum cluster size τ
- maximum distance between clusters *dist*

Original clustering $\xrightarrow{\text{Clusters}}$ **Refine large clusters** $\xrightarrow{\text{Clusters}}$ **Merge small clusters**

Given:

- Clusters
- maximum cluster size τ
- maximum distance between clusters *dist*

$\Longrightarrow$ Merge two clusters if

- the distance between them is less than *dist*
- the size of the merged cluster is less than or equal to τ

Function inlining decreases WCET and energy consumption but increases code size.

```
int max (int i, int j)
{
 return i>j?i:j;
}

int main()
{
 ...
 a = max(c,d);
 ...
 b = max(f,g);
}
```

```
int main()
{
 ...
 a = c>d?c:d;
 ...
 b = f>g?f:g;
}
```

Function inlining decreases WCET and energy consumption but increases code size.

```c
int max (int i, int j)
{
 return i>j?i:j;
}

int main()
{
 ...
 a = max(c,d);
 ...
 b = max(f,g);
}
```

```c
int main()
{
 ...
 a = c>d?c:d;
 ...
 b = f>g?f:g;
}
```

(WCET, code size, energy consumption) $\longrightarrow$ min

Function inlining decreases WCET and energy consumption but increases code size.

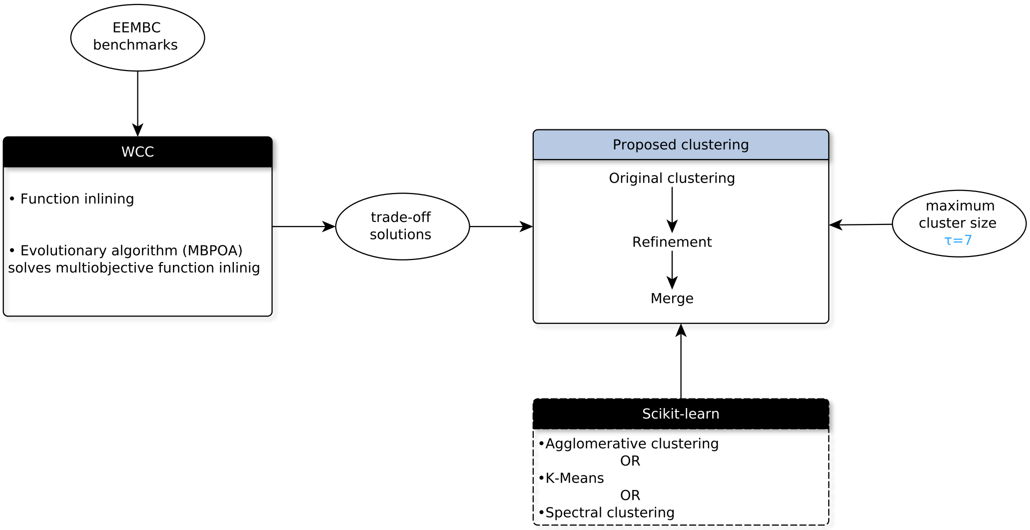
```c
int max (int i, int j)
{
 return i>j?i:j;
}

int main()
{
 ...
 a = max(c,d);
 ...
 b = max(f,g);
}
```

```c
int main()
{
 ...
 a = c>d?c:d;
 ...
 b = f>g?f:g;
}
```

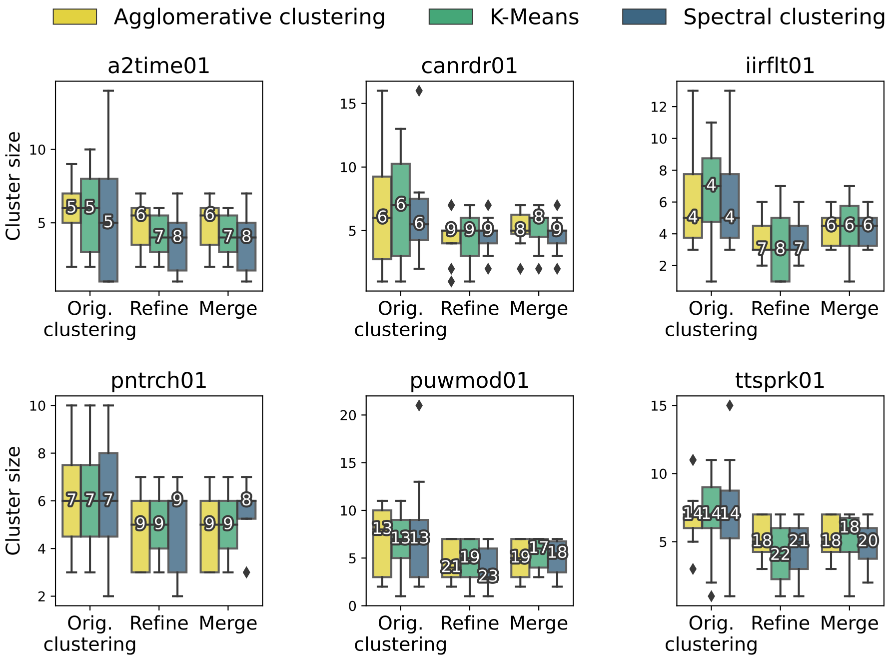
(WCET, code size, energy consumption) $\longrightarrow$ min

WCET-Aware Compiler Framework **WCC**

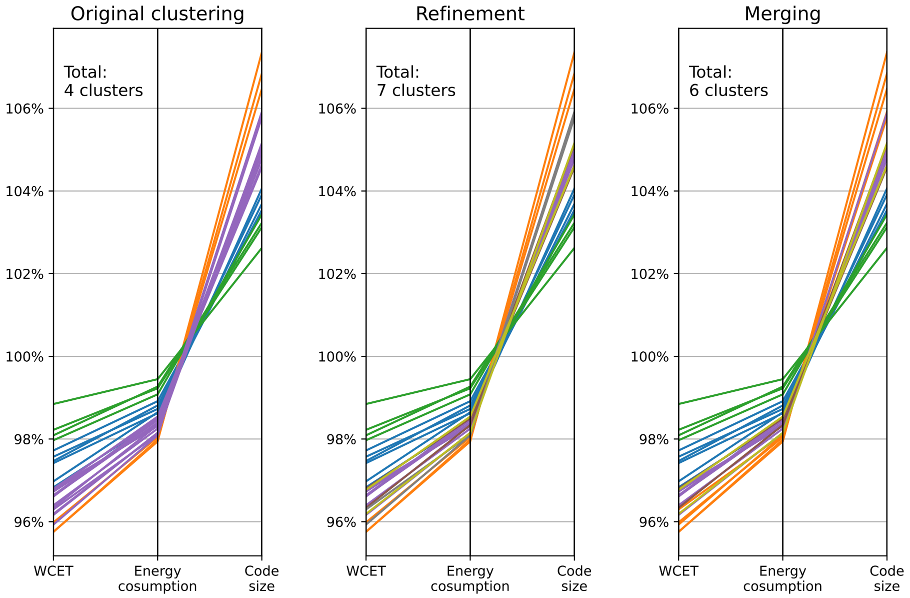
Setup


EEMBC benchmarks
WCC
• Function inlining
• Evolutionary algorithm (MBPOA) solves multiobjective function inlinig
trade-off solutions
Proposed clustering
Original clustering
Refinement
Merge
maximum cluster size
τ=7
Scikit-learn
•Agglomerative clustering
OR
•K-Means
OR
•Spectral clustering

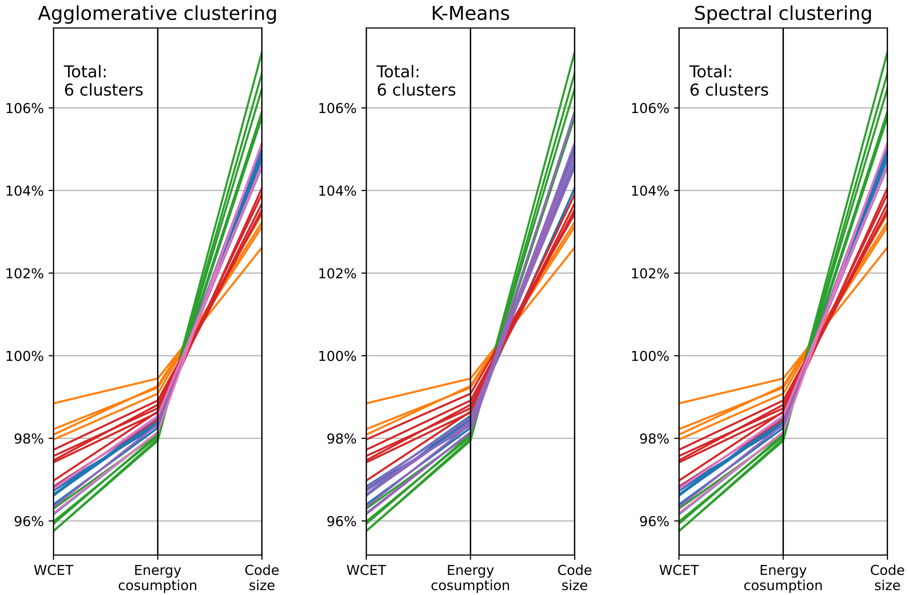
Cluster sizes after each stage of the proposed approach



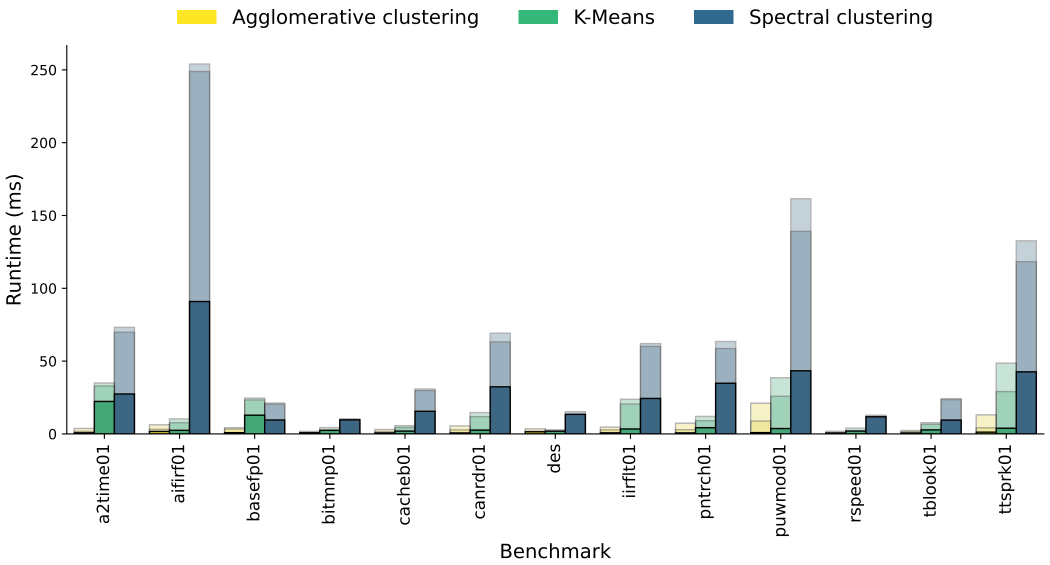
Clusters for benchmark iirflt01 and spectral clustering



Final clusters for benchmark iirflt01



Runtime


Agglomerative clustering
K-Means
Spectral clustering
Runtime (ms)
0
50
100
150
200
250
a2time01
aifirf01
basefp01
bitmnp01
cacheb01
canrdr01
des
iirflt01
pntrch01
puwmod01
rspeed01
tblook01
ttsprk01
Benchmark

Conclusion

- The proposed clustering method guarantees that the sizes of all clusters are less than a predefined limit.
- We demonstrated the approach on multiobjective function inlining with WCET, code size and energy consumption as objectives.
- K-Means, agglomerative and spectral clusterings showed similar results in terms of the number of clusters and their sizes, but agglomerative clustering showed the smallest runtime.

Maximum distance between two clusters

$$dist = \frac{d_{max}}{n-1} \tag{1}$$

n is the number of clusters in the input set S and d_{max} is the maximum distance between two points from the set S:

$$d_{max} = \max_{p,q \in S} ||p - q|| \tag{2}$$