



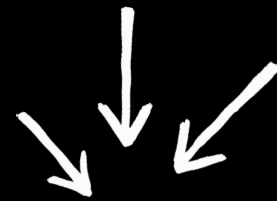
Selection of NTMP

A clear step-by-step process for
selecting the best non traditional
machining process

FACILITATED BY
KAILASH NATH. M
31121114061

The Goal

To come up with the best possible
NTMP for the given input criteria



A Genuine ranking system

The Research

Non traditional machining processes(NTMP) play a vital role in many industrial sectors like aerospace, medical, warfare, etc. Thus, the selection of the best NTMP is necessary to make best use of the process in the long run.

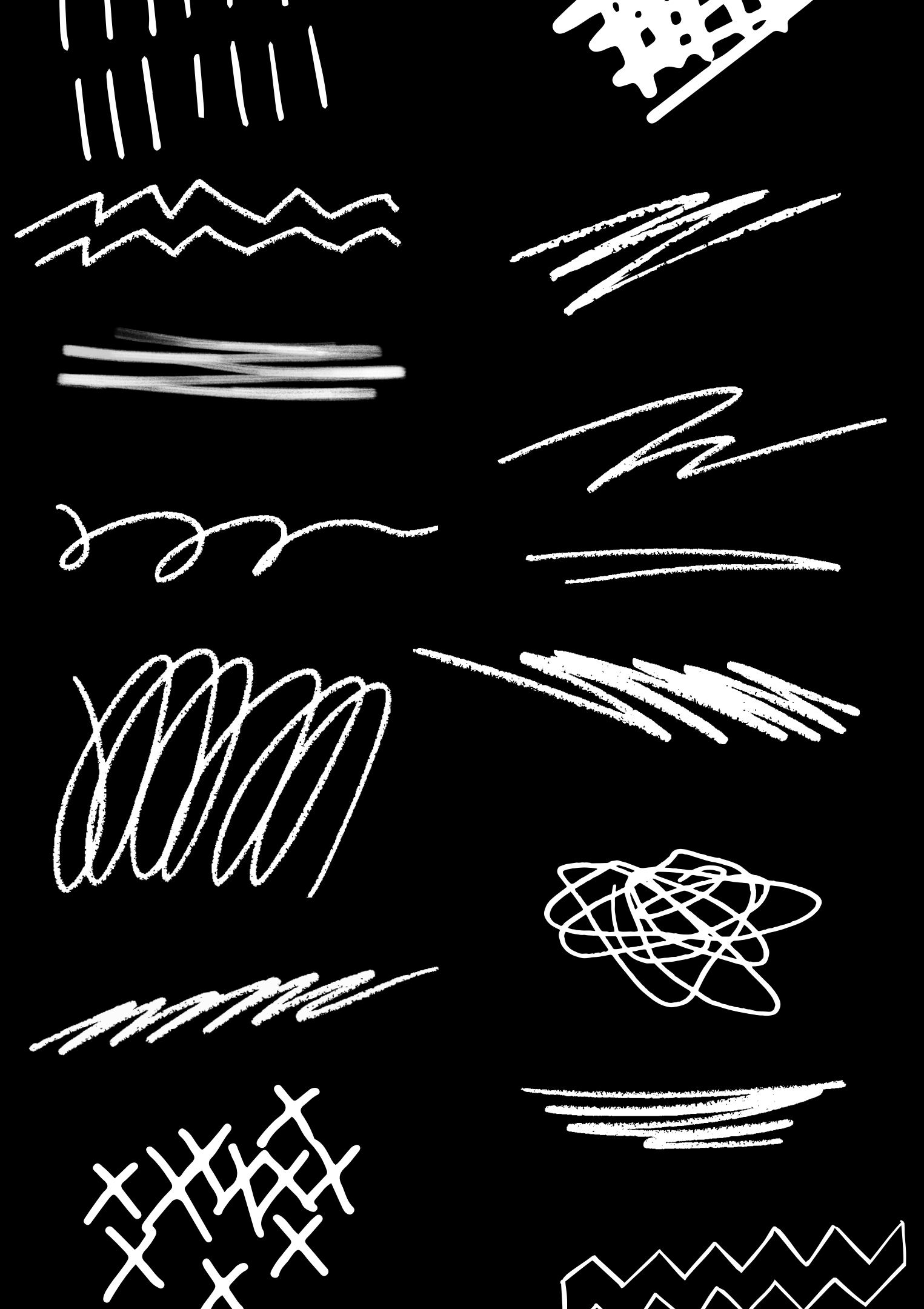


The Problem

Different people have different opinions. Processes quoted to be fair, may or may not be so. One single process can never satisfy all the expectations. However, ideally we could consider one which quenches the thirst of most of the parameters.

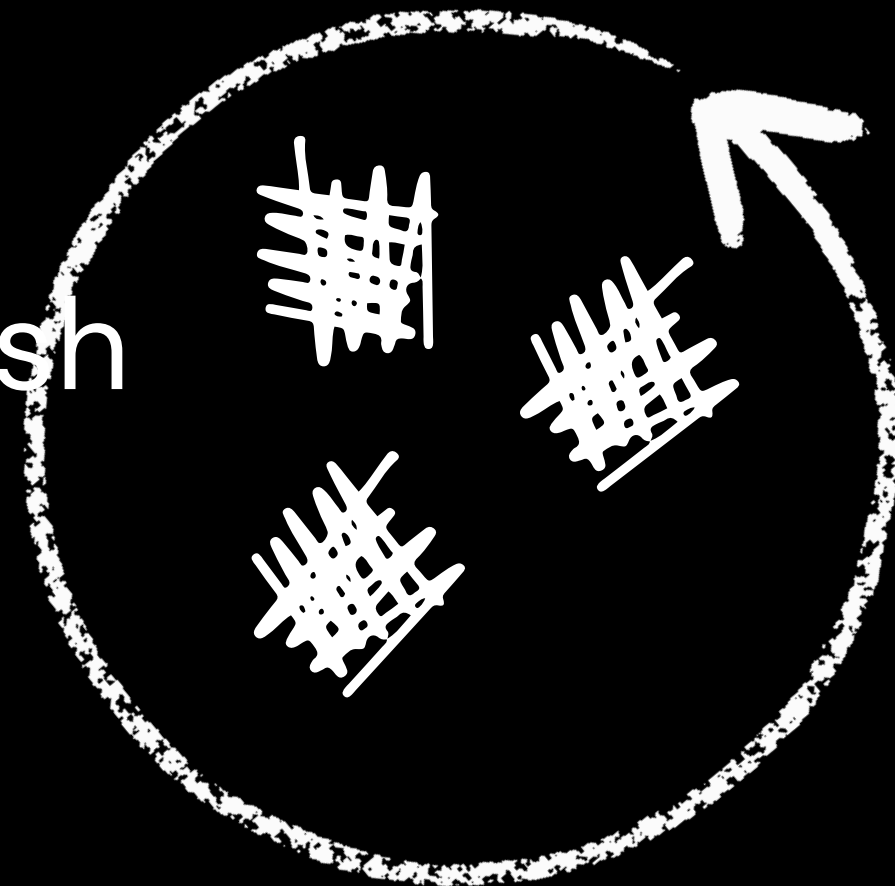
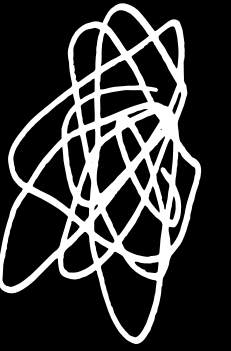
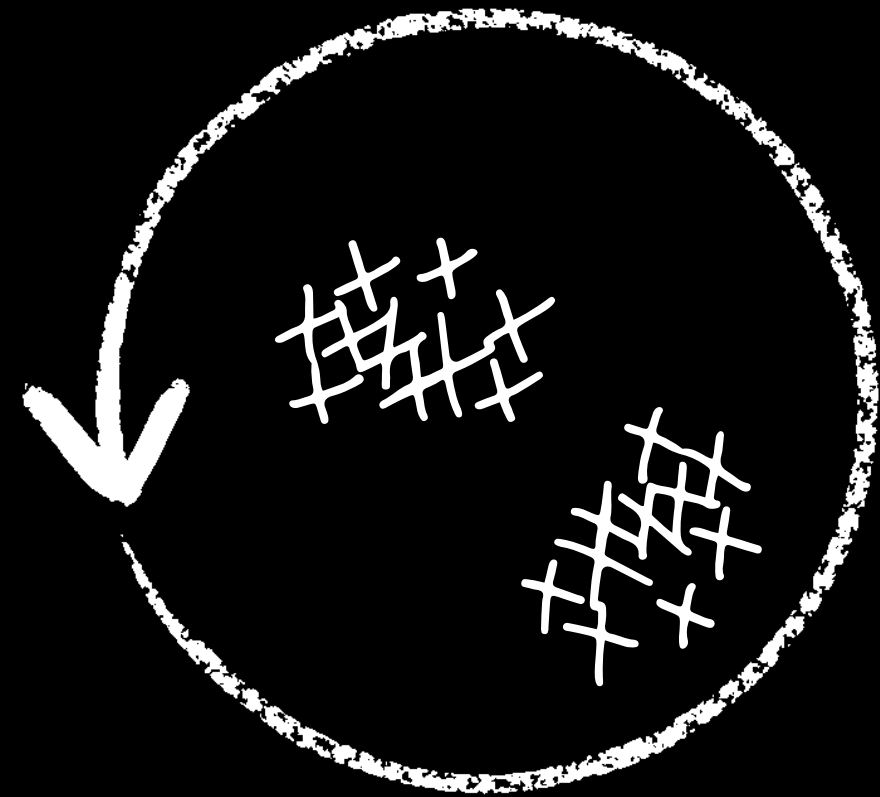
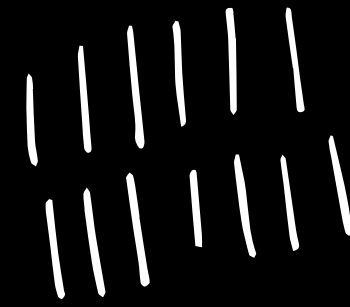
Solution

To generate an automated system which predicts the best suitable process in less time, thus increasing the productivity.



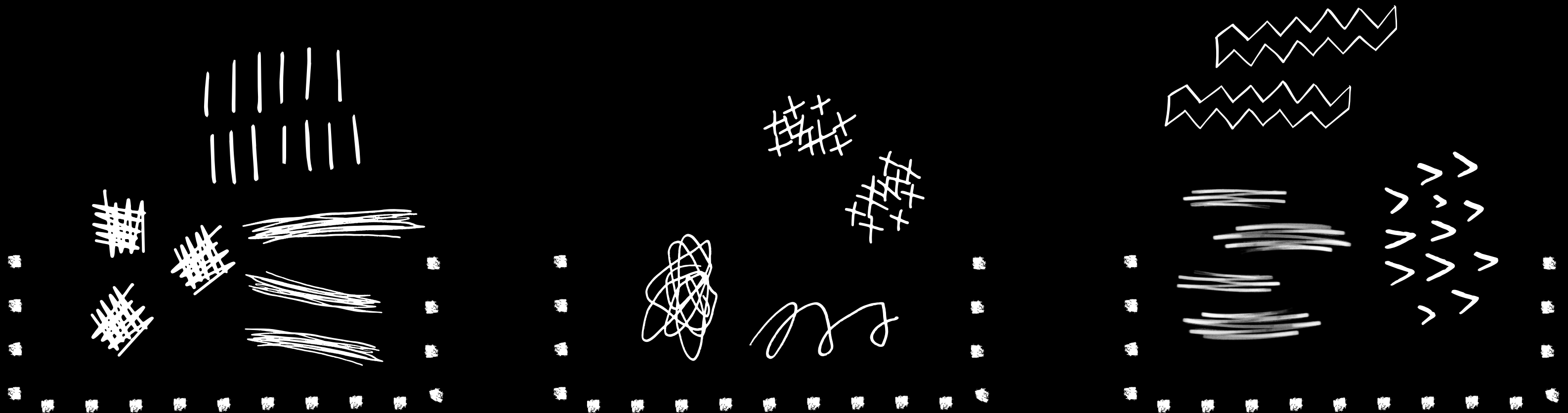
What to consider?

- Work material
- MRR
- Shape feature
- Tolerance and Surface finish
- Power requirement
- Cost



Principle

- Multi Criteria Decision Making (MCDM)
- Multi Attribute Decision Making (MADM)



FUZZY AXIOMATIC DESIGN (FAD)

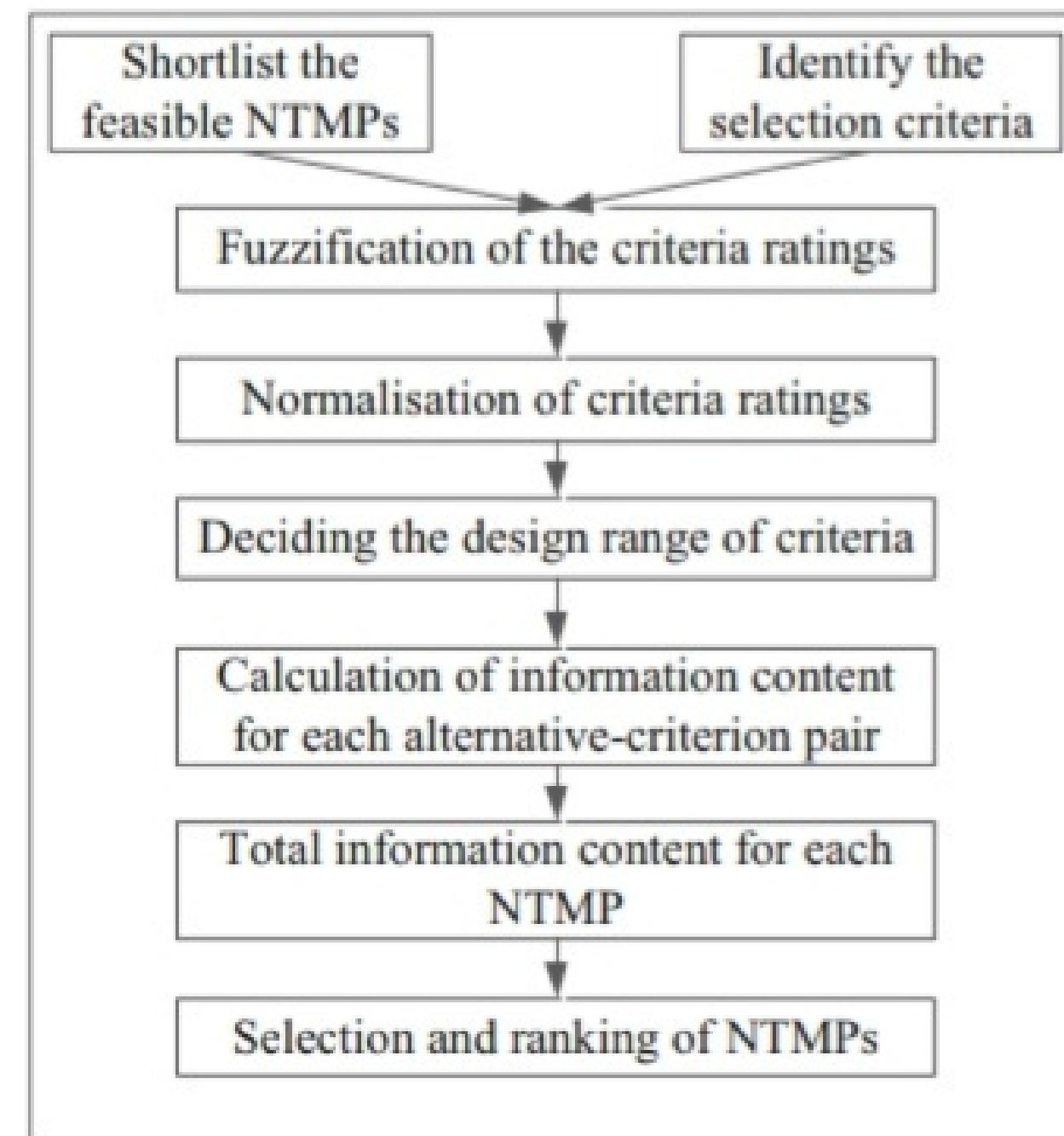
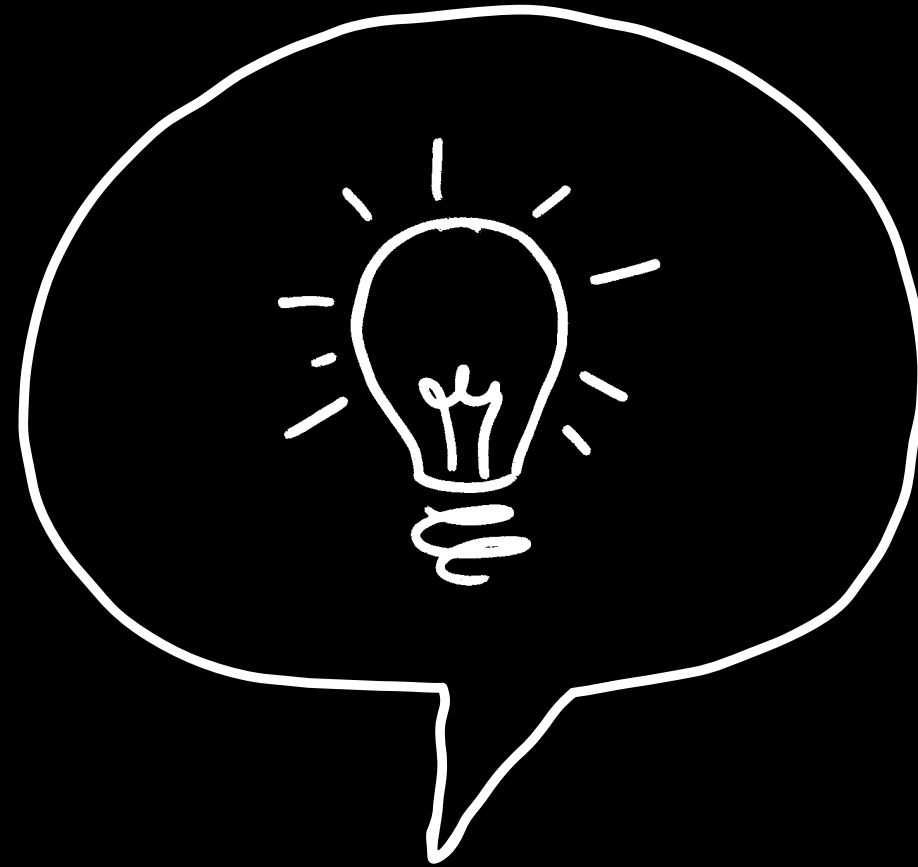


Fig. 4 Flowchart for NTMP selection based on FAD methodology



Time to code

Scoring criteria

MRR += 2

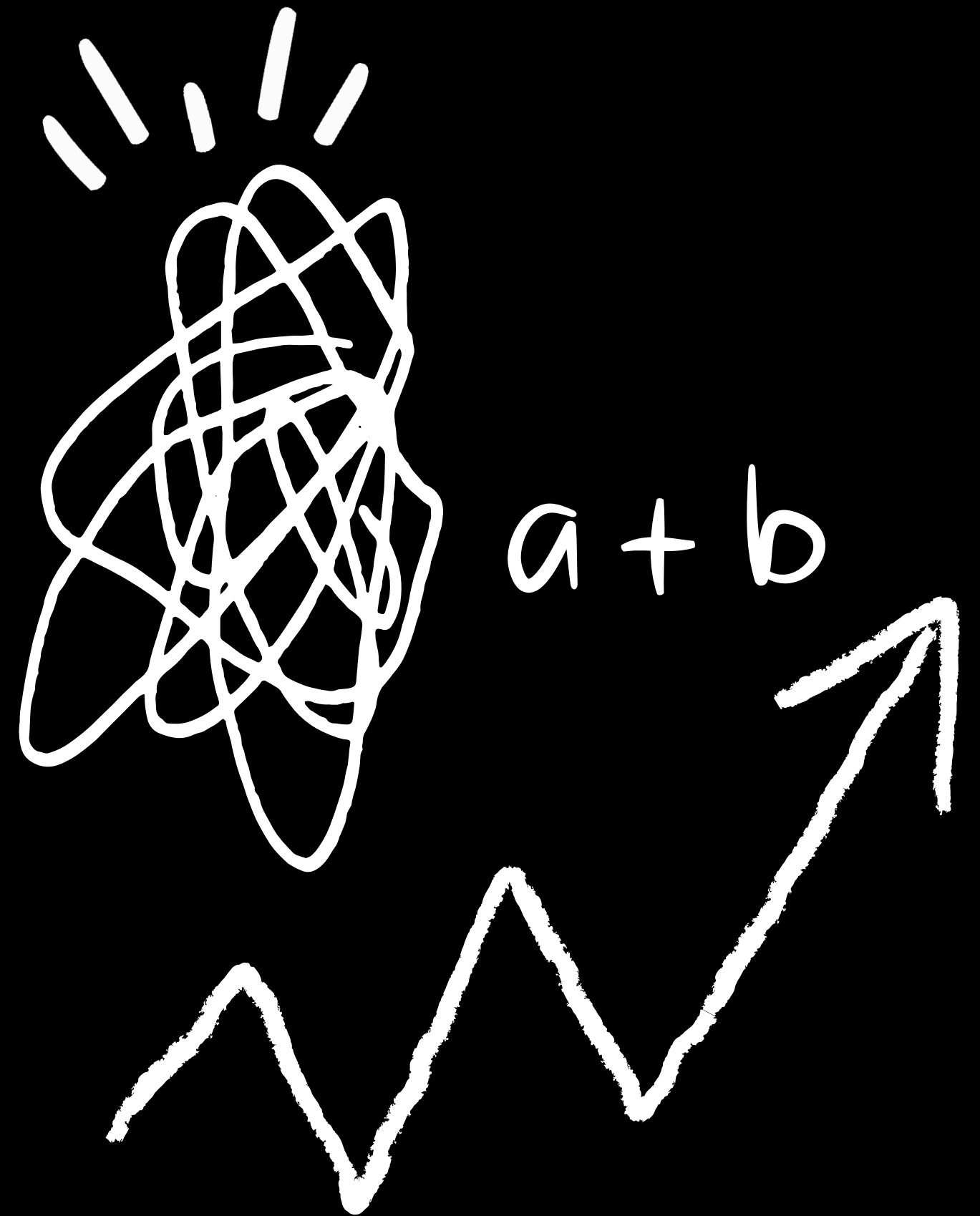
Shape feature += 2

Work material += 2

Tolerance += 1

Power requirement += 1

Cost += 1



The code

```
def select_best_ncm_process(mrr, shape_feature, work_material, tolerance_sf, power_req, cost):
    ncm_processes = [
        {"name": "AJM", "mrr": "High", "shape_feature": "Limited", "work_material": "Various", "tolerance_sf": "Moderate", "power_req":
            "Moderate", "cost": "Moderate"},
        {"name": "USM", "mrr": "Low", "shape_feature": "Intricate", "work_material": "Brittle", "tolerance_sf": "High", "power_req": "Low",
            "cost": "Low"},
        {"name": "CHM", "mrr": "Moderate", "shape_feature": "Complex", "work_material": "Various", "tolerance_sf": "High", "power_req":
            "Low", "cost": "Low"},
        {"name": "EBM", "mrr": "High", "shape_feature": "Limited", "work_material": "Various", "tolerance_sf": "Low", "power_req": "High",
            "cost": "High"},
        {"name": "LBM", "mrr": "Moderate", "shape_feature": "Complex", "work_material": "Various", "tolerance_sf": "Moderate",
            "power_req": "Moderate", "cost": "High"},
        {"name": "ECM", "mrr": "High", "shape_feature": "Intricate", "work_material": "Conductive", "tolerance_sf": "High", "power_req":
            "Moderate", "cost": "High"},
        {"name": "EDM", "mrr": "Moderate", "shape_feature": "Complex", "work_material": "Conductive", "tolerance_sf": "Low", "power_req":
            "High", "cost": "Moderate"},
        {"name": "PAM", "mrr": "High", "shape_feature": "Limited", "work_material": "Conductive", "tolerance_sf": "High", "power_req":
            "High", "cost": "High"}
    ]
```

```
# Define a scoring function based on input parameters
```

```
def score_process(process):
```

```
    score = 0
```

```
    if process["mrr"] == mrr:
```

```
        score += 2
```

```
    if process["shape_feature"] == shape_feature:
```

```
        score += 2
```

```
    if process["work_material"] == work_material:
```

```
        score += 2
```

```
    if process["tolerance_sf"] == tolerance_sf:
```

```
        score += 1
```

```
    if process["power_req"] == power_req:
```

```
        score += 1
```

```
    if process["cost"] == cost:
```

```
        score += 1
```

```
    return score
```

```
# Find the process with the highest score
```

```
best_process = max(ncm_processes, key=score_process)
```

```
return best_process["name"]
```

```
# Example usage:
```

```
mrr = "High"
```

```
shape_feature = "Limited"
```

```
work_material = "Various"
```

```
tolerance_sf = "Moderate"
```

```
power_req = "Moderate"
```

```
cost = "Moderate"
```

```
best_ncm_process = select_best_ncm_process(mrr, shape_feature, work_material, tolerance_sf, power_req, cost)
```

```
print("Best non-traditional machining process:", best_ncm_process)/
```



References

Selection of non-conventional machining process using CRITIC-CODAS method

Anita Kumari, Bappa Acherjee

<https://doi.org/10.1016/j.matpr.2021.12.152>

Selection of Non-Traditional Machining Process

DOI : 10.17577/IJERTV8IS110052

Authors : Rohith R , Shreyas B K , S Kartikgeyan, Sachin B A , Umesha K R, Dr. T S

Nanjundeswaraswamy.



Thanks for your time!

