Efficient Resource Provisioning for Smart Buildings Utilizing Fog and Cloud Based Environment

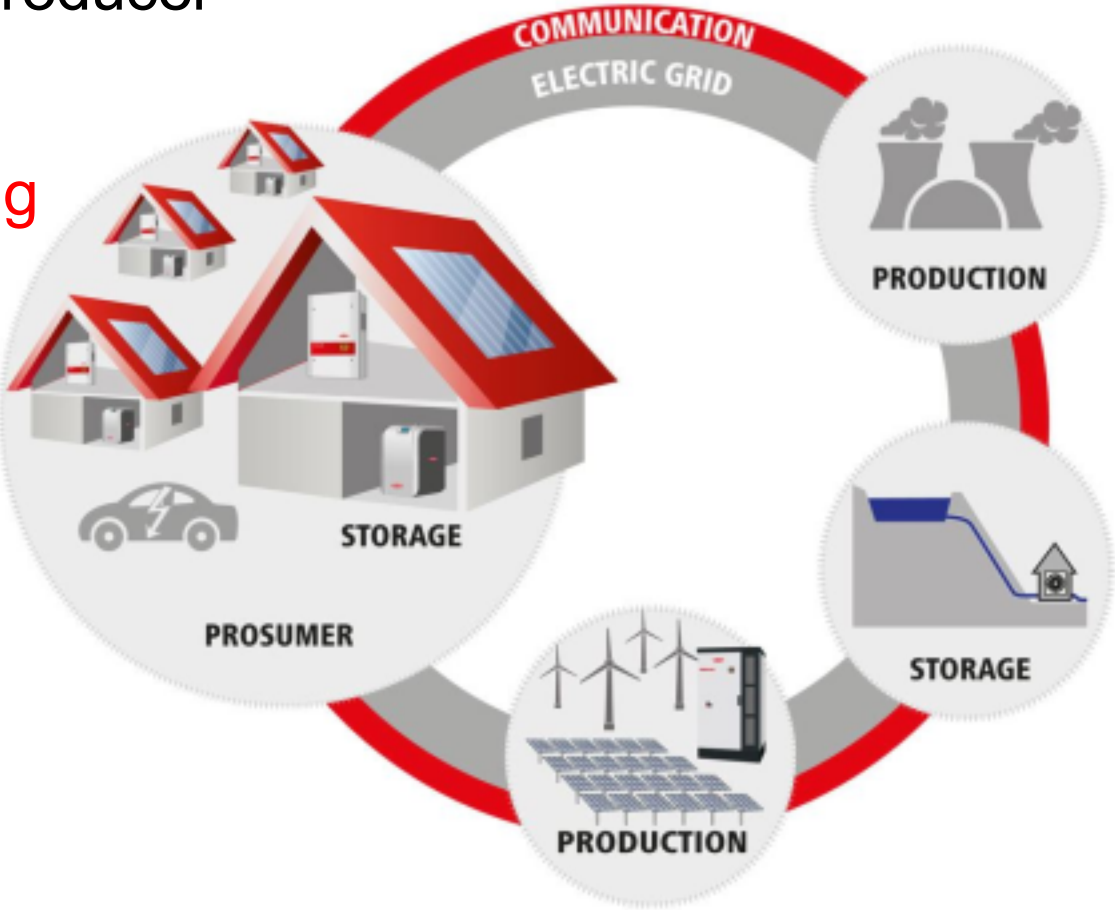
Anila Yasmeen, Nadeem Javaid, Obaid Ur Rehman, Hina Iftikhar, Muhammad Faizan Malik, and Fatima J. Muhammad

2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC)

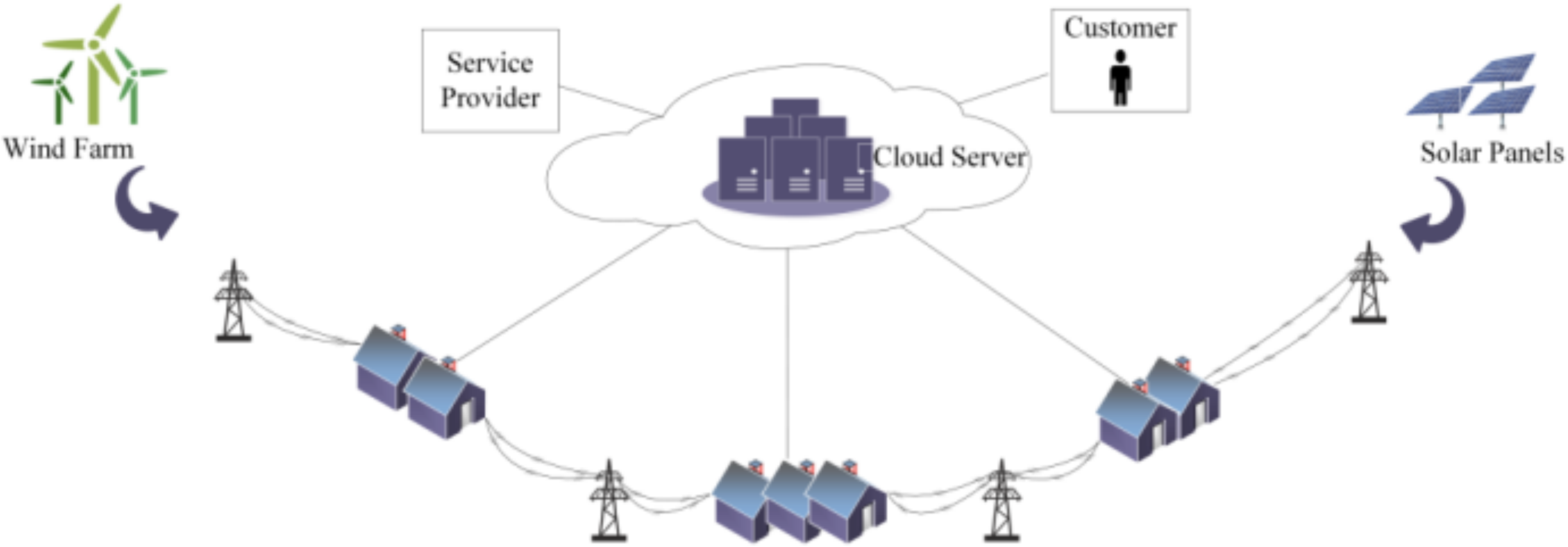
Smart Grid

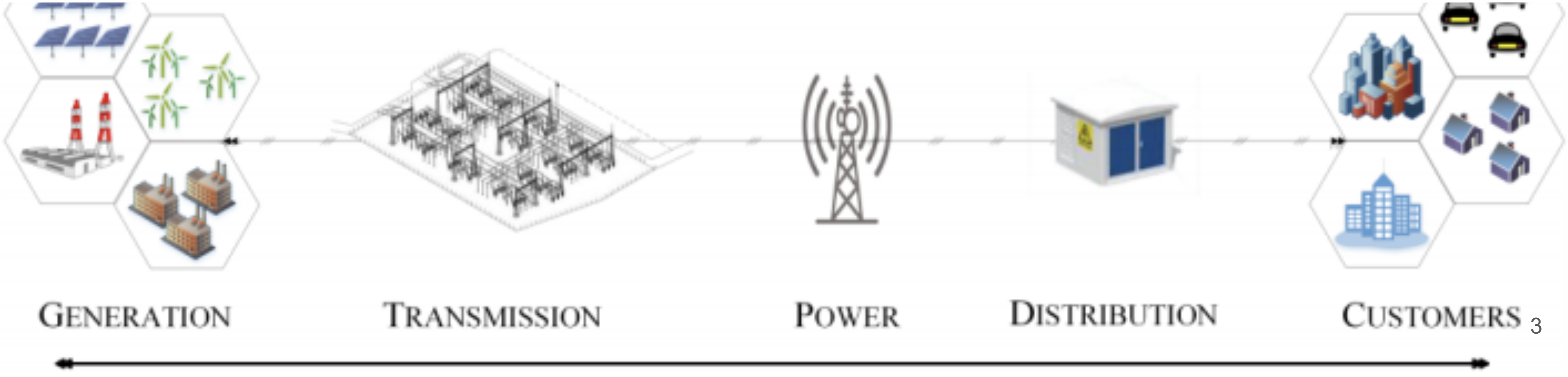
IoT is used for collecting information from consumer and producer

- Energy Saving
- Sustainability
- Reliability



Integration of Smart Grid with Cloud Computing



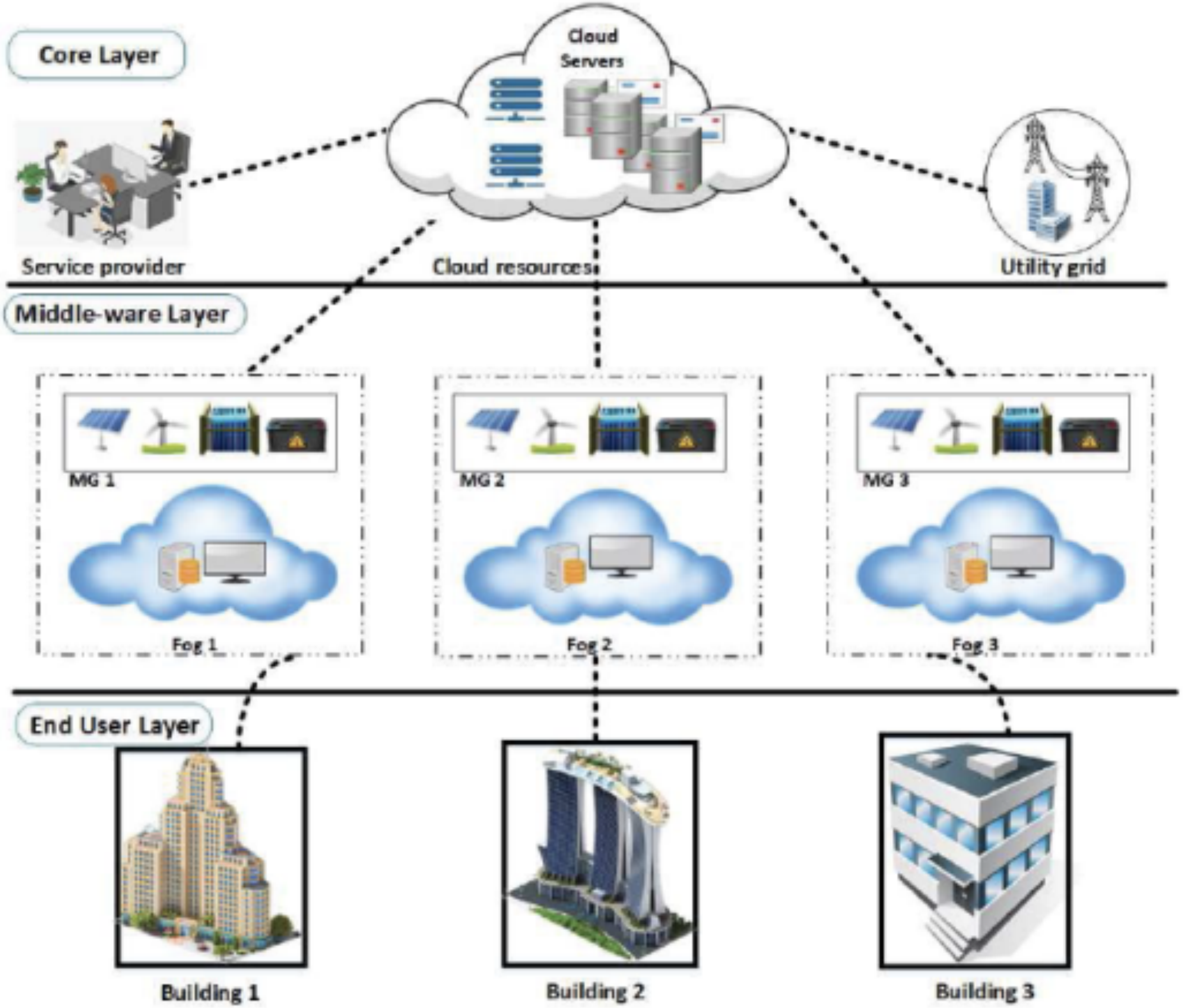


Contribution

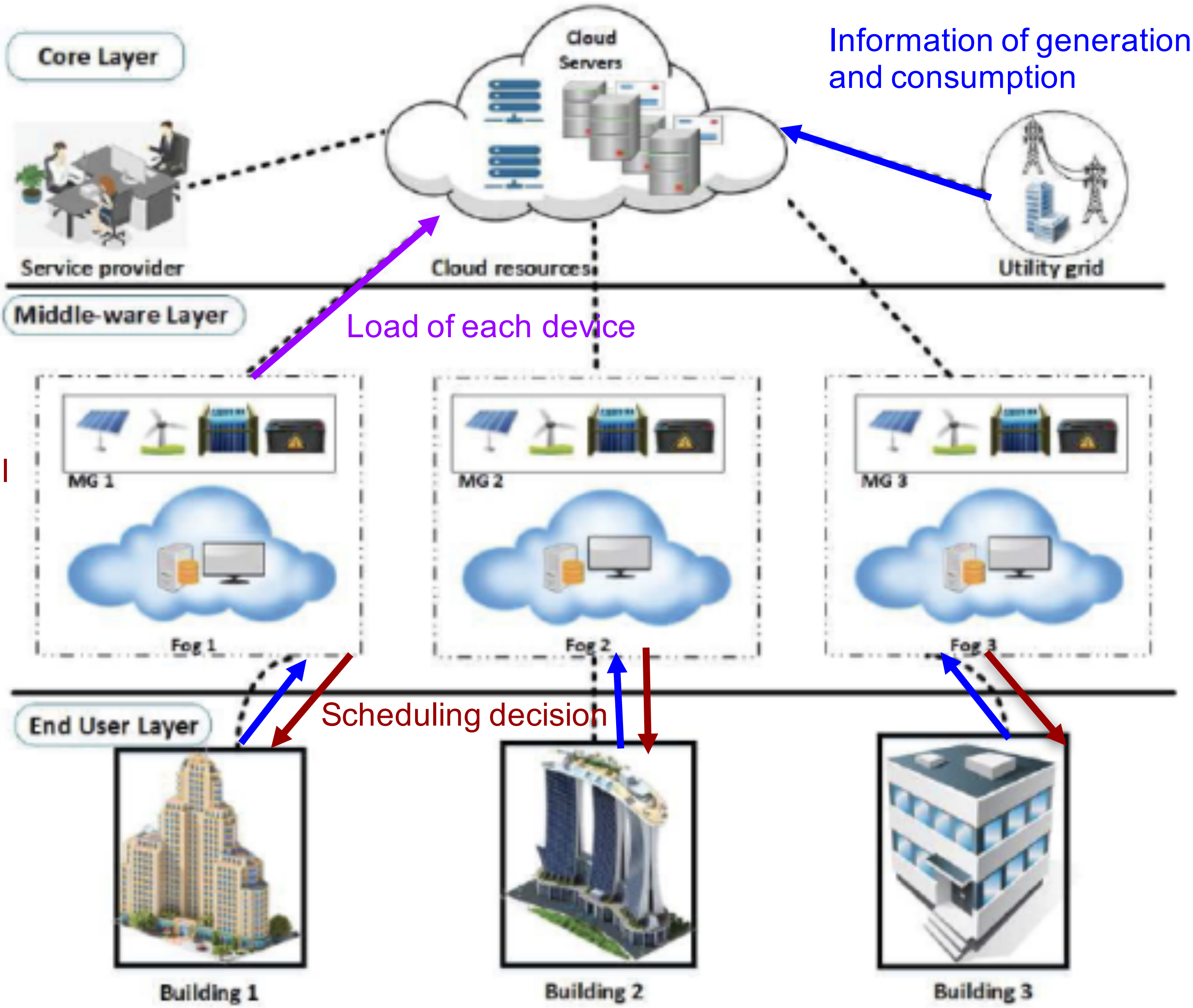
- A three layered model based on cloud and fog framework is proposed. These layers consist of core cloud layer, fog layer and end user layer.
- A new Hybrid Particle Swarm Optimization with Simulated Annealing (PSO-SA) is proposed to find an optimum allocation of tasks to the available VMs.

System Model

Power generation unit
Energy storage unit



System Workflow



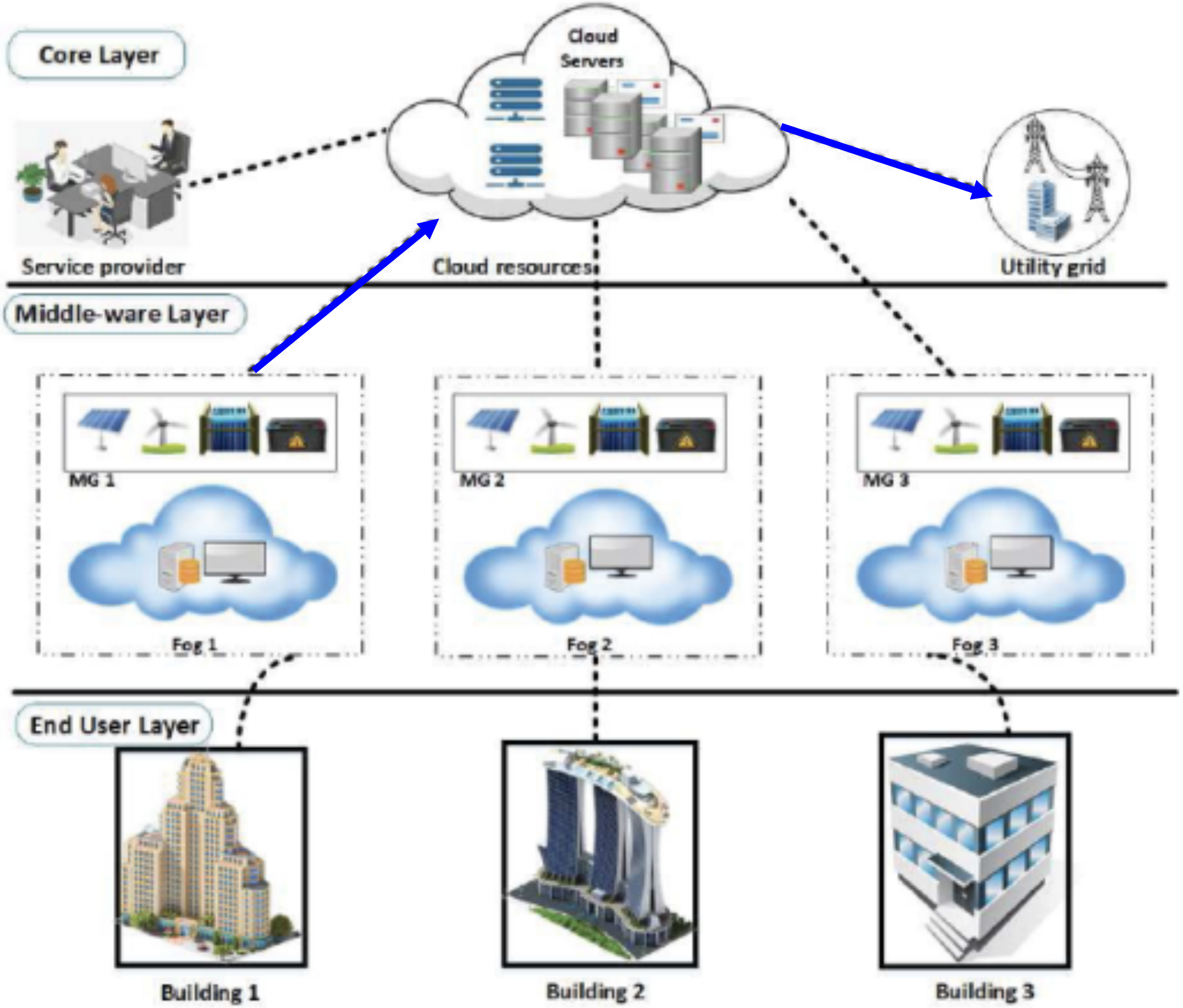
Information of generation and consumption

Load of each device

Calculate the optimal scheduling of power consumption

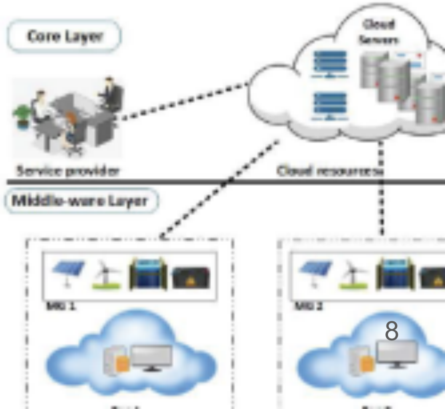
Scheduling decision

Lack of Power Generation From MG

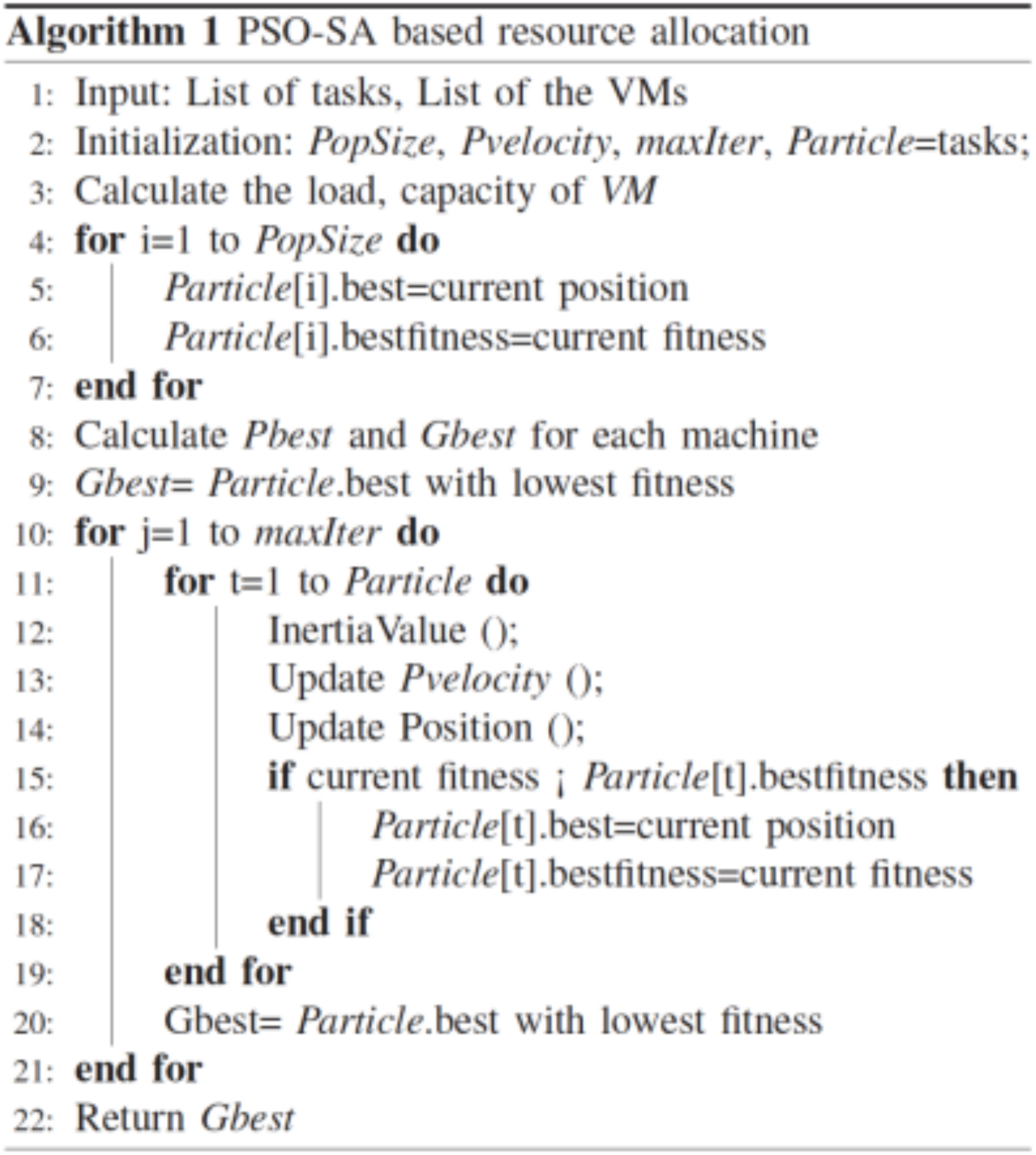


Problem:

- Goal: Minimizing the response time, cost
- VMs are allocated on basis of storage and memory requirement to host for balancing on fog network
- Servers are responsible to manage all hosts according to policy and may be more than one VMs are assigned to one host.
- Load balancing is the distribution of workload for multiple links to avoid overloading and to achieve minimum response time.



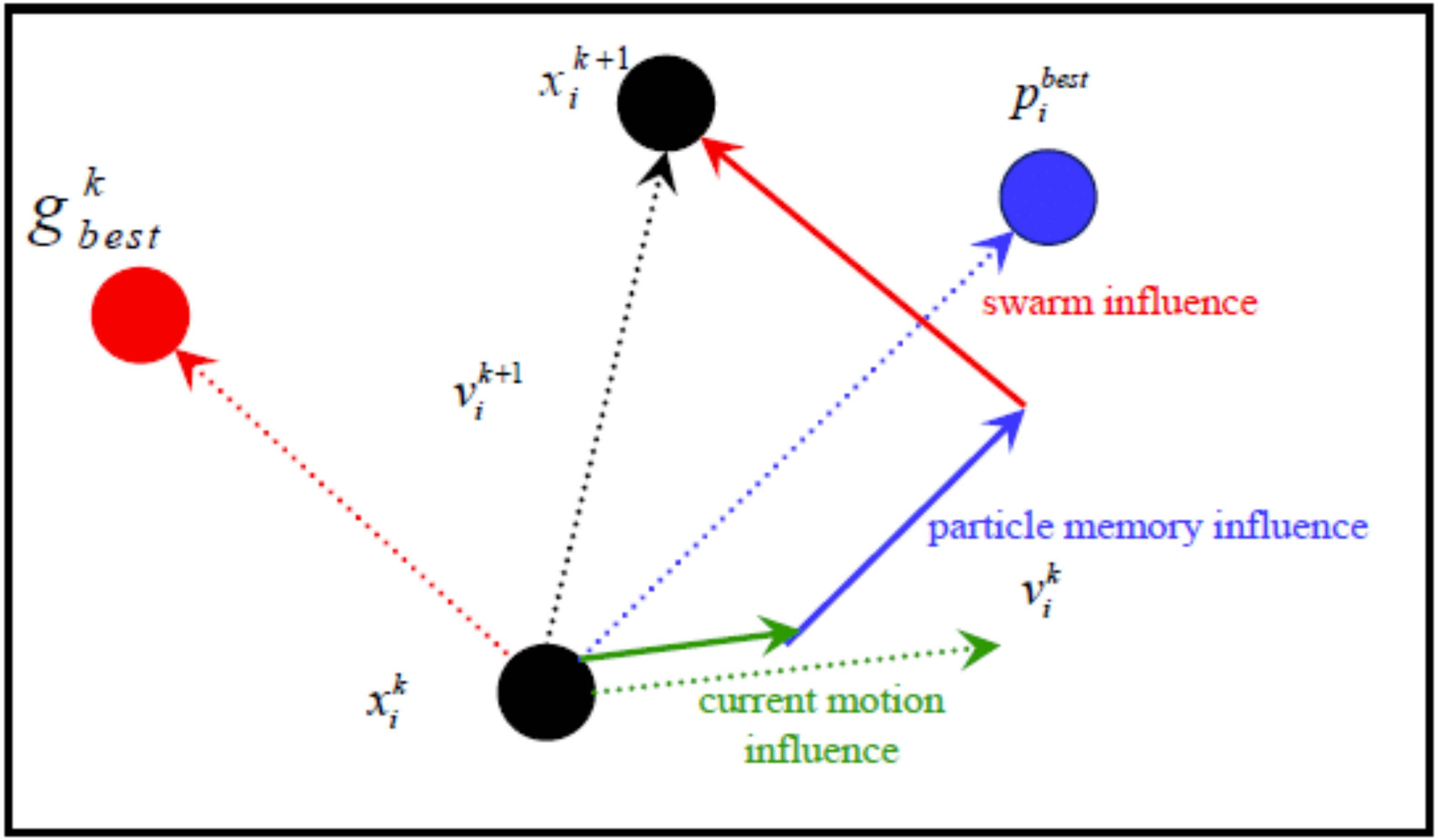
Loading Balancing Algorithm - PSO-SA



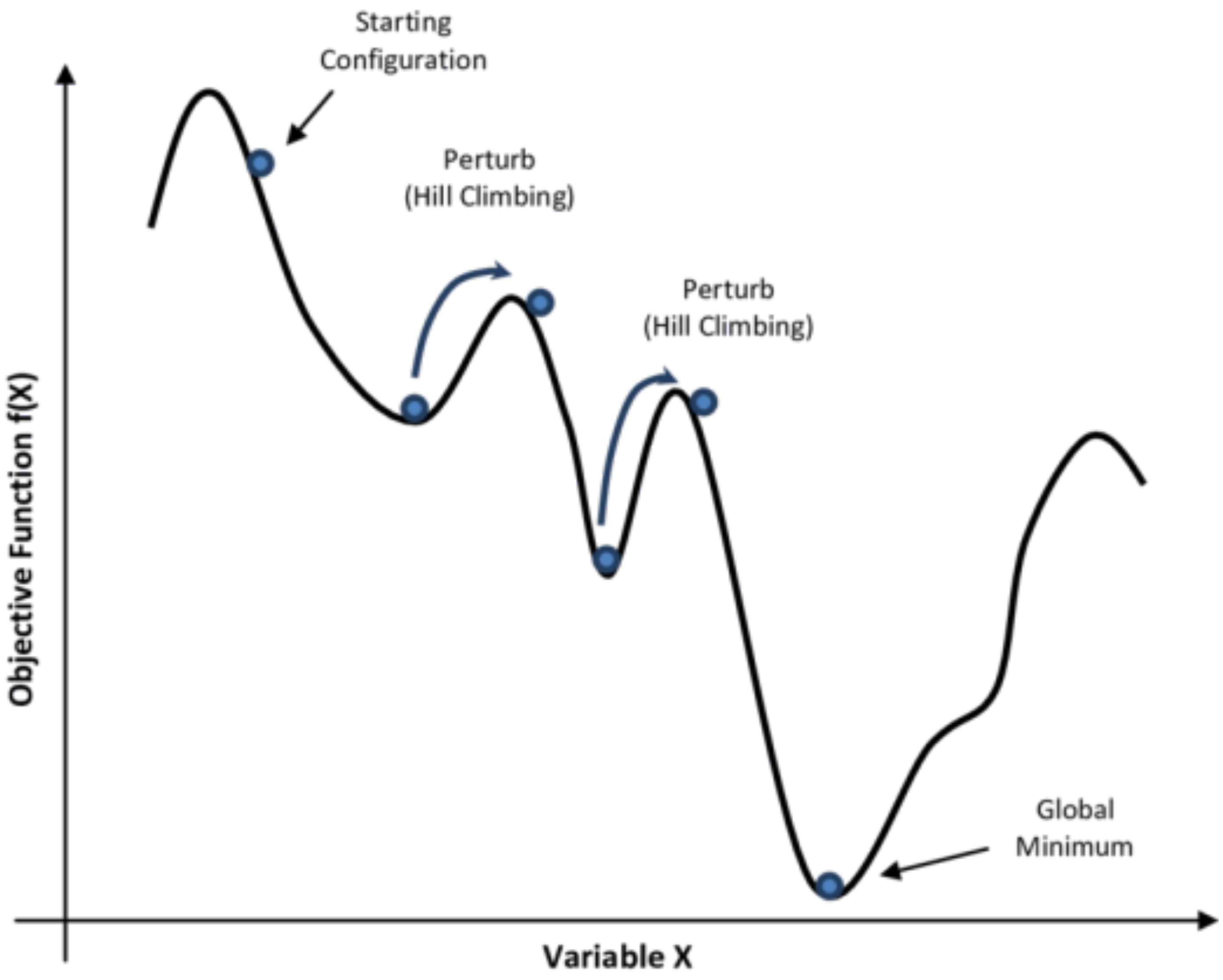
Algorithm 1 PSO-SA based resource allocation

```
Input: List of tasks, List of the VMs
Initialization: PopSize, Pvelocity, maxIter, Particle=tasks;
Calculate the load, capacity of VM
for i=1 to PopSize do
    Particle[i].best=current position
    Particle[i].bestfitness=current fitness
end for
Calculate Pbest and Gbest for each machine
Gbest= Particle.best with lowest fitness
for j=1 to maxIter do
    for t=1 to Particle do
        InertiaValue ();
        Update Pvelocity ();
        Update Position ();
        if current fitness ¡ Particle[t].bestfitness then
            Particle[t].best=current position
            Particle[t].bestfitness=current fitness
        end if
    end for
    Gbest= Particle.best with lowest fitness
end for
Return Gbest
```

Particle Swarm Optimization



Simulated Annealing



Simulation Setup

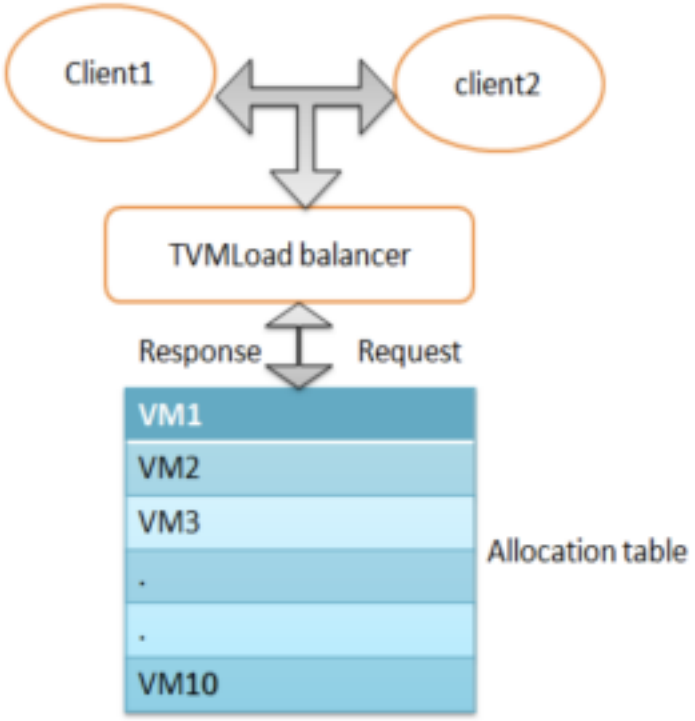
- Distributed fog framework and centralized cloud
- Exist 3 region. Each region includes 100 house
- Run 24 hours
- Loading balancing algorithm: PSO-SA, RR, and throttled

Metric

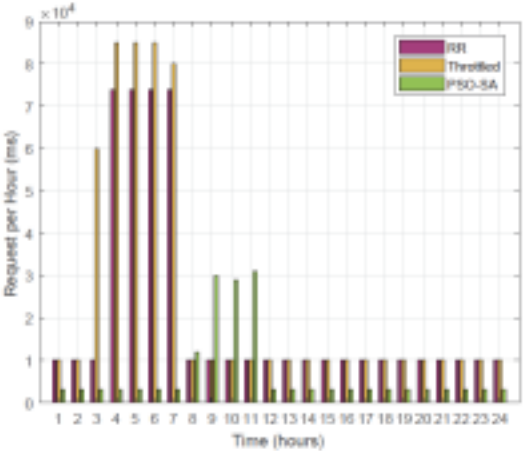
- Response time
- Process time
- Cost

Baseline Algorithm

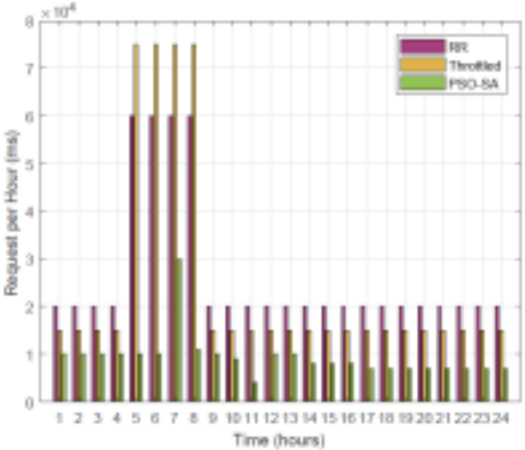
- RR: Allocate resources to each host by equal time slicing for utilization of resources.
- Throttled: Allocate the first VM that is available in the table.



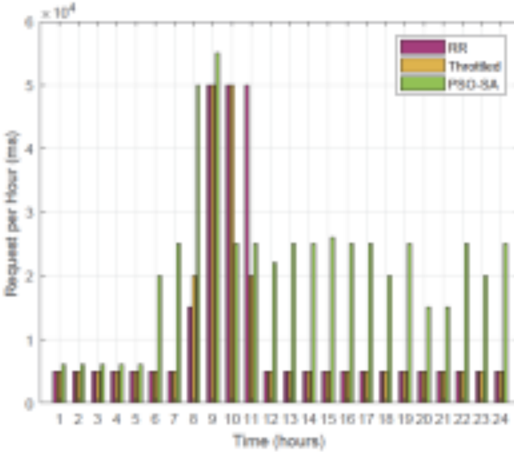
Request per hour of the algorithms



(a) Fog 1

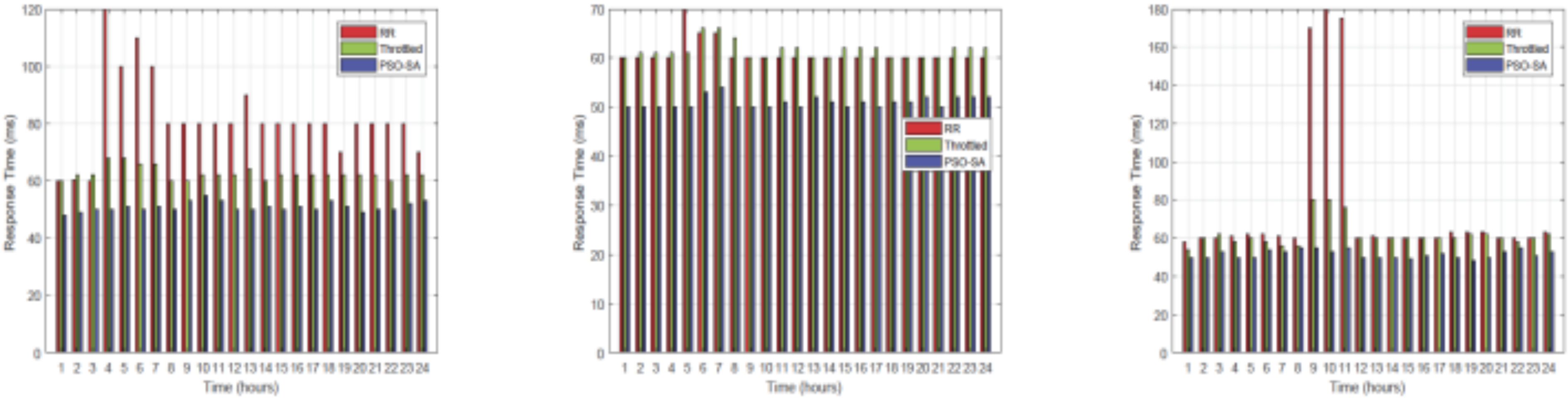


(b) Fog 2



(c) Fog 3

Response time of the algorithms

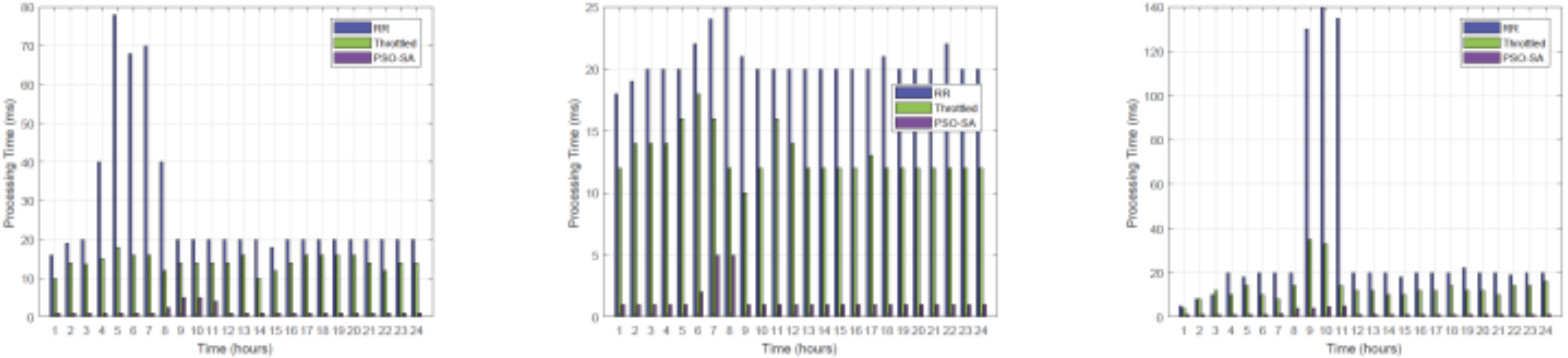


(a) Building 1

(b) Building 2

(c) Building 3

Processing time of the algorithm

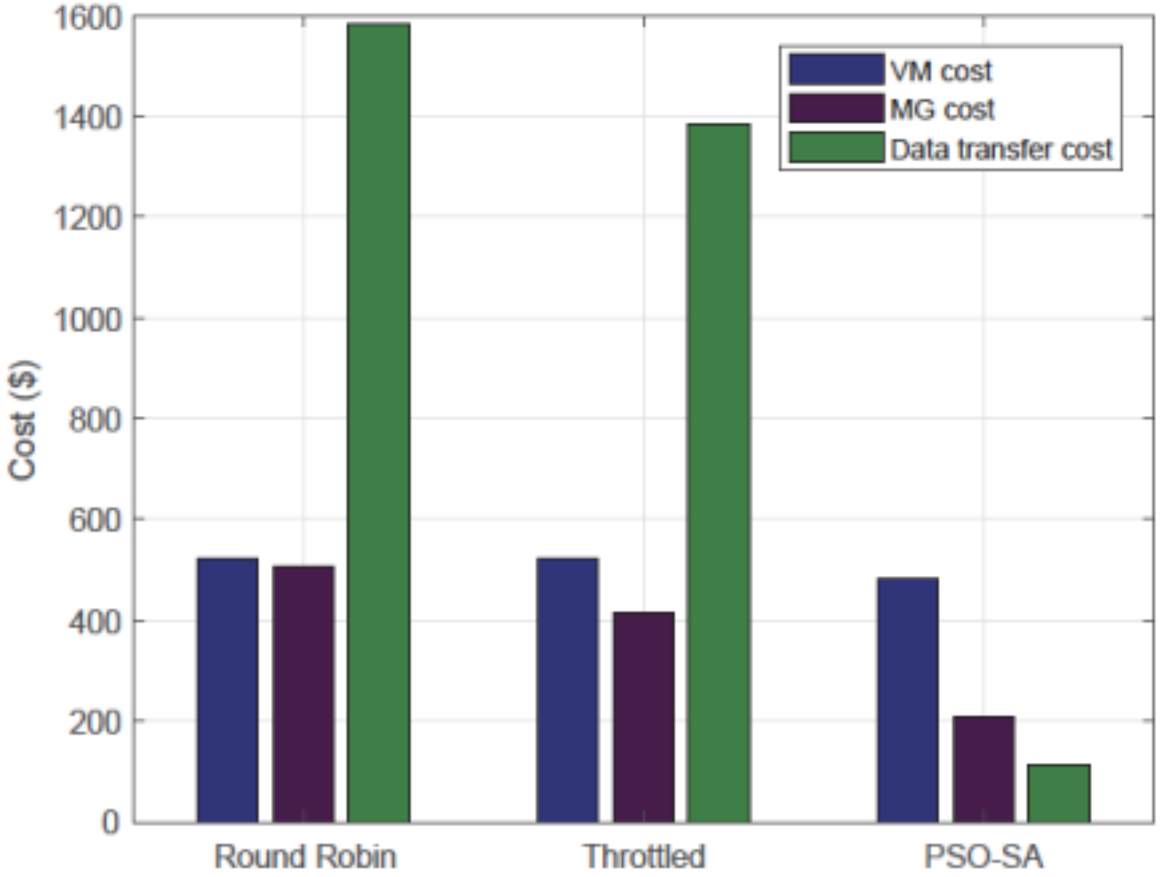


(a) Fog 1

(b) Fog 2

(c) Fog 3

Overall cost of the algorithms



Conclusion