```
void evaluate(float*capacity,
                    float*bandwith,
                    float*weight,
                    int chromsome[][],
                    int paths[][]
                    int &objective[]){

__shared__ float flow[E]={};
__shared__ float shared[E]={};
for(i=threadid;i<|D|;i+=blockDim){
        int p[]=paths[i][chromsome[blockid][i]];
        for(j=0,j<length_of(p);j++)
                {
                int band=bandwidth[blockid];
                atomicAdd(&flow[p[j]],band);
                }
}
__syncthreads();
for(i=threadid;i<E;i+=blockDim)
{
        if(flow[i]<capacity[i])
                shared[i]=flow[i]*weight[i];
        else
                shared[i]=INF;
}
__syncthread();
for(int s=E;s>1;s=(s+1)/2)
{       if(e<s/2)
                shared[e]+=shared[e+(s+1)/2];
        __syncthreads();
}
if(threadid==0)
        objective[blockid]=shared[0];
}
```