GRP

AI Assistant

April 29, 2025

1 GRP

(Green Routing Problem, GRP) GRP

2

GRP

1. (x, y, degradation, area)
2. **Graph Attention**
   -
   -
   -
3.
   -
   - (Glimpse + Pointer)
   -
4. **GRP**
   -
   -

3

GRP

4 GRP

GRP

4.1

MLP

```
self.repair_prediction = nn.Sequential(
    nn.Linear(embedding_dim, hidden_dim),
    nn.ReLU(),
    nn.Dropout(0.1),
    nn.Linear(hidden_dim, hidden_dim // 2),
    nn.ReLU(),
    nn.Dropout(0.1),
    nn.Linear(hidden_dim // 2, 1)
)
```

4.2

GRP

- (x, y):
- degradation:
- area:

4.3

GRP

```
step_context_dim = embedding_dim + 1  #    +
```

5

GRP Transformer REINFORCE

Algorithm 1 GRP

Require: θ d_e d_h n_{layers} n_{heads} α N_{epoch} B G_{size} baseline B_{type}

Ensure: θ^*

θ

baseline B_{model} (NoBaseline, ExponentialBaseline, CriticBaseline, RolloutBaseline)

Adam α

LR_Scheduler

val_set (val_size)

for $epoch = 0$ to $N_{epoch} - 1$ **do**

$train\_set$ ($epoch\_size$)

$train\_set$ B

"sampling"

for $batch$ in $train\_set$ **do**

$batch$ x baseline bl_val

(CPU/GPU)

//

$cost, log\_likelihood \leftarrow model(x)$

$reward \leftarrow -cost$

// GRP

if "grp" **then**

$repair\_area$

end if

// baseline

if bl_val None **then**

$bl\_val, bl\_loss \leftarrow baseline.eval(x, cost)$

else

$bl\_loss \leftarrow 0$

end if

// REINFORCE

$reinforce\_loss \leftarrow ((cost - bl\_val) \cdot log\_likelihood).mean()$

$loss \leftarrow reinforce\_loss + bl\_loss$

//

NaN Inf

//

if **then**

end if

end for

//

"greedy"

$val\_dataset$ val_cost

$avg\_reward \leftarrow -val\_cost.mean()$

// Baseline

$baseline.epoch\_callback(model, epoch)$

//

//

if **then**

end if

//

if $epoch == 0$ $avg\_reward > best\_reward$ **then**