

C4I 체계를 활용한 마이크로 위성 운영 알고리즘 개발

국방디지털융합학과 김동진
국방디지털융합학과 박유빈

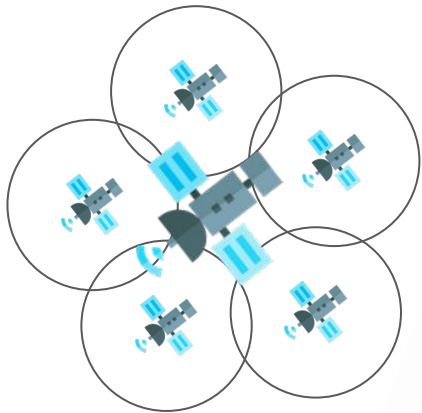
도전과제 목표 (팀 목표)

첫 번째 목표는 여러 대의 마이크로 위성을 활용하여 타겟 설정 알고리즘을 개발하는 것으로, 이는 감시, 정밀 타격, 지휘 통제 시스템 간의 상호운용성을 디지털 기술로 향상시키려는 목표를 가지고 있습니다.

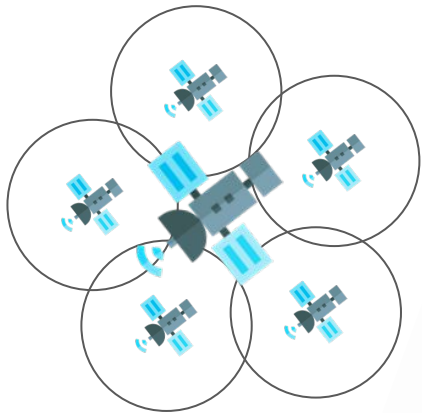
두 번째 목표는 매년 5월에서 6월 사이에 열리는 국방기술을 활용한 창업경진대회에 참여하여 C4I 시스템과 마이크로 위성 통합 알고리즘을 활용한 우수한 성과를 이루는 것입니다.

이 프로젝트는 국방과학기술의 활용성과 수출 가능성을 탐색하며, 혁신적인 국방 기술 개발을 통해 국가와 군에 기여할 수 있는 기회를 모색하고자 합니다.

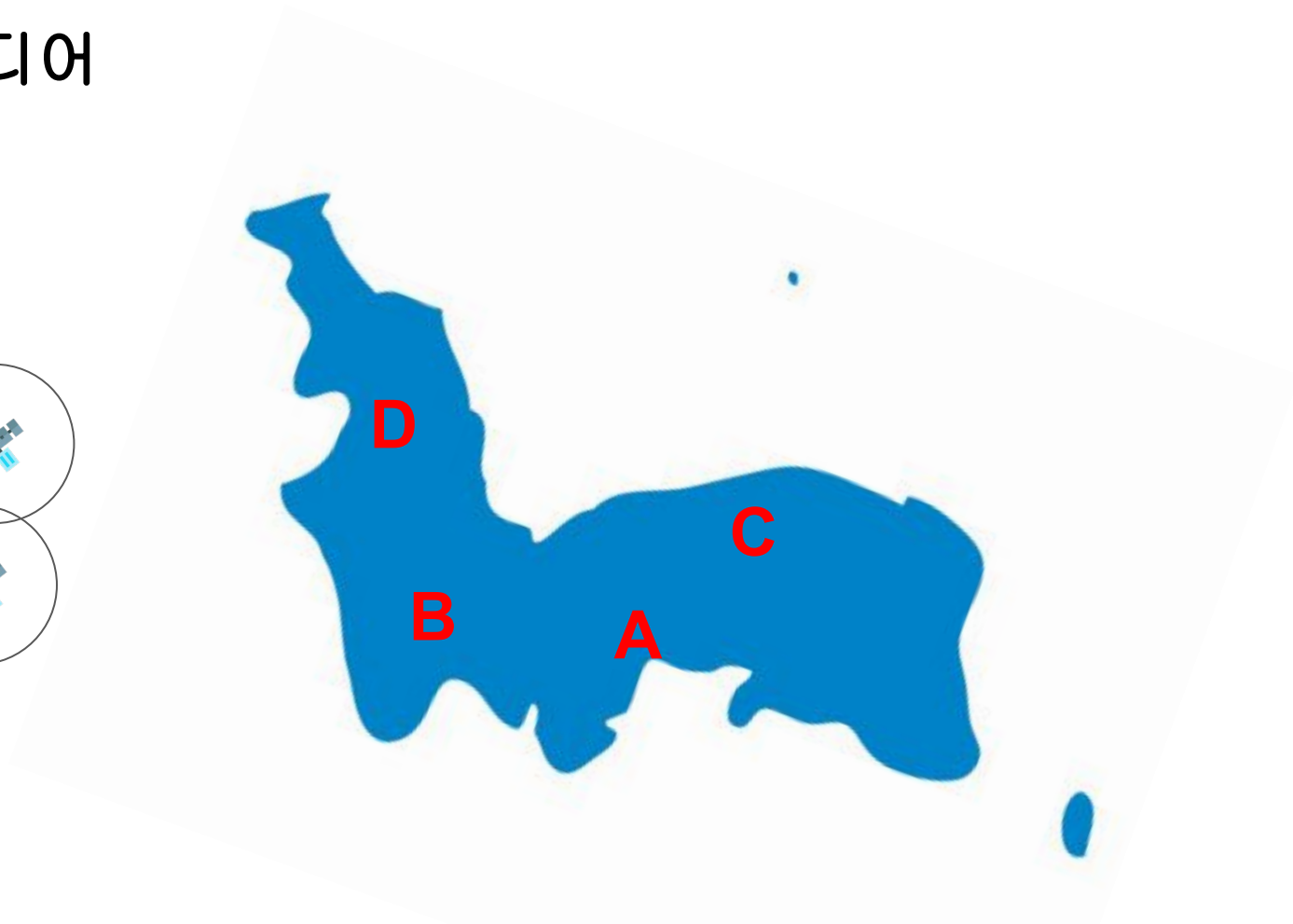
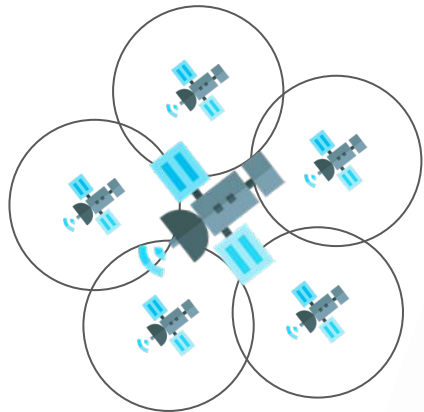
연구 아이디어



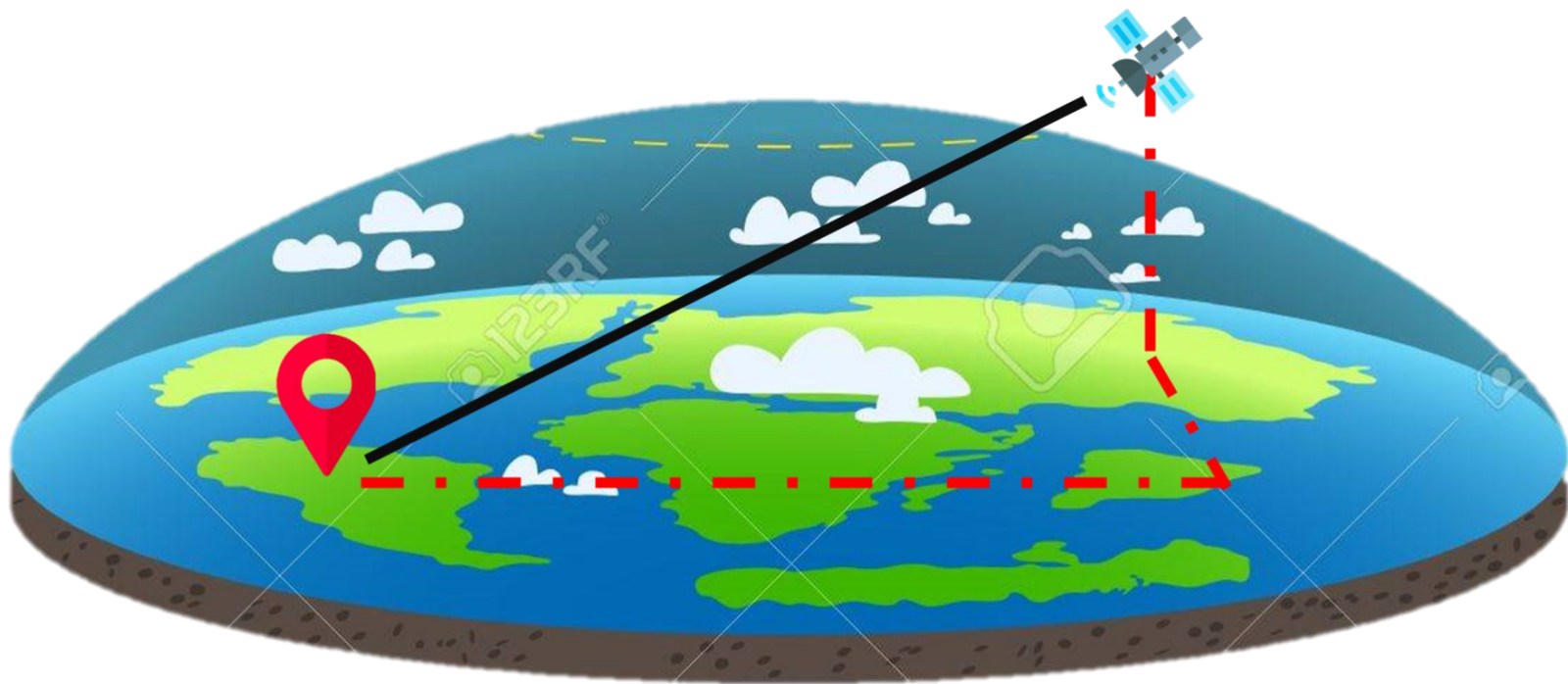
연구 아이디어



연구 아이디어



연구 아이디어



다중 군의 감시 임무 배정 알고리즘 개발

우리가 원하는 프로그램을 개발하고자, 사용할 수 있는 여러 알고리즘 들을 조사하면서 **k-nn** 알고리즘에 대해서 알게 되었다. **K-NN**은 머신러닝에서 분류와 회귀 작업에 사용되는 비지도 학습 알고리즘이다. 이 알고리즘은 새로운 데이터 포인트를 분류하거나 예측하기 위해 유사한 데이터 포인트를 찾는 데 거리 측정을 사용한다. 우리가 해야하는 것은 인공위성이 스스로 표적지 까지의 거리를 분석하고, 가중치를 계산해서 결정하는 알고리즘을 개발하려고 하기 때문에 **k-nn**이 적절하다고 판단했다.

해당 방법의 주요 특징은 아래와 같다.

- **단순성**: 이해하고 구현하기가 쉽다.
- **비선형 데이터 처리**: 선형 데이터 뿐만 아니라 비선형 데이터도 처리 가능하다.
- **매개변수 조정**: **K**라는 하나의 매개변수만 가지고 있고, 이는 최근접 이웃의 수를 결정한다.
- **훈련 데이터 저장**: **k-nn**알고리즘은 모델을 생성하지 않기 때문에 훈련 데이터를 저장해야 한다. 그리고 새로운 데이터 포인트를 분류하거나 예측할 때마다 이 데이터를 사용해야 한다.

또한 **k-nn** 알고리즘의 주요 장점은 아래와 같다.

- **효율성**: 대규모 데이터 세트에서도 효율적으로 작동할 수 있다.
- **견고성**: 해당 알고리즘은 이상치에 강하다.
- **다양성**: 다양한 유형의 데이터에 사용할 수 있다.
- **해석 가능성**: 결과를 해석하기 쉽다.

K-NN 알고리즘 학습

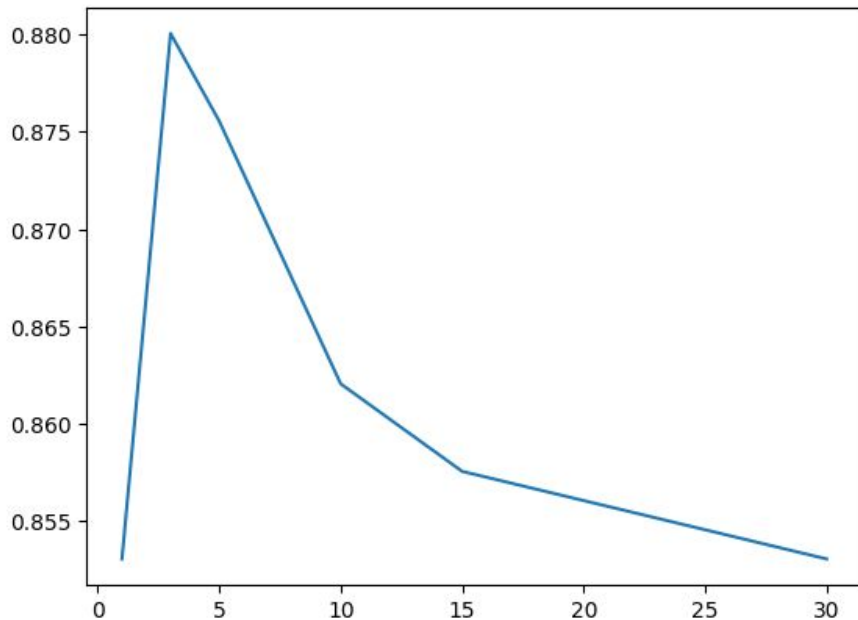
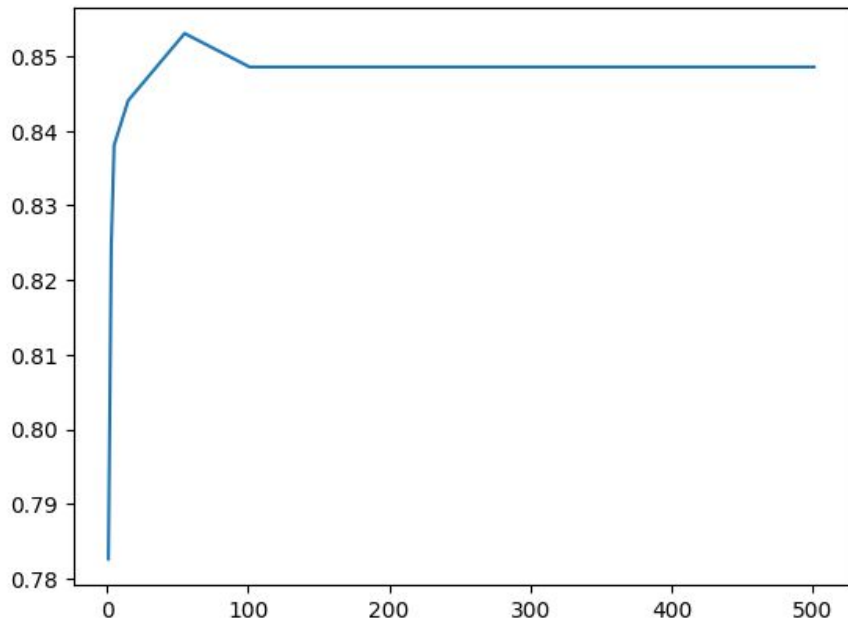
우리가 개발하고자 하는 알고리즘을 k-nn 기법을 바탕으로 개발 하려면 k-nn 알고리즘에 대해서 정확하게 알고 있어야 한다고 판단했다. 따라서 인터넷에서 구할 수 있는 데이터를 바탕으로 k-nn 알고리즘을 사용하여 데이터의 분석과 여러 계산을 연습해 보았다. 아래 코드는 k값에 따라서 변하는 데이터 예측 정확도를 계산해보는 코드이다. 해당 연습에서는 k가 55일때 최고의 성능이 나왔다.

```
from sklearn.model_selection import train_test_split
y = data['Response']
X = data.drop(columns=['Response'])
X_train, X_val, y_train, y_val = train_test_split(X, y, test_size=0.3)
numK= [1, 3, 5, 15, 55, 101, 501]
knn_scores=[]
for k in numK:
    # KNN 모델 생성 (K 값 설정)
    knn = KNeighborsClassifier(n_neighbors=k)
    # 모델 학습
    knn.fit(X_train, y_train)
    # 모델 평가 후 리스트에 저장
    knn_scores.append(knn.score(X_val, y_val))
#최대값 추출
best_score=max(knn_scores)
#최대값을 지정하는 인덱스(k값) 추출
best_k = numK[knn_scores.index(best_score)]
print(best_k)
```

... 55

K-NN 알고리즘 학습

또한 파이썬 툴을 사용하여 여러 상황을 부여하며, 각각의 k 값에 대한 정확도를 그래프로도 출력해 볼 수 있었다.



K-NN 알고리즘 학습

마지막으로 연습 데이터를 통해 추출한 최적의 하이퍼파라미터($k=55$)를 가지고 다시 데이터를 분석 했을때 평균 정확도를 계산해보았다. 또한 여러가지 함수를 통해 많은 데이터 분석을 시도해 보며 해당 알고리즘 기법을 익히는 시간을 가졌다.

```
▷ # 최적 하이퍼파라미터
best_k = 55
# Train/Test Set 분리 (70%/30%)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3)
# KNN 모델 학습 (최적의 k 값 사용)
knn = KNeighborsClassifier(n_neighbors=best_k)
knn.fit(X_train, y_train)

# 각 모델 평가
Knn_acc=knn.score(X_test, y_test)
# 정확도 출력
print(Knn_acc)
```

... 0.8275862068965517

향후 계획

합동군 감시 목표 배정 알고리즘 개발을 진행하여 테스트 해보기

국방기술을 활용한 창업경진대회 준비 (올해 개최 미정)

개발 알고리즘 성능평가 진행