

***TEZ ÖN ÇALIŞMALARI***

**Openstreetmap Kütüphanesini Kullanarak 5G İle Kablosuz İletişim Kuran Araçların SUMO Trafik Simülasyonunun Gerçekleştirilmesi**

**Doç. Dr. İzzet Fatih ŞENTÜRK**

**22435004016**

**İREM BÜYÜKBEKTAŞ**

**2023**

**SUMO (Simulation of Urban MObility) Nedir?**

Simulation of Urban MObility (SUMO), açık kaynaklı bir trafik simülasyon yazılımıdır. SUMO, kentsel alanlardaki yol ağı, araçlar, yayalar, trafik sinyalleri ve diğer ulaşım unsurlarını modellemek için kullanılır. Bu simülasyon aracı, trafik akışını analiz etmek, trafik yönetimi stratejilerini değerlendirmek, ulaşım projelerini planlamak ve trafik sıkışıklığı gibi sorunları çözmek için kullanılabilir.

SUMO, kullanıcıların gerçek dünya şartlarına benzer trafik senaryolarını oluşturmasına olanak tanır. Kullanıcılar, yol ağı topolojisini oluşturabilir, trafik akışını ayarlayabilir, araçların davranışlarını belirleyebilir ve çeşitli trafik yönetim senaryolarını simüle edebilirler. Bu sayede, farklı ulaşım senaryolarının nasıl çalışacağını ve etkilerini önceden analiz etmek mümkün olur.

SUMO'nun kullanım alanları arasında trafik optimizasyonu, trafik yönetimi stratejilerinin değerlendirilmesi, trafik akışının simülasyonu, ulaşım planlaması, trafik sıkışıklığı ve emisyon analizi bulunmaktadır. Ayrıca, SUMO'nun farklı simülasyon araçları ve genişletilebilir bir mimarisi vardır, bu da kullanıcıların ihtiyaçlarına göre özelleştirilmiş simülasyonlar oluşturabilmelerini sağlar.

**SUMO Kurulumu**

SUMO arayüzü için <https://www.eclipse.org/sumo/> bu sayfa üzerinde bulunan dosya indirildi ve bilgisayara kurulumu sağlandı. SUMO’da kullanılmak üzere python kütüphanesini de kuran aşağıdaki komut ile SUMO indirildi.

python -m pip install sumo

**SUMO Kullanarak Kentsel Ağlarda Trafik Simülasyonu**

1. **Yol ağı oluşturma:** İlk adımda, ***netgenerate*** aracını kullanarak her biri 200m uzunluğunda ve 3 şeritli 5x5 bir kare yol ağı oluşturuldu. Bu komut, grid.net.xml adında bir yol ağı dosyası oluşturacak:

netgenerate -- grid -- grid.number=5 -L=3 -- grid.length=200 -- output-file=grid.net.xml

1. **Rastgele seyahatlerin oluşturulması:** İkinci adımda, ***randomTrips.py*** aracını kullanarak belirli sayıda araç için rastgele seyahatler oluşturuldu. Bu komut, flows.xml adında bir akış dosyası oluşturacak :

randomTrips.py -n grid.net.xml -o flows.xml --begin 0 --end 1 --period 1 --flows 200

1. **Rotaların oluşturulması:** Üçüncü adımda, ***jtrrouter*** aracını kullanarak her bir aracın alacağı rotaları oluşturuldu. Bu komut, grid.rou.xml adında bir rota dosyası oluşturacak:

jtrrouter --flow-files=flows.xml --net-file=grid.net.xml --output-file=grid.rou.xml --begin 0 --end 10000 --accept-all-destinations

1. **Sürekli yönlendiricilerin oluşturulması:** Dördüncü adımda, sürekli yönlendirici dosyası oluşturuldu. Bu komut, rerouter.add.xml adında bir sürekli yönlendirici dosyası oluşturacak:

generateContinuousRerouters.py -n grid.net.xml --end 10000 -o rerouter.add.xml

1. **SUMO yapılandırma dosyasının oluşturulması:** Beşinci adımda, SUMO yapılandırma dosyasını oluşturulması gerekmektedir. Bu SUMO yapılandırma dosyası, ***grid.net.xml*** (yol ağı dosyası), ***grid.rou.xml*** (rota dosyası) ve ***rerouter.add.xml*** (sürekli yönlendirici dosyası) adlı dosyaların kullanılacağını belirtir. Ayrıca, ***grid.output.xml*** adında detaylı araç bilgilerini kaydedecek bir çıkış dosyası da tanımlanmıştır.

<configuration>

<input>

<net-file value="grid.net.xml" />

<route-files value="grid.rou.xml" />

<additional-files value="rerouter.add.xml" />

</input>

<time>

<begin value="0" />

<end value="10000" />

</time>

<output>

<fcd-output value="grid.output20.xml" />

</output>

</configuration>

1. **Simülasyonun çalıştırılması:** Son adımda, komut satırında SUMO'yu çalıştırmamız gerekiyor. Bu komut, SUMO arayüzünü açacak ve simülasyonu görsel olarak izlememizi sağlayacaktır. Araçların hız ve konum gibi ayrıntılı bilgileri, her 100 adımda bir grid.output.xml adlı çıkış dosyasına kaydedilecektir. Aşağıdaki komutu kullanarak simülasyonu çalıştırabiliriz:

sumo-gui -c grid.sumocfg --device.fcd.period 100

**SUMO Çalıştırıldığında Alınan Hata ve Çözümleri**

1. “phonopy” adlı paketi kurarken Microsoft Visual C++ 14.0 veya daha yeni bir sürümün eksikliğinden kaynaklı hata alındı. Microsoft C++ Builds Tools indirildi ve Microsoft Visual C++ Redistributable sürümlerinden 2015-2022 sürümleri eksik olduğu için indirme işlemi yapıldı.
2. sumo-grid.py sayfasında bulunan “flow-files” parametresinin artık kullanılmadığı için “route-files” olarak değiştirildi.
3. sumo-gird.py sayfasında path değişkeninde bir klasör yolu tanımlıdır. Bilgisayarda SUMO’nun Python araçlarını içeren “tools” klasörünü bulmak için aşağıdaki kod çalıştırıldı ve doğru path "C:\\Program Files(x86) \\Eclipse \\Sumo\\tools\\" olarak bulundu.

import sumolib

print(sumolib.\_\_file\_\_)

Ama kod çalıştırıldığında dosya yolu hatası devam etti ve path aşağıdaki gibi düzeltildi:

"C:\\Progra~2\\Eclipse\\Sumo\\tools\\"