

LLM 기반 최적화 모델 자동 생성 파이프라인

강윤환, 조한얼*, 김찬수*, 이상철

1 한국과학기술연구원 인공지능·정보·추론 (AI/R) 연구실

LLM-based Optimization Modeling Automation Pipeline

Yoonhwan Kang, Haneol Cho*, Chansoo Kim*, Sangchul Lee

1 AI-Information-Reasoning (AI/R) Laboratory, Korea Institute of Science and Tech. (KIST)

요약

본 연구는 전문가의 지식이 필수적인 수리 최적화 모델링과 코드화를 자동화하는 AI 기반의 지능형 파이프라인을 제안하고 그 성능을 검증한다. 여기에서는 대규모 언어 모델(LLM, Large Language Model)을 엔진으로 삼고 LangGraph를 프레임워크로 사용한다. 동 연구에서 개발한 파이프라인은 "생성-검증-피드백"의 순환적 자기 교정 구조를 코드 실행 시 발생하는 오류나 비가용성 문제를 스스로 해결할 수 있다. GPT-4o mini와 Llama-3 8B 모델을 대상으로 500개의 문제 데이터셋을 활용한 성능 평가를 통해, 약 87%의 높은 성공률을 얻어 실효성을 입증했다. 본 연구는 LLM이 단순 생성 도구를 넘어 능동적인 문제 해결 에이전트로 기능할 수 있는 가능성을 제시하고, 수리 최적화 기술의 새로운 접근법을 제안한다는 점에서 큰 의의가 있다.

I. 서론

최적화 기술은 생산 공정, 물류, 인력 배치 스케줄링 등 다양한 산업 분야에서 비용을 절감하고 효율성을 극대화하는 핵심 기술로 사용되어 왔다. 하지만 현실의 복잡한 문제를 수리적 모델로 정형화하고 이를 최적화 모델에 맞게 코드화 하는 과정은, 해당 분야의 전문 지식과 수리적 모델링 역량을 동시에 요구하는 매우 까다로운 작업이다. 이러한 이유로 여기에는 높은 시간 및 비용 부담과 전문가 의존성으로 인해 보편적인 확대가 저해되어왔다.

최근 LLM 기술은 빠르게 발전하며, 자연어를 이해하고 복잡한 논리를 수학적 모델로 변환하는 능력을 보여주어 본 연구에서 적극적으로 활용하였다.

본 논문은 이러한 배경 하에, LLM을 핵심 엔진으로 사용하고 LangGraph 프레임 워크를 통해 전체 워크플로우를 구축함으로써, 수리 최적화 전과정을 자동화하는 새로운 파이프라인을 제시한다.

II. 본론

최근 몇 년간, LLM은 문장과 언어를 기반으로 한 생성형 도구로 자리 잡았다. 전통적으로 전문가의 영역이던 최적화 모델링을 대규모 언어 모

델(LLM)로 자동화하려는 시도가 최근 목격된다. LLM을 활용하여 사용자의 자연어 문제 기술을 바탕으로 수학적 모델을 자동 정형화하고[1, 2], 이를 특정 최적화 모델에서 실행 가능한 코드로 구현하는 접근법이 제시되었다 [3, 4]. 본 연구에서 제안하는 파이프라인은 LLM의 수리모델 및 코드 생성 능력을 기반으로, LangGraph를 프레임워크로 활용하여 순차적 모델 생성 및 피드백 루프의 자동화 과정을 구현한다.

이 파이프라인은 "문제 이해 → 모델 생성 → 코드 구현 → 검증 및 피드백 → 모델 재생성 → 조건만족시 종료" 이라는 순환적 구조를 통해, 기존의 단방향성 구조보다 높은 성능 및 정확성을 달성한다.

2.1. 파이프라인 구조 및 구성 요소

시스템은 각 기능이 유기적으로 연결된 모듈 형태로 구성된다. 전체 워크플로우는 LLM 에이전트 구조 설계에 적합한 LangGraph 프레임워크를 사용하여 상태를 공유하는 그래프 형태로 관리되며, 각 단계의 결과가 다음 단계의 입력으로 전달된다.

첫 단계로, 사용자의 문제 시나리오가 입력되면 RAG (Retrieval Augmented Generation) 기술을 통해 사전 구축된 최적화 문제 벡터 데이터베이스에서 가장 유사한 '문제-모델' 예제를 검색한다. 이렇게 검색된

컨텍스트는 LLM이 문제의 핵심을 명확히 파악하고 표준화된 형식의 수학 모델을 생성하도록 돕는다.

다음으로, 강화된 컨텍스트를 바탕으로 LLM은 목적 함수, 결정 변수, 제약 조건을 포함한 최적화 모델을 정형화하고, 이를 특정 최적화 솔버에서 실행 가능한 코드로 자동 변환한다.

마지막 단계로 파이프라인의 핵심인 실행 기반의 자동 검증 및 자기 교정 루프가 작동한다. 생성된 코드는 실제 실행 환경에서 구동되어 최적해 발견, 비가용성 확인, 오류 발생 등 모델의 상태를 실시간으로 확인한다. 만약 실행 결과가 비가용으로 판명되거나 오류가 발생하면, 시스템은 해당 실패 정보를 개선을 위한 피드백으로써 기존 최적화 모델과 함께 LLM에 다시 전달한다. LLM은 이 피드백을 바탕으로 제약 조건 완화 등의 수정을 거친 새로운 수학 모델을 생성하며, 이 모델은 다시 코드화 및 검증 단계를 거친다.

이러한 순환적 수정 과정은 LangGraph의 조건부 분기 통해 작동하며 시스템이 스스로의 실수를 통해 학습하여 유효한 해를 찾아 나가는 능동적인 문제 해결을 가능하게 한다.

2.2 실험 설계 및 결과

설계한 파이프라인의 실효성을 검증하기 위해 구체적인 실험환경을 설정하고, 정량적인 성능 평가를 수행하였다.

2.2.1 실험 환경 및 모델

- 베이스라인 모델: 기본적인 파이프라인의 성능을 확인하기 위해 OpenAI의 GPT-4o mini 모델 API를 베이스라인으로 사용
- 로컬 비교 모델: API 모델의 한계를 파악하고 파인 튜닝의 가능성을 탐색하기 위해, Llama-3 8B 모델을 로컬 환경에 구축하여 비교
- 프레임워크: 모델들은 LangGraph 기반의 워크플로우 내에서 동작하며, 비가용성 및 에러가 발생했을 때 이를 해결하기 위한 수정루프가 설정. 특히 비가용성 결과가 발생하면, 시스템은 ISS(Irreducible Inconsistent Subset) 분석을 통해 만족하지 못한 제약 조건을 파악하고, 이를 프롬프트에 추가하여 수학적 모델을 재생성

2.2.2 데이터셋 및 평가 기준

- 데이터셋: 연구를 위해 총 500개의 "텍스트 시나리오와 정답(최적해)"으로 구성
- 평가 기준: 성공률을 핵심 지표로 한 성능지표로 선정. 생성된 코드가 도출한 최적 목적 함수 값이 데이터셋의 정답과 일치하는 경우를 성공으로 정의

2.2.3. 실험 결과 및 분석

데이터셋 테스트 결과, GPT-4o mini API를 사용한 베이스라인 모델은 약 87%의 성공률을 기록했다. Llama-3 8B 로컬 모델은 약 75%의 성공률을 보였다.

또한 파이프라인의 수정 루프가 필요한 여러 문제에서 시스템이 자율적으로 문제를 해결하는 것을 확인했다. 예를 들어 149번 문제의 경우, 최초 시도에서는 정답을 도출하지 못하였지만 두 번의 수정을 거친 후 정답을 도출하여 자기 수정 루프가 효과적으로 작동함을 확인하였다.

<표 1> 테스트셋 모델 별 실험 결과

	GPT-4o-mini w/ Feedback-loop	Llama-3 8B w/ Feedback-loop	GPT-4o-mini	Llama-3 8B
Success Rate	87%	75%	80%	71%
Pass Rate	99%	98%	81%	75%

III. 결론

본 연구는 전문가 의존성이 높았던 수리 최적화 모델링 과정을 자동화하기 위해 LLM 기반 LangGraph 구조의 지능형 파이프라인을 제안하고 그 실효성을 검증했다. 제안된 파이프라인은 GPT-4o mini 모델기반으로 약 87%의 높은 성공률을 달성했으며, 특히 "생성-검증-피드백"의 순환적 자기 수정 루프 구조가 실제 문제 해결 과정에서 효과적으로 작동한다. 이는 우리의 연구에서 제안한 프레임워크가 단순히 정형화된 모델을 생성하는 단계를 넘어, 실행 결과를 바탕으로 스스로의 오류를 진단하고 수정하는 능동적인 문제 해결 주체로 기능할 수 있음을 보여준다.

ACKNOWLEDGMENT

This research was funded by the grant Nos. 2023-00262155; 2024-00339583; 2024-00460980; and 2025-02304717 (IITP) funded by the Korea government (the Ministry of Science and ICT).

참 고 문 헌

- [1] A. Notz, D. T. Zhang, and N. P. Sudermann-Merx, "Unveiling the Potential of Large Language Models in Formulating Mathematical Optimization Problems," *arXiv preprint arXiv:2310.05210*, 2023.
- [2] J. Hakizimana, A. Oustry, and L. Tible, "Optimus: Optimization Modeling Using MIP Solvers and LLM," *arXiv preprint arXiv:2402.13412*, 2024.
- [3] G. Tack, C. N. Chou, M. D. SumA, and K. N. Brown, "Autoformulation of mathematical optimization models using LLMs," *arXiv preprint arXiv:2309.09242*, 2023.
- [4] K. Yang, W. H. Chen, J. H. Fu, S. Q. Liu, H. I. Ting, and C. J. Wang, "OR-LLM-Agent: Automating Modeling and Solving of Operations Research Optimization Problem with Reasoning Large Language Model," *arXiv preprint arXiv:2404.09825*, 2024.