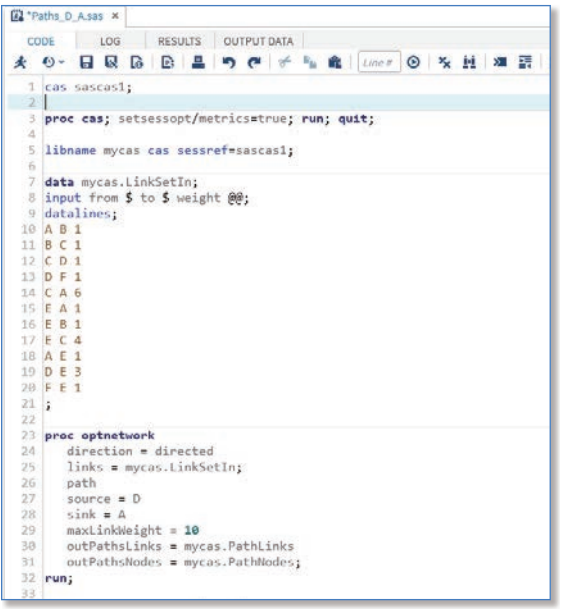
SAS® Optimization

복잡한 비즈니스 및 계획 문제를 가장 빠르게 해결하는 최적의 솔루션



SAS® Optimization은 어떤 솔루션인가?

SAS Optimization은 다양하고 강력한 최적화, 시뮬레이션 및 프로젝트 일정 관리 기법을 제공하여 제한된 리소스와 다른 관련 제약 사항 내에서 최상의 결과를 도출할 수 있는 활동을 파악합니다.

SAS® Optimization이 중요한 이유는?

기업은 더 많은 대안과 시나리오들을 검토하여 목표 달성을 위한 최적의 리소스 할당 및 계획을 결정할 수 있습니다. 운영 리서치 분석 통합으로 의사결정 프로세스의 안정성이 향상되고 이를 반복적으로 사용할 수 있습니다. 이를 통해 기존의 분석 및 BI 시스템을 최대한 활용하고 경쟁력을 확보할 수 있습니다.

SAS® Optimization은 누구를 위한 솔루션인가?

운영 리서치 기법을 통해 의사결정 안내 모델을 구축하여 실제 문제를 해결하려는 운영 리서치 또는 경영 과학 분야의 사용자들을 위해 설계되었습니다.

모든 기업이 직면하는 일상적인 비지니스 문제를 해결하려면 최상의 결과를 도출할 수 있는 활동을 파악해야 합니다. SAS Optimization 소프트웨어를 이용하면 매장의 적절한 재고 여부 및 신규 시설을 위한 최적의 입지 파악, 공급망 물류 최적화, 직원 업무 일정 수립 및 리소스 할당 등 여러 복잡한 문제를 신속하게 모델링하여 해결할 수 있습니다.

Optimization은 데이터 사이언스 분야에 포함되는 일종의 처방 분석으로, 제약 조건이 있는 특정 목적 함수값을 최대화 또는 최소화하는 수학적 알고리즘을 이용하고, 이를 통해 실현 가능한 솔루션 중에서 "최상"의 솔루션을 찾아냅니다. 이는 운영 리서치 및 경영 과학 분야에서 비롯되었습니다.

SAS Optimization은 강력한 모델링 기능과 솔루션 기법을 제공하여 기업이 더 많은 대안 시나리오를 검토하고 최상의 시나리오를 선택할 수 있도록 지원합니다. 이러한 유형의 문제 해결에 대한 일반적인 접근법으로는 선형계획법, 정수계획법, 확률론적 프로그래밍 및 제약조건 프로그래밍이 있습니다. SAS Optimization은 다음과 같은 기능을 제공합니다.

- 리소스 및 다른 관련 제약 조건 내에서 실행되면서 최상의 결과를 생산할 방법 파악
- 최적화된 리소스 할당을 결정하고 기업의 목표를 달성하기 위한 최상의 방법 선택
- 구조, 일관성, 적응성, 반복성이 추가된 의사결정 프로세스
- 기능별로 여러 개의 소프트웨어 패키지를 다뤄야 하는 번거로움 감소

주요 특징

- **의사결정 방식 개선** SAS Optimization은 최신 수학적 최적화 기법과 데이터 준비, 탐색, 분석 및 리포팅 기능이 하나로 통합된 단일 환경을 지원합니다. 이러한 통합 환경을 통해 복잡하고 현실적인 문제에 대한 최상의 대응책을 파악하여 적용할 수 있습니다.

- **복잡한 최적화 문제를 신속하게 해결** SAS Optimization은 SAS의 최신 분산 처리 인-메모리 엔진인 SAS® Viya®를 활용하여 최적화 모델링 결과를 놀라운 속도로 제공합니다. 어려운 문제들에 대한 최적의 솔루션을 신속하게 찾을 수 있습니다.

- **원하는 프로그래밍 언어를 선택해서 사용** Python, Java, R 및 Lua를 사용하는 프로그래머들은 SAS 코드를 배우지 않고도 SAS Optimization의 기능을 이용할 수 있습니다. 자신에게 익숙한 프로그래밍 언어를 그대로 사용하여 강력하고 신뢰할 수 있으며 검증된 SAS 알고리즘을 사용할 수 있습니다.

주요 기능

SAS Optimization은 강력하고 직관적인 대수(Algebraic) 최적화 모델링 언어와 다양한 알고리즘을 제공합니다. 선형, 혼합 정수 선형, 비선형, 이차 및 네트워크 최적화를 포함한 다양한 모델을 생산하고 제약 만족(Constraint Satisfaction) 문제를 해결할 수 있습니다. 이러한 고급 프로그래밍 기법들을 통해 한정된 리소스 하에서 최적의 방법을 찾아내고 목표를 달성할 수 있습니다.

SAS Optimization의 모든 기능은 다른 SAS 데이터 관리, 분석, 탐색 또는 리포팅 기능들과 호환되므로 변경이나 지원이 어려운 복잡한 분석 툴 모음을 사용할 필요가 없습니다.

운영 리서치 분석가는 일반적으로 데이터 및 인프라 문제를 해결하는 데 상당한 시간을 할애합니다. SAS Optimization을 이용하면 최소의 제약 하에서 모델링 작업에 착수할 수 있습니다. 통합 프로세스를 통해 보다 손쉽게 최초 검증을 위한 모델 검토를 수행하고 이를 수정하며 새로운 데이터로 모델을 실행할 수 있습니다.

다양한 최적화 모델을 지원하는 통합 모델링 언어

단일 모델링 및 솔루션을 이용하면 한 세트의 구문과 명령어만으로도 최적화 및 제약 만족 모델을 다양하게 구축할 수 있습니다.

시스템 및 관련 계획 문제의 핵심 요소를 포착하기 위해 최적화 모델을 구축할 때는 다른 요소들(정수 또는 이진 변수, 네트워크 구조 등)이 추가되기 때문에 모델 유형이 변경되는 경우가 많습니다. 최적화 모델은 정의와 정제를 통해 기능이 향상됩니다. SAS Optimization을 활용하면 모델이 변경되거나 개선되더라도 투명한 모델링 환경을 그대로 유지할 수 있으므로 시간을 절약하고 생산성을 향상시킬 수 있습니다.

모델이 정의된 유형에 정확히 들어맞지 않고 다른 요소들이 포함된 경우에는 SAS Optimization Solver를 SAS Optimization 언어 내 빌딩 블록으로 사용하여 하이브리드 사용자 정의 알고리즘을 구축할 수 있습니다.

- 직관적 모델식을 위한 유연한 대수 구문(Algebraic Syntax)
- SAS 기능의 충분한 활용 지원
- 선형, 비선형, 이차, 혼합 정수 솔버(Solver)를 직접 호출
- 다양한 문제에 대한 지원을 비롯한 사용자 정의 최적화 알고리즘에 대한 신속한 프로토타이핑 지원
- 업계 표준 MPS/QPS 포맷 입력 데이터 세트
- 강력한 프리솔버(Presolver)를 통한 실질적인 문제 규모 축소

강력한 최적화 솔버(Solver) 및 프리솔버(Presolver)

SAS Optimization은 단순성과 성능 개선을 위해 간소화 및 조정된 최적화 솔버 스위트(Solver Suite)를 제공합니다. 강력한 프리솔버가 실질적인 문제의 규모를 줄여 큰 문제를 신속하게 해결할 수 있습니다.

선형 프로그래밍 솔버는 프라이멀 및 듀얼 심플렉스, 네트워크 심플렉스, 크로스오버 내점(Interior Point)을 포함합니다. 일반적인 비선형 최적화 솔버(Solver)는 활성 세트(Active Set) 및 내점(Interior Point), 동시 해결(Concurrent Solve), 멀티스타트 알고리즘(Multistart Algorithm)을 포함합니다.

분산 분기한정(Branch-and-Bound) 혼합 정수 선형 프로그래밍 솔버는 절단면, 휴리스틱(Heuristics), 충돌 검색(Conflict Search), 옵션 조정을 포함합니다. 분산 컴퓨팅을 통해 최적의 솔루션을 찾기 위한 시간을 단축할 수 있으며 규모가 큰 문제일수록 그 시간이 단축됩니다. 최신식 솔버는 대규모 최적화 문제를 해결하도록 특별히 개발되었습니다.

- 선형 솔루션 알고리즘: 프라이멀 및 듀얼 심플렉스, 네트워크 심플렉스, (실험적) 크로스오버의 내점, 동시 해결 기능
- 혼합 정수 선형 프로그래밍 솔루션 알고리즘: 절단면을 가진 분기한정 정수, 프라이멀 휴리스틱, 충돌 검색, 옵션 조정
- 블록 앵글, 블록 대각선 또는 임베딩(Embedding)된 네트워크 구조와 관련한 선형 프로그래밍 및 혼합 정수 선형 프로그래밍 문제를 위한 분해 알고리즘(자동화된 Dantzig-Wolfe)
- 이차 프로그래밍 솔루션 알고리즘: 대규모 최적화 문제를 위해 설계된 최신식 솔버를 적용한 내점
- 비선형 솔루션 알고리즘: 액티브 세트, 내점 동시 해결 기능, 비블록 문제를 해결하는 멀티스타트 알고리즘

네트워크 최적화

PROC OPTMODEL과 PROC OPTNETWORK에서 모두 액세스 가능한 네트워크 알고리즘을 통해 네트워크의 특징을 조사하여 네트워크 중심 문제에 대한 최선책을 찾습니다. 최적화 및 진단 알고리즘에는 최단 경로, 최대 플로우, 최소 비용 플로우, 순회 판매원 문제, 연결 및 이중 연결 구성 요소, 클리크 및 사이클 열거, 최소 신장 트리, 선형 배정 등이 포함됩니다.

경로 열거 알고리즘은 지정 시작 노드와 끝 노드 간 모든 경로를 찾습니다. 하나 또는 다수의 시작 노드, 그리고 하나 또는 다수의 끝 노드를 지정할 수 있습니다.

- 최적화 및 진단 알고리즘 항목:
  - (Articulation Point를 가진) 연결 구성 요소 및 이중 연결 구성 요소
  - 클리크 및 사이클 열거(Clique And Cycle Enumeration)
  - 이행식 폐쇄(Transitive Closure)
  - 최소 절단(Minimum Cut)
  - 최소 스패닝 트리(Minimum Spanning Tree)
  - 요약(Summary)
  - 선형 배정(Linear Assignment)
  - 최소 비용의 네트워크 플로우(Minimum-Cost Network Flow)
  - 최단 경로(Shortest Path)
  - 순회 판매원 문제(Traveling Salesman Problem)
  - 경로 열거(Path Enumeration)

연결 구성 요소 알고리즘은 네트워크 내 연결 지정에 씬 포맷(Thin Format)을 이용하는 옵션을 제공합니다. 이 포맷은 특히 대규모 네트워

크 분석에서 적은 메모리를 사용하지만 네트워크 연결 정보를 저장하는 방식 때문에 모든 네트워크 알고리즘에 적합하지는 않습니다.

비선형 문제에 대한 더 나은 솔루션을 찾을 수 있는 멀티스타트 알고리즘

비볼록 비선형 최적화를 위한 멀티스타트 알고리즘을 통해 다양한 로컬 솔루션으로부터 최적의 글로벌 솔루션을 보다 쉽게 찾아낼 수 있습니다. 이러한 반복적인 알고리즘은 여러 개의 시작점을 선택한 후 각 시작점으로부터 동시에 최적화를 시작합니다. 모든 시작점으로부터 최상의 솔루션을 찾습니다.

분해 알고리즘

분해 알고리즘(자동화 Dantzig-Wolfe)은 규모가 크고 구조화된 선형 및 혼합 정수 선형 최적화 문제에 유용합니다.

분해 알고리즘은 전체 문제를 부분적인 문제들의 집합으로 분해한 후 각 문제별로 결정 변수를 적용하여 이들을 동시에 해결합니다. 하위 문제들에 대한 동시 솔루션이 전체 솔루션 프로세스와 조율되어 솔루션 도출 시간이 상당히 줄어듭니다.

로컬 검색 최적화 및 제약 조건 프로그래밍

로컬 검색 최적화(LSO) 솔버를 통해 기존의 최적화 솔버가 사용하는 추정을 따르지 않는 (주로 비선형) 최적화 문제를 해결합니다. 함수들은 블랙박스 시뮬레이션 등을 기반으로 불연속이거나 작동이 원할하지 않고 평가 계산 비용이 높을 수 있습니다.

도메인 축소/제약조건 전달과 룩어헤드(Look-ahead) 및 백트래킹(Backtracking) 등 다양한 검색 전략을 이용하여 제약조건 만족 문제를 해결합니다.
CLP 프로시저는 분산 모드에서 표준 제약 만족 문제를 해결합니다. 유일한 예외는 분산 모드에서 지원되지 않는 PROC CLP의 OBJECTIVE 문입니다. 또한 EVALVARSEL 문을 사용하여 여러 변수 선택 전략을 지정하는 경우 해당 전략은 가용 노드 수가 지정된 전략 수 이상인 경우에만 분산 모드로 실행됩니다. 그렇지 않은 경우에는 단일 노드에서 차례대로 실행됩니다.

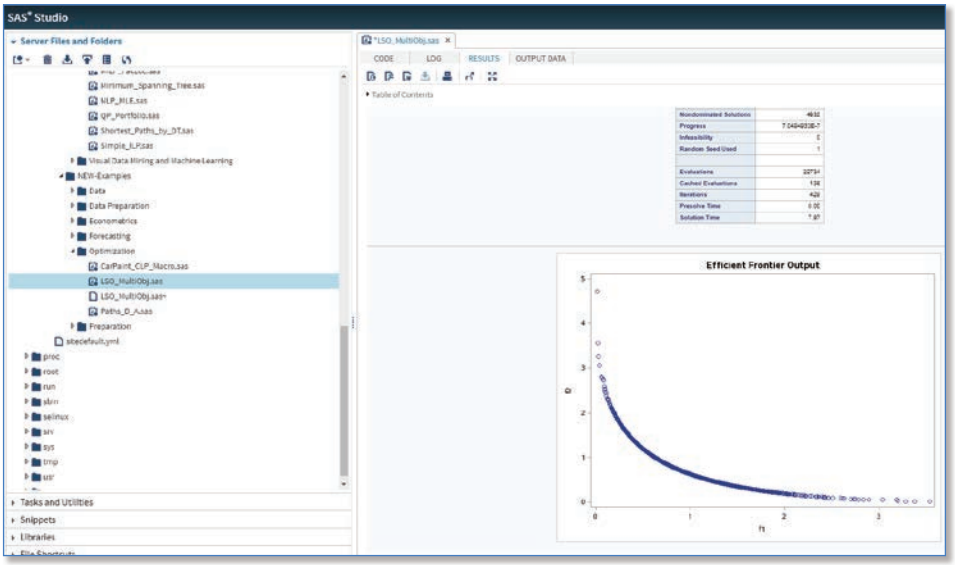
- 로컬 검색 최적화: 포괄적 알고리즘, 글로벌 GA형 휴리스틱 및 패턴 검색을 포함한 하이브리드 병렬 알고리즘, 다목적 최적화 (Multiobjective Optimization)
- 도메인 축소/제약조건 전달 및 다양한 검색 전략(look ahead and backtracking)을 이용하여 제약 만족(Constraint Satisfaction) 문제 해결

액세스 가능한 클라우드 기반 인-메모리 엔진

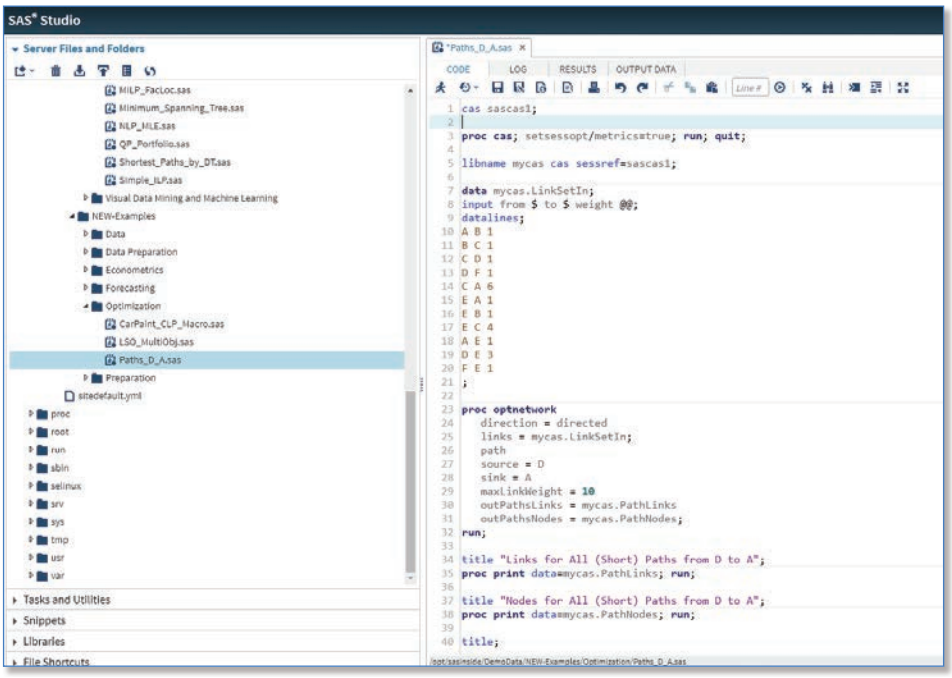
SAS Optimization은 SAS Viya 엔진을 사용하여 인사이트를 더욱 빠르게 도출합니다. SAS Viya는 고가용성, 더 빠른 인-메모리 프로세싱, 개방형 소스 언어로 코딩하는 기능, 기본 클라우드 지원을 포함한 새로운 기능들을 SAS Platform에 제공합니다.

SAS Optimization은 확장 가능하고 유연한 환경에서 공용 및 전용 클라우드를 제공할 수 있으며, 필요에 따라 프로세싱 속도를 조절할 수 있습니다. 가장 적합한 컴퓨팅 리소스를 이용하여 규모가 큰 문제들을 해결합니다.

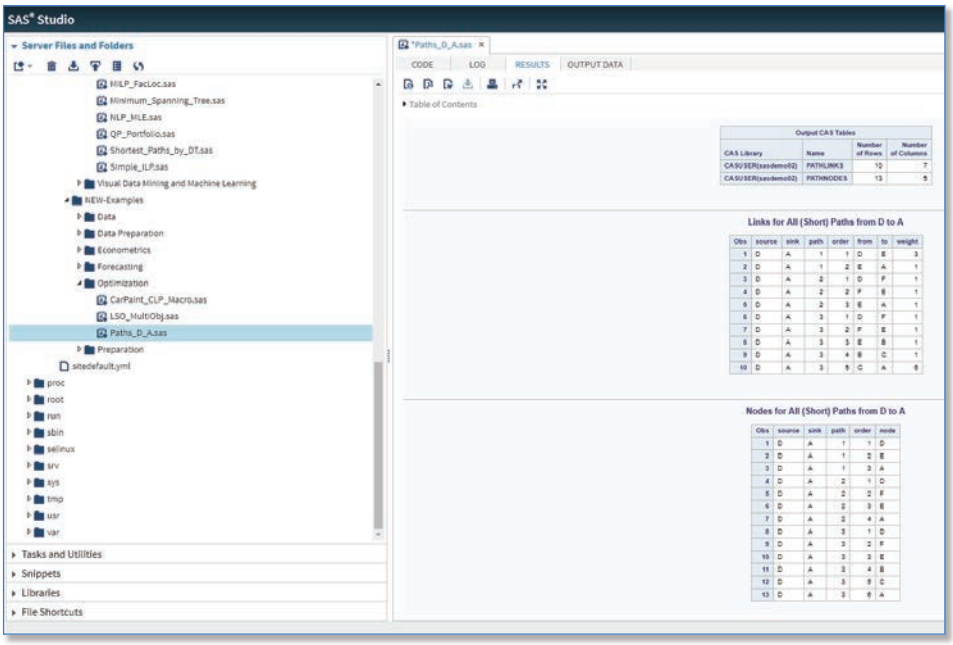
- SAS 플랫폼의 확장 가능한 분산형 인-메모리 엔진인 SAS Viya에서 실행
- 분석 및 데이터 작업을 여러 컴퓨팅 노드로 분산
- 인-메모리 데이터에 대한 신속한 멀티 사용자 접속 제공
- 고가용성을 위한 내고장성 포함
- SAS® Viya® REST APIs를 통해 SAS Analytics의 기능을 다른 애플리케이션에 추가 가능



비지배 솔루션의 파레토(Pareto) 최적 경계의 아웃풋. 이 아웃풋은 OPTIMODEL 프로시저 및 LSO 솔버를 통해 다목적, 비선형 최적화 문제를 해결함으로써 생성됩니다.



SAS Optimization의 SAS Studio 코딩 인터페이스는 자동완성 기능 등으로 구문 작성을 지원하므로 강력한 SAS Optimization 솔버를 보다 빠르게 실행할 수 있습니다. 다음 예시는 OPTNETWORK 프로시저에서 경로 알고리즘 사용에 관한 네트워크 데이터를 로딩하는 코드로, 노드D에서 시작해서 노드A로 끝나는 모든 경로를 찾습니다.



PROC OPTNETWORK에서 경로 알고리즘을 실행하여 도출된 아웃풋

더 자세한 내용은 sas.com/korea/optimization에서 확인하실 수 있습니다.

THE POWER TO KNOW.