Journal of the Korea Academia-Industrial cooperation Society
Vol. 21, No. 5 pp. 93-102, 2020

https://doi.org/10.5762/KAIS.2020.21.5.93
ISSN 1975-4701 / eISSN 2288-4688

PIDO 프레임워크를 이용한 시스템 레벨의 선박 최적설계 구현

박진원
대우조선해양 특수성능연구소

Implemention of the System-Level Multidisciplinary Design Optimization Using the Process Integration and Design Optimization Framework

Jin-Won Park
Special Purpose Performance Research Center, Daewoo Shipbuilding & Marine Engineering Co., Ltd.

요 약 자동차, 항공기, 선박과 같은 대형 복합 기계시스템 설계는 다분야 설계최적화의 영역이다. 다양한 영역의 전문지식과 경험을 동시에 요구하기 때문이다. 최근 급격한 기술발전과 더불어 인간의 편의 증진을 위한 요구로 이들 시스템의 복합도와 복잡도가 점증되고 있다. 이런 복합 시스템의 설계를 위해서는 도메인별 지식뿐만 아니라 다양한 분야의 지식, 경험 그리고 관점을 융합할 수 있는 통합 시스템 설계, 즉 다분야 설계최적화가 필요하다. 과거 다분야 설계최적화는 주로 설계자의 직관과 경험에 의존함으로써 해의 정확도나 시간 효율면에서 효용성이 크게 높지 않았다. 최근 다분야 설계최적화는 정보통신기술(IT)의 발전에 힘입어 프로세스통합 및 설계최적화(PIDO) 프레임워크에 의해 주로 구현된다. 본 논문은 오픈소스 PIDO 프레임워크인 RCE를 이용하여 합리적 수준의 노력과 시간 투입으로 효율적인 다분야 설계최적화를 구현하는 프로세스와 방법론을 찾고자 한다. 벤치마킹 예제로 벌크선 개념설계 모델에 본 논문이 제안한 다분야 설계최적화 프로세스와 방법론을 적용해 보았다. 최적설계 결과에 대한 시각적 분석을 통해 제안된 방법론의 타당성을 확인하였다.

Abstract The design of large complex mechanical systems, such as automobile, aircraft, and ship, is a kind of Multidisciplinary Design Optimization (MDO) because it requires both experience and expertise in many areas. With the rapid development of technology and the demand to improve human convenience, the complexity of these systems is increasing further. The design of such a complex system requires an integrated system design, i.e., MDO, which can fuse not only domain-specific knowledge but also knowledge, experience, and perspectives in various fields. In the past, the MDO relied heavily on the designer's intuition and experience, making it less efficient in terms of accuracy and time efficiency. Process integration and the design optimization framework mainly support MDO owing to the evolution of IT technology. This paper examined the procedure and methods to implement an efficient MDO with reasonable effort and time using RCE, an open-source PIDO framework. As a benchmarking example, the authors applied the proposed MDO methodology to a bulk carrier's conceptual design synthesis model. The validity of this proposed MDO methodology was determined by visual analysis of the Pareto optimal solutions.

Keywords : Design Optimization, Process Integration, PIDO, MDO, Mindmap, RCE, DSM, Pareto Optimal

*Corresponding Author : Jin-Won Park(DSME Co., Ltd.)
email: jwpark1@gmail.com
Received February 4, 2020 Revised March 16, 2020
Accepted May 8, 2020 Published May 31, 2020

1. 서론

자동차, 항공기, 선박과 같은 하드웨어 중심(hardware-intensive)의 대형 복합 시스템의 설계는 본질적으로 여러 공학분야 간의 상호작용과 다양한 결과분석을 요구하는 다분야설계 및 분석(MDA: multidisciplinary design and analysis, 이하 MDA) 활동에 해당한다. 구조, 유체, 추진, 기계, 전기/전자 등 다양한 공학 분야의 축적된 경험과 지식의 활용 그리고 노력의 균형된 통합이 필수적이기 때문이다. 오늘날 시스템은 급격한 기술발전과 더불어 인간의 편의 증진을 위해 더 복합화, 복잡화 되어가고 있다. 예로 가까운 미래에 인공지능 기반의 자율주행과 원격제어가 가능한 완전 전기자동차가 등장할 것이다. 항공기와 선박도 같은 기술 추종 추세이다. 이런 복합화 요구와 결과적으로 복잡성이 점증 되는 시스템의 다목적적 설계를 위해서는 도메인별 특화된 지식뿐만 아니라 다양한 분야의 그것들을 융합할 수 있는 통합적인 시스템 설계가 필요하다. 실제 시스템의 구성요소(하위시스템)는 서로 다르면서도 밀접하게 연계되어 있으므로 한 분야의 최적화가 반드시 전역 최적화(global optimization)를 보장하지는 않기 때문이다. 통합 시스템 관점의 전역 최적화를 위해서는 MDA에 최적화의 관점까지 추가한 다분야 설계최적화(MDO: multidisciplinary design optimization, 이하 MDO)가 필요하다[1-3].

본 논문은 추진, 구조, 중량 등 개별 하부시스템이 아닌 전체 시스템 레벨의 MDO 구현을 위해 오픈소스 PIDO 프레임워크인 RCE(remote control environment, 이하 RCE)를 이용하여 합리적 수준의 노력과 시간의 투입으로 효율적으로 이를 구현하는 설계방법론을 모색하기 위함이다. 최신 버전인 RCE 9.1.0을 이용하여 여러 최적화 관련 연구에서 벤치마킹 예제로 자주 인용되는 화물중량 3,000~500,000톤급, 속력 14~18노트급 벌크선(bulk carrier) 개념설계 모델에 본 논문에서 제안하는 MDO 프로세스와 방법론을 적용해 본다[4]. MDO로 얻은 최적설계 결과의 시각화 분석을 통해 제안된 MDO 방법론의 타당성을 확인한다.

2. PIDO 프레임워크

최근 MDO는 컴퓨터, IT기술의 발전에 힘입어 프로세스통합 및 설계최적화 (PIDO: process integration and design optimization, 이하 PIDO) 프레임워크에 의해 주로 구현된다. PIDO 프레임워크는 그래픽사용자 인터페이스(GUI: graphic user interface, 이하 GUI) 기반의 개발환경으로 2000년대 초부터 관련 연구가 본격화되었다[1,2].

컴퓨터공학에서 주로 사용되는 용어인 프레임워크는 개발환경을 지칭하는 것으로 본 논문에서는 여러 설계/해석 모듈의 연계, CAD/CAE 등 내외부 해석 자원의 통합, 모듈 간 데이터교환의 자동화, 분야/지역 간 분산협업, 최적화 및 결과생산의 자동화 등을 위한 일체의 기능 또는 도구를 제공하는 GUI 기반의 실체적 개발환경을 의미한다[2,3]. 기존 MDO연구에서는 'MDO 프레임워크' 또는 'PIDO 프레임워크'를 혼용하고 있으나[2,3], Fig. 1에서 보이는 바와 같이 MDO는 목적이고 PIDO는 수단이므로 용어의 명확화와 연구취지에 부합하도록 PIDO 프레임워크라고 부르고자 한다.

본 눈문은 Fig. 1와 같이 연성된 단일단계 MDF 접근법(coupled, single-stage multi discipline feasible approach)으로 벌크선 최적설계 문제를 정식화한다[5]. 구조, 중량, 추진 등과 같이 분해된 하부시스템 모듈은 각 계산단위(iteration)별로 서로 입력/출력값을 주고 받으며 조합된 시스템-레벨에 단일 최적화 모듈 적용을 통해 최적해를 얻게된다.

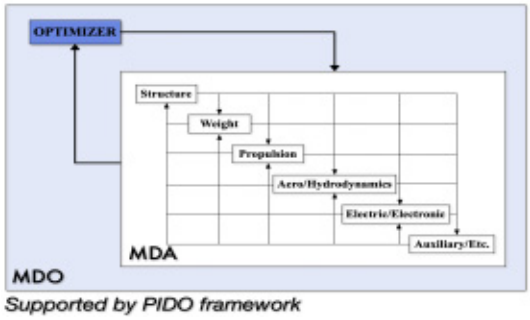


Fig. 1. Relationship between MDA, MDO, and PIDO

대표적인 PIDO 프레임워크는 Phoenix의 ModelCenter, Dassault의 iSIGHT, Esteco의 modeFRONTIER, Noesis Solutions의 Optimus, Dynardo의 optiSLang, 국내 피도텍의 PIAnO 그리고 오픈소스인 RCE가 있다. PIDO의 공통적인 핵심기능은 아래와 같다[1].

- **통합화**(integration): 다양한 설계 및 해석 모듈을 통합하고 조정
- **자동화**(automation): 디지털 프로토타이핑, 설계 및 해석 도구의 실행, 설계변수 변경 등을 자동화

- **최적화**(optimization): 입력변수 범위 내에서 제시된 제한조건을 만족하도록 반복계산을 수행하여 하나 또는 그 이상의 설계 목적을 최적화

PIDO S/W는 MDF(single-stage, multi discipline feasible), IDF(single-stage, individual discipline feasible), CO(multi-stage, collaborative optimization) 등와 같은 일반화된 MDO방법론을 자체 내장하지 않고 사용자가 직접 정식화해야 하는 점이 편리성 측면에서는 단점이자 유연성 측면에서 장점일 수 있다. 대표적인 상용 PIDO S/W 2종(국내·외)과 RCE의 주요특성을 Table 1과 같이 간단히 비교해 보았다.

Table 1. Key features of PIDO S/W: ( ) is relative rank

	ModelCenter	PIAnO	RCE
Interface	Simpler(1)	Simple(2)	Simple(3)
Applicability	Universal(1)	Universal(2)	Universal(3)
Optimizer	25(1)	2(3)	8(2)
Analysis	Easy(1)	Easy(2)	Difficult(3)
Budget	High(3)	High(2)	Open-free(1)

모두 입력/출력 인터페이스가 단순하고 쉬우며, 항공, 조선, 일반기계 등 적용 분야의 제한이 없다. ModelCenter, PIAnO 등과 같은 상용 프레임워크는 외부 CAD/CAE 자원과의 연계와 분산환경 지원 기능이 뛰어나지만 수억 원대에 이르는 도입비용과 블랙박스화된 내부코드로 설정과 실행은 쉽지만 사용자화(또는 자율성)가 어려운 단점이 있다. 반면 RCE는 JAVA 기반의 연구목적 오픈소스로 기능 면에서 상용 프레임워크에 크게 뒤지지 않으며 파이썬 스크립터와 엑셀을 실행용 컴포넌트로 지원하기 때문에 새로운 기능이나 모듈의 개발과 추가가 쉽다. 더불어 오픈소스 최적화 알고리즘인 DAKOTA 패키지를 내장하고 있어 연구나 실용적인 목적에서 부족함이 없다[6]. 단, 자율성이 높아서 설정과 실행이 상대적으로 조금 더 까다롭고 오픈소스이기 때문에 개발자의 적극적인 지원이나 구체적인 메뉴얼이나 관련 책자를 기대하기는 어렵다. 단, 매년 업그레이드가 제공되며 최근 적용사례가 늘고 있어 앞으로의 발전 가능성이 기대된다. 본 연구에서 RCE를 선택한 이유는 오픈소스임에도 불구하고 인터페이스, 최적화 알고리즘 및 실행 설정에 있어 상용 S/W에 비해 크게 부족함이 없었기 때문이다. 특히 파이썬 스크립터가 가능하다는 점은 코드의 이식성과 재사용성이 높다는 점에서 RCE는 연구목적에서는 가장 적합한 PIDO S/W라고 생각한다.

RCE는 2005년 초 독일의 우주항공청(DLR: deutsches zentrum für luft- und raumfahrt, 이하 DLR)을 주축으로 독일 내 여러 기업과 조선소 등이 참여한 공동 연구과제 SESIS(ship design and simulation system, 이하 SESIS)의 일부로 개발된 PIDO 프레임워크이다. Fig. 2에서 볼 수 있는 바와 같이 SESIS는 선박의 개념설계 및 시뮬레이션을 위한 컴퓨터시스템으로 독일 전역에 산재한 조선소, 연구소, 기업이 인터넷을 통해 협업할 수 있는 분산 개발환경을 제공하기 위한 산학중점 연구이다[7].

(a) Traditional way (b) SESIS way

Fig. 2. Interface and data exchange through SESIS[7]

RCE의 GUI는 Fig. 3과 같이 5개의 기본 윈도 또는 콘솔로 구성되어있다. 윈도는 사용자에 의해 재배열될 수 있으며 필요에 따라 종료하거나 다시 실행할 수도 있다. 따라서, 윈도의 위치나 유무는 경우에 따라 다를 수 있다. RCE 설정과 실행에 특히 관련된 윈도와 구성요소는 다음과 같다.

- Project Explorer (좌상단): RCE 프로젝트 관리 및 파일 탐색
- Workflow Editor (중상단): 워크플로 정의 및 설정
- Palette (우상단): 데이터흐름, 실행, 최적화/실험계획 등과 관련된 다양한 워크플로 컴포넌트 제공
- Properties (우하단 3번째 탭): 입력/출력 정의 등 선택된 모든 워크플로 컴포넌트의 설정
- Workflow Console (우하단 4번째 탭): 워크플로 실행 간의 실시간 결과와 진행을 콘솔에 출력
- Optimizer (우하단 마지막 탭): 워크플로 실행 결과 를 그래프 및 데이터로 출력. 단, 실행 후 워크플로 상의 Optimizer 컴포넌트를 이중클릭으로 선택하는 경우에만 나타남

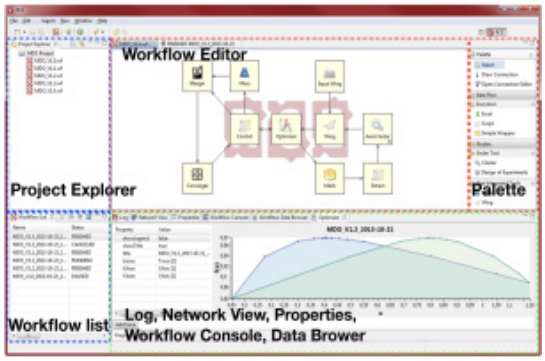


Fig. 3. Different views of RCE GUI[7]

Palette로부터 Workflow Editor로 최적화나 실행을 위한 다양한 컴포넌트를 끌어와 배열한 후 코드입력 및 입출력 조건 등을 설정하고 컴포넌트를 서로 연결한 후 마침내 실행하면 최적화 설계를 자동으로 진행하게 된다. 결과는 그래프나 스프레드시트 형태로 제공한다. 스프레드시트는 엑셀에 내장된 기능이나 별도의 통계분석 패키지를 이용하여 추가적인 분석을 가능하게 해 준다.

재사용 가능한 컴포넌트 방식을 기반으로 한 RCE는 MDO 개발 기간과 비용을 줄일 수 있다는 장점으로 항공기, 위성, 선박 등과 같은 복잡한 시스템의 설계, 시뮬레이션 및 최적화 연구에 사용되고 있다[6,7]. 애초 개발 의도가 물리적으로 분산된 자원과 노력을 네트워크를 통해 통합하기 위한 분산 개발환경이므로 여러 분야, 여러 지역에 분포한 공학자와 개발자가 복잡한 시스템의 설계와 시뮬레이션을 함께 하기 위한 분산형 협업설계 시 유용할 것이다.

3. MDO 구현

MDO는 MDA에 최적화가 추가된 것이다. MDA는 복합시스템의 설계/해석 코드 또는 모듈의 집합이라고 할 수 있다. 모듈을 단순히 PIDO 프레임워크에 배열(work flowing)한다고 해서 실행 차원의 비용과 시간 최적화 그리고 결과 차원의 최적 결과를 얻을 수는 없다. 기존 모델을 MDO 구현에 적합하도록 단순화하고 모듈화하는 사전처리 작업이 필요하다.

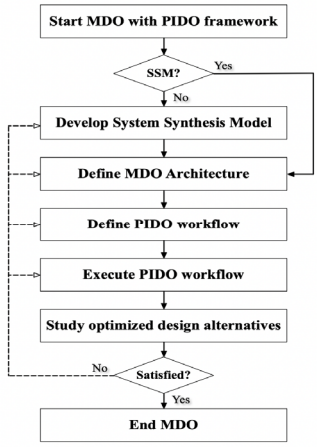


Fig. 4. Proposed procedure of MDO implementation

Fig. 4는 본 연구에서 사용된 MDO 프로세스로 크게 설계조합모델(SSM: ship design synthesis model, 이하 SSM) 개발, MDO 아키텍처와 PIDO 워크플로 정의, 실행과 결과 분석으로 나뉜다. 각 단계별 주요특징과 산출물을 분류하면 아래 Table 2와 같다.

Table 2. Characteristics of proposed procedure

	SSM	MDO	PIDO
Concept	Equating	Modularization	Integration
Outputs	Equations and parameters	Modules	Optimum solutions
Time consumption	High	Moderate	Low

SSM 개발단계는 과거 설계정보나 실험결과를 토대로 방정식을 얻는 단계로 다양한 자료와 수학모형 등을 분석, 검증하는 과정이 필수적이므로 시간이 많이 소모된다. MDO 아키텍처 과정은 다수 방정식 간의 상관/인과관계에 따라 이를 모듈화하는 과정으로 문제의 성격이나 규모에 따라 모듈의 크기나 수는 다를 수 있다. 마지막 PIDO 단계는 모듈들을 서로 연결, 모듈 간의 입/출력 관계를 정의하고 문제의 형태에 맞는 최적화 알고리즘을 적용하여 최종적으로 최적해를 얻는 과정으로 구분된다. 각 단계별 구체적인 절차와 방법에 대해서는 아래 3.1~3.4절을 참고한다.

3.1 SSM 개발

첫 번째 단계는 SSM 개발이다. 사전적으로 조합(synthesis)은 '잘못 따위가 있는지 알아보기 위하여 서로 맞추어 보다'라는 의미로 설계조합은 시스템공학(systems engineering)의 3단계 순차적 활동인 '요구조건 분석', '기능분석 및 할당'의 단계를 거쳐 얻은 물리적 계층구조를 실제 조합하는 것을 말한다[7]. SSM은 주로 이전의 설계결과나 분석, 실험 또는 운영 실적 등을 통해 얻은 데이터를 통계 모형화하여 얻는다. 항공기, 선박, 함정 등의 개념설계 수준에서 다양한 SSM이 개발되어 시스템이나 하부시스템의 최적설계에 활용되고 있다[2,4,7-9].

본 연구는 오픈소스 RCE를 이용한 MDO 구현 방법론의 개발이 목적이므로 다양한 선박설계 연구에서 벤치마킹 예제로 자주 인용되는 벌크선 개념설계 SSM을 사용한다[4]. Table 3은 해당 SSM을 구성하는 목적함수(3), 설계변수(6), 제약조건(9)을 설명한다. 여기서 사용되는 SSM의 구체적인 공식과 약어는 참고문헌[4]를 따른다.

Table 3. Major parameters of the bulk-carrier SSM

OBJECTIVES (3)	
Minimize transportation cost	CT (£)
Minimize light-ship weight	Wl (ton)
Maximize annual cost	CA (ton/year)
DESIGN VARIABLES (6)	
Length (overall)	L (m)
Beam	B (m)
Depth	D (m)
Draft	T (m)
Speed	Vk (knots)
Block coefficient	CB (-)
CONSTRAINTS (9)	
Length by Breadth	$L/B \geq 6$ (-)
Length by Depth	$L/D \leq 15$ (-)
Length by Draft	$L/T \leq 19$ (-)
Deadweight (DWT)	$3{,}000 \leq DWT \leq 500{,}000$
Draft and DWT	$T \leq 0.45\ DWT^{0.31}$
Draft and Depth	$T \leq 0.7D + 0.7$
Block coefficient	$0.63 \leq CB \leq 0.75$
Speed	$14 \leq Vk \leq 18$
Froude Number	$Fn \leq 0.32$
Stability	$GMT = KB + BMT - KG \geq 0.07\ B$

3.2 MDO 아키텍처 정의

아키텍처는 시스템의 구조 또는 구성을 의미하며 각 구성요소가 무엇이며 어떻게 상호작용하는지, 정보가 어떻게 교환되는지를 설명한다. MDO 아키텍처 정의단계는 위 단계에서 얻은 SSM을 MDO 구현에 적합하도록 분해(decomposition, 또는 단순화), 재구성하는 것을 말한다. 참고문헌[4]의 SSM은 계산시간을 줄이는 것을 목적으로 순차분해(sequential decomposition)한 것으로 전체 시스템을 여러 개의 하부시스템으로 분해하지 않고 하나로 취급하므로 MDO를 위한 SSM으로는 적합하지 않다[9].

간단한 문제의 경우 수식의 전개가 쉽고 계산시간까지 줄일 수 있어 편리하나 문제의 규모가 커지고 수식 간의 관계가 복잡해질 경우 다루기가 까다롭고 계산 결과에 수치적 오류가 포함될 가능성이 있다.

PIDO 프레임워크를 이용한 MDO 구현은 단일 SSM을 여러 개의 하부시스템으로 분해, 단순화하여 취급하는 것이 과정에서의 효율과 결과에서의 효과를 높이는데 더 유리하다[10].

복잡성을 단순화하기 위한 기법으로는 다양한 방법이 있으나 본 연구에서는 Fig. 5에서 볼 수 있는 바와 같이 마인드맵(mindmap)과 설계구조행렬(DSM: design structure matrix, 이하 DSM)을 사용한다. 마인드맵은 SimpleMind, DSM은 DSM Matrix v1.6을 사용하였다. 마인드맵은 일상에서 흔히 사용되는 시각화 기반의 생각정리도구로 복잡한 SSM을 가지치기하여 분야별로 체계화된 도식을 얻는데 유용한 도구이다. SSM의 복잡성을 그림 한 장으로 정리하여 전체를 한눈에 살펴볼 수 있다는 데 의미가 있다. Fig. 6의 DSM은 시스템 구성요소 간의 종속성(dependency)을 행렬로 도식화하는 것으로 시스템분석과 프로젝트 계획 등에 활용되는 범용 방법론이다[11]. 먼저 마인드맵으로 경험에 기반하여(heuristic-based) 원래의 SSM을 설계 관련 5개의 모듈, 제약조건 모듈 1개와 최적화 모듈 1개로 나누었다. 복잡도를 증가시켜 계산시간의 증가나 반복계산을 초래하는 각 모듈 간의 상호작용(참조)을 최소화할 수 있도록 변수나 수식의 구성을 반복 재검토하여 SSM을 재정의하였다. 다만, 마인드맵으로 1차 단순화된 SSM만으로는 실제 각 부분(모듈)간의 종속성을 최소화하는 데 한계가 있다.

DSM의 파티셔닝(partitioning) 기능을 이용하여 SSM의 복잡성을 증가시키는 피드백 요소를 제거 또는 최소화할 수 있다. 마인드맵으로 SSM을 1차적으로 분해

Fig. 5. Mindmap to simplify the typical bulk-carrier SSM

및 단순화하고, 2차로 DSM을 이용하여 SSM의 실행 시퀀스를 조정하는 순차적 개념이다. Fig. 6 (a)는 마인드맵으로만 얻은 1차 시스템 분해 및 단순화 결과이며, (b)는 DSM 파티셔닝으로 모듈 간의 종속성을 제거한 시퀀스 조정 결과이다. 2회에 걸친 처리로 MDO에 적합한 최종 MDO 아키텍처를 얻었다. 마인드맵으로 모듈 내부를 단순화(독립성 확보, 복잡도 감소)하고, DSM으로 모듈 간 외부적 종속성을 최소화(피드백 제거 또는 최소화)하여 최적의 MDO 실행 시퀀스를 얻을 수 있었다. 재조정된 시퀀스는 다음 단계 PIDO 워크플로 정의 때의 기준이 된다. 효과적인 MDO 구현과 모듈의 재사용을 위해서는 이 단계에서의 집중과 노력이 중요하다. DSM에 대한 구체적인 이론과 활용법은 참고문헌[11]을 참고한다.

ID	Name	Seq	1	2	3	4	5	6	7	8	9
1	DESIGN VARIABLE	1	1								
2	OBJECTIVE	2		2	●		●				
3	COST	3			3	●		●			
4	RESISTANCE	4	●			4					
5	WEIGHT	5	●			●	5				
6	PAYLOAD	6				●	●	6			
7	STABILITY	7	●						7		
8	FEASIBILITY	8	●					●	●	8	
9	CONSTRAINT	9	●			●		●	●		9

(a) draft DSM by mindmap

ID	Name	Seq	1	2	3	4	5	6	7	8	9
1	DESIGN VARIABLE	1	1								
4	RESISTANCE	2	●	4							
7	STABILITY	3	●		7						
5	WEIGHT	4	●	●		5					
6	PAYLOAD	5		●		●	6				
3	COST	6		●			●	3			
8	FEASIBILITY	7	●		●		●		8		
9	CONSTRAINT	8	●	●	●		●			9	
2	OBJECTIVE	9				●		●			2

(b) modified DSM after partitioning

Fig. 6. Optimized dependency sequence of the PIDO workflow

3.3 PIDO 워크플로 정의

재정의된 SSM 모듈을 RCE workflow editor에 개별 컴포넌트로 할당하고 각 모듈별로 계산/해석 코드를 프로그래밍하고 입출력을 정의한다. 모듈은 파이썬 스크립터 외 엑셀로도 정의할 수 있다[6].

Fig. 7은 톤수 계산을 위한 'WEIGHT' 모듈의 파이썬 스크립터이다. 코드 상단에는 RCE 지정 문법에 따라 입력데이터*(RCE.read_input("⟨variable⟩"))*를 정의하여 RCE가 연결된 다른 모듈(컴포넌트)로부터 데이터를 읽어오게 한다. 중간부에는 각 모듈별 수학모형을 프로그래밍하고 끝에는 다른 모듈로 전달할 출력데이터 *(RCE.write_output("⟨variable name⟩", ⟨variable⟩))*를 정의한다. 나머지 다른 모듈도 위에서 얻은 MDO 아키텍처의 최적 시퀀스에 따라 배열하고 순차적으로 프로그래밍하고 입출력데이터를 정의한다.

```
########## Inputs from optimizer(DVs)############
L=RCE.read_input("L")
B=RCE.read_input("B")
D=RCE.read_input("D")
Cb=RCE.read_input("Cb")
P=RCE.read_input("P")      # from the module 'RESISTANCE'

########## Main #############
Ws = 0.034*L**1.7 * B**0.7 * D**0.4 * Cb**0.5  # Steel weight
Wo = 1.0*L**0.8 * B**0.6 * D**0.3 * Cb**0.1    # Outfit weight
Wm = 0.17*P**0.9                               # Machinery weight
Wls = Ws + Wo +Wm                              # Lightship weight

########## OUTPUT #############
RCE.write_output("Ws", Ws)      # To COST
RCE.write_output("Wo", Wo)      # To COST
RCE.write_output("Wls", Wls)    # To PAYLOAD and OBJECTIVE
```

Fig. 7. Python code lines in the RCE workflow component 'WEIGHT'

각 모듈의 프로그래밍과 설정이 끝나면 모듈 간의 종속성을 정의하기 위해 RCE가 제공하는 연결편집기(connection editor)를 이용하여 모듈 간의 입출력 관계를 설정한다(Fig. 8).

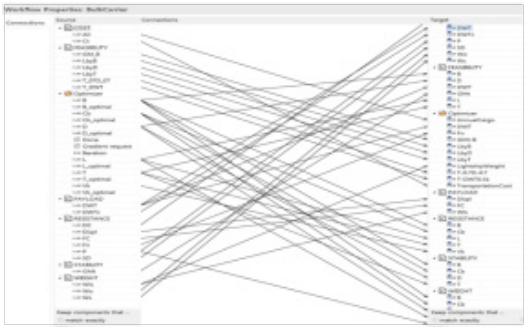


Fig. 8. Connection editor to interface the modules (or components)

모듈 내에 정의된 입출력데이터가 관계된 모듈 서로 간에 연결되지 않으면 RCE 실행이 허용되지 않으며, 데이터가 잘못 연결될 때는 의도하지 않은 잘못된 결과를 얻을 수도 있으므로 연결 설정에 유의해야 한다. 최종적으로 Fig. 9의 완성된 워크플로를 얻는다.

Fig. 9. Completion of the PIDO workflow definition

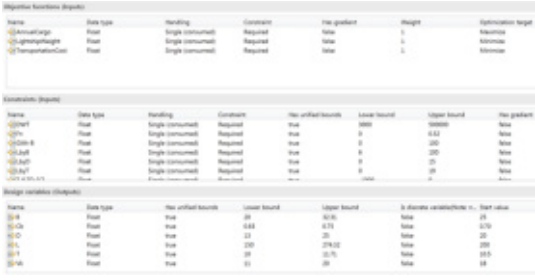
3.4 PIDO 워크플로 실행

주어진 MDO 문제의 유형(gradient-based 또는 derivative-free)에 적합한 최적화 알고리즘을 선택하여 설계최적화를 실행하는 단계이다. RCE는 미국의 Sandia 국립연구소에서 개발하여 여러 프로젝트에 적용, 신뢰성을 검증받은 최적화 알고리즘인 DAKOTA 패키지를 포함하고 있다[12]. 최신 버전의 RCE는 Table 4와 같이 8개의 최적화 알고리즘을 제공한다.

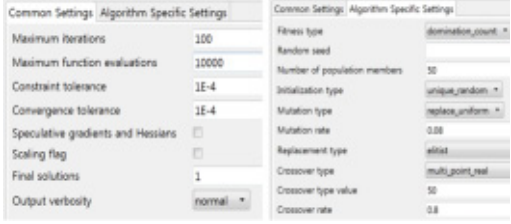
Table 4. DAKOTA optimization algorithms in RCE[12]

Category	Sub-category	Algorithm	Types of variable		
			Continuous	Discrete	General
Gradient-based	Local	Quasi-Newton	x		
Derivative-free	Local	COBYLA	x		
		HOPSPACK	x		
		Surrogate-based	x		
	Global	Coliny EA	x	x	
		SOGA	x	x	x
		MOGA	x	x	x

문제에 적용될 최적화 알고리즘은 미분 가능한 목적함수를 다룰지에 대한 여부(category), 해의 탐색영역을 전역이냐 국부로 할 것이냐(subcategory)의 여부에 따라 구분된다. 선박이나 항공기와 같은 복잡한 시스템에 대한 최적설계의 경우는 수렴속도와 해의 구현 측면에서 유리한 구배기반 국부최적화(gradient-based, local)에 비해 다소 시간이 걸리더라도 전체영역을 탐색하여 가장 좋은 최적의 해를 구하는 전역최적화가 적합하다. 또한 크기, 형상 최적화 등과 같이 미분가능한 방정식으로 목적함수를 모형화할 수 있는 일반적인 최적화 문제와 달리 연속형, 이산형 변수가 혼재된 복잡한 형태의 문제이며, 서로 상충하는 복수의, 다양한 형태의 최적화 목적함수(예. 선박의 경우 소형화 vs. 고효율)를 다루어야 한다. 따라서 본 논문에서는 이런 여러 요구조건을 모두 충족하는 유일한 해결책인 다목적 유전알고리즘(MOGA: multi-objective genetic algorithm, 이하 MOGA)을 사용하였다. 생명체의 진화방식을 모방한 MOGA는 정해진 설계변수의 범위 내에서 전역해를 사실상 무한 탐색할 수 있으며 다양한 형태의 변수 간에 결합성이 좋아 공학설계의 주요 관심사인 다목적 최적화 연구에 자주 사용된다[6].



(a) Inputs/outputs: objectives (3), constraints (8), design variables (6)



(b) Set MOGA algorithm parameters

Fig. 10. Set the MOGA optimizer for the given problem

최적화 설정의 첫 단계로 워크플로 최상단의 최적화 모듈을 선택하고 입출력데이터 테이블에 SSM에서 정의한 목적함수, 제약조건 및 설계변수를 입력한다 (Fig. 10 (a)).

MOGA 최적화 설정은 주로 개체(population), 세대(generation), 변이율(mutation rate), 교배율(crossover rate), 수렴 공차(convergence tolerance) 등의 수치를 조정한다. 본 연구에서는 한 세대 당 개체수는 50, 세대수는 100으로 설정하고 나머지 옵션은 기본값을 선택했다.

최적화 설정을 조정한 후 RCE GUI 상단의 녹색 화살표 아이콘(, 'execute workflow')을 클릭하고 관련 스크립터 solver를 연결하면 자동으로 최적설계를 수행하게 된다. 본 연구에서는 파이썬 스크립터를 사용했으므로 파이썬 실행 파일을 연결했다.

본 연구에서 활용한 RCE는 64비트 시스템의 Windows와 Linux OS에서 사용할 수 있으며 2019년 6월 배포된 최신 버전인 9.1이다. 운영환경은 Windows 7, PC자원으로 CPU는 2.6GHz Intel Core i7, 메모리는 2400MHz DDR4 16GB 등이다. 본 논문에서 다룬 최적화 문제는 개념설계 수준의 비교적 단순한 SSM으로 분산환경 구성과 같은 추가 설정 노력 없이 PC에서 단독실행(stand-alone) 되었다.

3.5 최적 설계후보 검토

PIDO 워크플로 실행 결과 반복계산을 통해 설계후보(design candidates) 1,142개를 얻었다. RCE는 자체적으로 그래프와 테이블 형태의 출력을 제공한다. Fig. 11은 상충하는 두 목적함수 연간화물량(annual cargo)과 경하톤수(light-ship weight)의 반복 계산에 따른 수렴 특성을 보여주는 2차원그래프와 데이터 테이블이다.

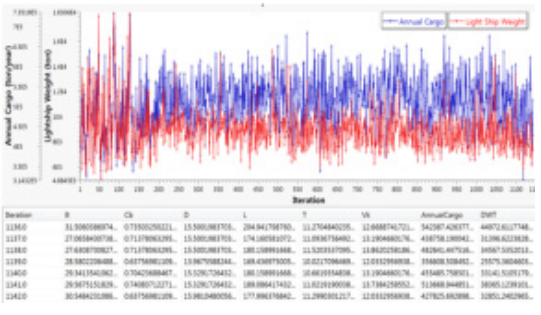


Fig. 11. Convergency properties and design data table

물론 상용 PIDO 프레임워크보다 내장된 데이터 분석 기능은 떨어지지만, 통계분석과 시각화 분석기능이 뛰어난 다른 통계분석 패키지와 결합하면 해소할 수 있는 문제다. 본 연구에서는 RCE에서 엑셀 파일로 결과를 출력한 후 통계패키지 JMP에서 파일을 임포팅하여 별도의 시각화 분석을 수행하였다. 연간화물량(망대), 운송비용(망소), 경하톤수(망소)의 3가지 목적함수가 있으나 사용자(고객) 관점에서의 큰 관심인 경하톤수(건조비용)를 x축으로, 연간화물량(수익)을 y축으로 하는 2차원 점그래프를 얻었다.

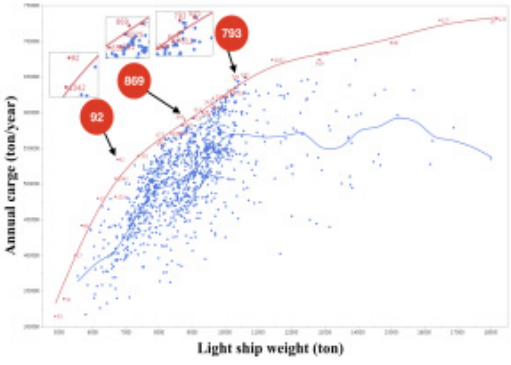


Fig. 12. Pareto frontier in the two-dimensional space

다목적 최적설계에서는 단일 최적해가 아니라 여러 개의 최적해가 존재한다. 이 최적해의 조합으로 이루어진 곡선(목적함수 2개일 때)을 파레토 경계(Pareto frontier, 또는 곡선)라고 한다. 목적함수가 3개일 경우 곡면이다. Fig. 12에서 총 1,142개의 점 중 55번, 31번, 92번, 869번, 793번 등 37개의 설계점, 즉 파레토 최적해(Pareto optimal solutions)를 근사적으로 잇는 파레토 곡선(적색 실선)을 얻었다. 하지만 37개 중 어느 하나만을 반드시 선택해야 하는 최종 의사결정 과정은 누구에게나 큰 인지적 부담이다. 이때 등장하는 개념이 '무릎점(knee point)'이다. 두 개의 목적함수가 있을 때 그중 하나의 작은 손실로 다른 하나의 큰 효용을 얻을 수 있는 영역으로 파레토 곡선의 구배가 '⌈'의 모양과 같은 불연속 지점 부근에 실제 우수한 설계 해가 존재할 가능성이 크다. 예를 들어 Fig. 12의 파레토 곡선상 무릎점에 해당하는 92번, 869번, 793번이 의사결정자가 우선 관심을 가지도록 권고되는 초기 우수 설계대안에 해당한다. 하나의 목적함수(망소: 경하톤수)의 미소한 증가로 또 다른 목적함수(망대: 연간운송량)의 값을 높일 수 있기 때문이다. 파레토 최적해에서 무릎점을 찾는 이유와 방법에 대해서는 참고문헌[13]을 참고한다.

Table 5. Summary of the best design alternatives(3)

DESIGN	92	869	793
Length(L), m	150.5	180.2	180.2
Breadth(B), m	32.2	32.0	31.5
Depth(D), m	13.9	16.3	15.5
Draft(T), m	10.2	11.5	11.6
Speed(Vk), knots	16.7	17.0	18.0
Block coeff.(CB)	0.72	0.71	0.74
Light ship weight (Wl), ton	6,761	8,776	10,297
Deadweight (DWT), ton	34,574	39,886	44,803
Annual cargo, ton/year	533,963	589,853	645,802
TYPES	Handysize	Handymax	

벌크선은 화물중량(DWT)에 의해 구분되는데 설계 92는 Handysize급(1~3.5만톤, L: 130~150m), 869와 793은 Handymax급(3.5~5.0만톤, L: 150~200m)에 해당한다. 화물선적 용량과 관련된 설계 제원 총 길이(L)와 흘수(T)도 실적 화물선의 경우에 비추어 볼 때 최적설계 결과가 합리적인 수준임을 알 수 있다[14].

4. 결론

오픈소스 PIDO 프레임워크인 RCE를 이용한 MDO 구현 프로세스를 살펴보았다. 선박설계에서 자주 인용되는 벌크선 개념설계 SSM을 MDO 구현에 적합한 아키텍처로 변환한 후 프로세스를 통합(process integration)하고 다목적 유전알고리즘(MOGA)을 적용한 설계최적화(design optimization)를 자동화하였다. 제약조건을 충족하는 파레토 최적해(Pareto optimal solutions) 37개로 파레토 경계를 근사했다. 37개 중 무릎점에 해당하는 3개의 우수 설계대안(best design alternatives)을 얻었다. 최적설계 결과가 실제 동급선박 제원의 범위에 대체로 해당하므로 MDO의 적용 가능성과 효과가 있다고 볼 수 있다. 본 연구에서 MDO 구현 결과 확인할 수 있었던 RCE의 장·단점에 대해 Table 6에 정리해보았다.

Table 6. Pros and cons of MDO implementation using RCE

Pros	- Provide user-friendly GUI environment to structure MDO problem · easy input/output setting · simple connecting between modules - Provide useful optimization algorithms - Generate limitless multiple solutions in relatively short period of time (e.g., few days) - Open publicly and update annually
Cons	- No kind manuals and samples - Less analysis modules compared to commercial S/W

오픈소스임에도 불구하고 최대한의 편의를 고려한 사용자인터페이스와 신뢰성 있는 최적화 알고리즘의 내장 그리고 실행조건 설정에 따라 사실상 제한 없이 복수대안의 신속한 생성이 가능했으며, 무엇보다 누구나 제약없이 사용할 수 있으며 매년 업그레이드되어 성능이 점차 향상된다는 점은 장점이다. 반면 상용S/W 수준의 설명서나 샘플을 찾기가 힘들며 구조/유동해석 등 다양한 공학해석 모듈이 지원되지 않기 때문에 필요할 경우 일일이 고된 사용자화가 필요하다는 점은 앞으로 개선되었으면 하는 부분이다.

본 연구에서 구현한 MDO 방법론은 선박뿐만 아니라 SSM을 이미 가지고 있거나 어렵지 않게 개발할 수 있는 다른 기계시스템의 통합 시스템 설계에도 같은 방식과 절차를 적용할 수 있다. 또한, 물리계층 구조상의 더 하위 요소로 나누어 모델을 만들거나 실험, 분석 등을 통한 데이터 추가 보완 등으로 SSM의 충실도(fidelity)를 높이면 좀 더 상세한 수준의 시스템 설계에도 적용할 수 있다. PIDO 프레임워크를 이용한 MDO는 더 짧은 시간, 더 많은 복수대안을, 비교적 정확하게 산출해냄으로써 설계 초기단계에서 의사결정의 신속성과 다양성 그리고 유연성을 보장한다는 측면에서 효과적인 설계방법론이라고 할 수 있다. 오픈소스 RCE는 제안된 MDO 방법론을 지원하는 데 부족함이 없음을 확인하였다.

References

[1] S. J. Min, "PIDO (Process Integration and Design Optimization)," *Korean Journal of Computational Design and Engineering*, vol.10, no.1, p.28, 2004.

[2] S. O. Cho, J. W. Lee, Y. H. Byun, "A Study On the Integration of Analysis Modules and the Optimization Process in the MDO Framework," *Journal of the Korean Society for Aeronautical & Space Sciences*, vol.30, no.7, pp.1-10, Jan. 2002.
DOI: https://doi.org/10.5139/JKSAS.2002.30.7.001

[3] K. K. Lee, "Proposal of PIDO Framework for Engineering," *Steel Engineering Symposium*, Korean Society for Technology of Plasticity, pp.140-152, May 2011.

[4] M. G. Parsons, R. L. Scott, "Formulation of Multicriterion Design Optimization Problems for Solution With

Scalar Numerical Optimization Methods," *Journal of Ship Research*, vol.48, no.1, pp.61-76, 2004.

[5] M. Balesdent, et al., "Space Engineering," Springer, Jan. 2017, pp.7-12.
DOI: https://doi.org/10.1007/978-3-319-41508-6_1

[6] DLR, "RCE User Guide, Build 9.1.1," German Aerospace Center, German, p.17, 2019.

[7] S. Schrodter, T. Gosch, "SESIS - Ship Design and Simulation System," Flensburger shipbuilders, Germany, pp.1-2.

[8] D. Seider, "Open Source Framework RCE: Integration, Automation, Collaboration," *4th Symposium on Collaboration in Aircraft Design*, DLR, France, 27 Nov. 2014.

[9] J. Park, et.al., "Recent Early-Phasse Naval Ship Systems Engineering for Republic of Korea Navy Surface Combatant Design," *Naval Engineers Journal*, vol.128, no.3, pp.103-115, Sep. 2016.

[10] H. W. Park, M. S. Kim, D. H. Choi, "Adaptive Decomposition Technique for Multidisciplinary Design Optimization," *Journal of the Korean Society for Aeronautical & Space Sciences*, vol.31, no.5, pp.18-23, June 2003.
DOI: https://doi.org/10.5139/JKSAS.2003.31.5.018

[11] J. Park, "Application of Systems Engineering Based Design Structure Matrix Methodology for Optimizing the Concept Design Process of Naval Ship," *Journal of the Society of Naval Architects of Korea*, vol.56, no.1, pp.1-10, Feb. 2016.
DOI: https://doi.org/10.3744/SNAK.2019.56.1.001

[12] B. M. Adams, et al., "A Multilevel Parallel Object-Oriented Framework for Design Optimization, Parameter Estimation, Version 5.4," User's manual, Sandia National Laboratories, USA, pp.113-135.

[13] J. Branke, K. Deb, H. Dierolf, M. Osswald, "Finding Knees in Multi-objective Optimization," Springer, USA, 2004, pp.722-731.
DOI: https://doi.org/10.1007/978-3-540-30217-9_73

[14] Stockcargo, Types of Bulk Carriers Ship, Typical Features [Internet], c2013[cited Dec. 2013], Available From: http://stockcargo.eu/types-of-bulk-carriers-ship-blogarticle/ (assessed Mar. 12, 2020)

박 진 원(Jin-Won Park) [정회원]

- 2003년 2월 : 서울대학교 대학원 조선해양공학과 (공학석사)
- 2008년 9월 : 미국 Virginia Tech 항공해양공학과 (공학박사)
- 2009년 12월 ~ 2011년 12월 : 국방과학연구소 파견 연구원
- 2016년 6월 ~ 2018년 11월 : 방위사업청 시스템공학 담당
- 2019년 9월 ~ 2020년 5월 : 한얼솔루션 통합설계실 수석연구원
- 2020년 6월 ~ 현재 : 대우조선해양 특수성능연구소 책임연구원

〈관심분야〉
시스템공학 분석(SE Analysis), Data Science, Concept Design, Set-Based Design, Design Optimization