

혼합 유전 알고리즘을 이용한 GDP/MINLP 로 표현된 공정 최적화

Process Optimization Formulated in GDP/MINLP Using Hybrid Genetic Algorithm

송 상 옥, 장 영 중, 김 구 회, 윤 인 섭
(Sang-Oak Song, YoungJung Chang, Kuhoi Kim and En Sup Yoon)

Abstract : A new algorithm based on Genetic Algorithms is proposed for solving process optimization problems formulated in MINLP, GDP and hybrid MINLP/GDP. This work is focused especially on the design of the Genetic Algorithm suitable to handle disjunctive programming with the same level of MINLP handling capability. Hybridization with the Simulated Annealing is experimented and many heuristics are adopted. Real and binary coded Genetic Algorithm initiates the global search in the entire search space and at every stage Simulated Annealing makes the candidates to climb up the local hills. Multi-Niche Crowding method is adopted as the multimodal function optimization technique, and the adaptation of probabilistic parameters and dynamic penalty systems are also implemented. New strategies to take the logical variables and constraints into consideration are proposed, as well. Various test problems selected from many fields of process systems engineering are tried and satisfactory results are obtained.

Keywords : genetic algorithms, simulated annealing, MINLP, GDP

1. 서론

최적화 기법은 자연과학과 공학의 여러 분야에서 연구되어 온 주제이다. 최적화 기법의 선택은 다루고자 하는 계의 특성에 따라 달라진다. Wolpert 등 [1][2]이 제안한 no free lunch 정리에 의하면 최적화 기법은 그 기법이 잘 다루는 유형의 문제에 편향되어 있을 수밖에 없으며, 따라서 전체 문제들의 공간에 대해서 보여주는 평균적인 수행 성능은 모두 같게 된다. 이러한 평균적인 수행 성능의 측면을 떠나서라도, 최적화 기법의 선택에는 효율성과 신뢰성의 두 양 극단의 절충이 필요하게 된다.

공정시스템 분야에서 다루는 분야에는 공정 합성을 주축으로 하여 공정 개발, 공정 수정 등이 포함된다. 공정 설계 문제로 통칭되는 이들 계의 수학적 형태는, 수많은 물질 및 에너지 수지식, 마찬가지로 많은 여러 열역학적 방정식, 장비나 유틸리티의 비용식, 그리고 여러 경험식들과 결합되어 나타난다. 이러한 큰 규모의, 비선형성이 강한 계를 최적화하고자 한다면 그에 상응하는 최적화 기법이 필요한 것은 자명하다. 공정시스템 분야에서 다루는 최적화 문제 중 하나인 다기간 최적화 (multiperiod optimization)는 최근에 각광을 받기 시작한 적용 분야이다. 여기에는 기존의 공정 설계 문제에 회분식 공정의 스케줄링과 연속 공정의 제품 생산 주기 등의 의사 결정이 포함된다. 또한 시장의 변화를 고려한 원료의 구입과 생산 제품의 판매 등의 supply chain 최적화도 포함된다.

이러한 최적화 문제들을 정식화하고 해결하는 기법은 크게 두 가지로 대별되어 왔다. 그 하나는 규칙 기반 경험법칙

으로 표현된 문제를 인공지능 기법들을 이용하여 해결하는 방법이고, 다른 하나는 수리계획법 (mathematical programming)으로 문제를 구성하고 해를 찾는 방법이다. 전자의 대표적인 예로는 Douglas [3]가 제안했던 계층적 분해 방법 (Hierarchical Decomposition)과 Linnhoff에 의해 제안된 Pinch Analysis 등의 방법을 들 수 있다. 하지만 이들은 공정의 크기가 커질수록, 체계적인 구조에 대한 고찰이 결여되고, 그 결과가 항상 유효한 것이 아닌 경험에 의존하는 등의 한계를 드러내게 된다. 또 다른 방식인 수리계획법에 의한 최적화는 최근까지 MINLP (Mixed Integer NonLinear Programming)로 표현되어 왔고 그 결정론적 해결 알고리즘의 개발에 초점이 맞추어져 왔다. 최근에는 MINLP가 가지고 있는 내재적인 한계를 극복하기 위해 GDP (Generalized Disjunctive Programming)가 제안되었고, 이들 문제를 해결하기 위한 알고리즘들이 개발되고 있다. 주로 Branch and Bound 방식에 기반을 둔 이러한 알고리즘들은, 적당한 크기의 잘 정식화된 문제들의 전역해를 보장해 주고 있다[4].

하지만 이들의 접근 방법은 중첩 구조가 가능하거나 혹은 발생 가능하다는 것을 전제하고 있다[5][6]. 또한 이들 알고리즘은 문제가 다극점을 가질 지라도 하나의 전역해를 찾는 데 중점을 두고 있다. 이는 전역해 뿐만 아니라 여러 국지해를 필요로 하는 몇몇 문제에 대해서 그 한계를 드러내게 된다. 예를 들면, 비용을 최소화 하는 전역해가 구해졌다고 하더라도, 그 전역해 범위에서 공정을 운전하기가 부담스러운 경우가 있다. 또한 이러한 접근 방법이 가지는 문제점중의 하나는, 문제의 구조와 탐색 공간이 복잡해질수록 그 해결 시간이 기하급수적으로 증가하게 된다는 것이다.

Holland[7]와 Goldberg[8]에 의해 체계적으로 정립된 유전 알고리즘 (Genetic Algorithms)은 복잡한 생태계를 모사함으로써 적합도가 높은 개체를 골라내는 자연 선택방법을 사용한

논문접수 : 2002. 7.11., 채택확정 : 2002. 12. 24.

송상옥, 김구회, 윤인섭: 서울대학교 응용화학부

(andy@pslab.snu.ac.kr / kimkh@pslab.snu.ac.kr / esyoon@pslab.snu.ac.kr)

장영중: UIUC(ychang10@uiuc.edu)

다. 이 방법은 진화 전략 (Evolutionary Strategy)과는 다른 접근 방식에서 출발했으나 서로간의 장단점을 취합하는 과정에서 현재에는 그 경계가 모호해졌다. 이들 방식은 개체군을 가지고 탐색을 시작함으로써 동시에 공간의 여러 부분을 탐색할 수 있다는 장점이 있다.

Kirkpatrick 등[9]에 의해 제안된 의사 풀림 (Simulated Annealing)은 분자들의 열 평형 하에서의 계의 거동을 나타내는 통계 역학과 조합 최적화의 유사성에 기반하고 있다. 의사 풀림에 사용되는 통계 변수들의 열역학적 취급은 그 이후에 금기 탐색 (Taboo Search)이나 대규모 단계 마르코프 체인 (Large-Step Markov Chain) 등의 무작위 탐색 기법에 채용되어 왔다. 이 방식은 적당히 작은 범위에서 다른 미분 기반 탐색 기법처럼 hill-climbing 의 성질을 가진다.

두 최적화 기법을 혼합해서 사용하는 데에는 크게 세 가지의 방법이 있다. 그 하나는 하나의 최적화 기법을 다른 최적화 기법의 meta 최적화 기법으로 사용하는 것이다. 즉, 하나의 최적화 기법에 사용되는 매개변수들을 다른 최적화 기법으로 최적화한 다음, 최적화된 매개변수들을 사용하여 전자의 최적화 기법으로 원래의 문제 공간을 탐색해 나가는 방법이다. 또 다른 하나는 두 최적화 기법을 엄밀히 비교하여 문제 공간에 알맞은 정성적인 장점들을 취합하여 새로운 기법을 만들어 내는 것이다. 이 때, 새로이 만들어진 기법은 전혀 다른 수학적 특성을 가지는 모델이 될 확률이 높으며, 새로 만들어진 기법으로부터 원래의 두 기법을 정확히 다시 분리해 내는 것은 불가능하다. 예를 들면 유전 알고리즘과 의사 풀림의 혼합 기법 중에서 유전 알고리즘을 그 주축으로 하되 그 유전 연산자들을 의사 풀림의 형태로 바꾸는 것을 들 수 있다. Sirag 와 Weisser[10]가 제안한 열역학적 연산 확률을 가지는 유전 연산자라든가, Adler[11]가 제안한 SAM (SA-Mutation)과 SAR (SA-Recombination)은 그 대표적이 예라고 할 수 있다. 마지막 혼합 기법은 하나의 최적화 기법으로 문제를 다루다가 적당한 선에서 다른 최적화 기법으로 그 동안 얻은 결과를 인계하는 것이다. 결과를 인계받은 최적화 기법은 자신의 알고리즘대로 최적화를 수행해 나간 후 그 값을 직접 결과로서 제시할 수도 있고, 다시 앞의 최적화 기법에 되돌려 반복 수행을 할 수도 있다. 전자는 mixed 알고리즘, 후자를 hybrid 알고리즘이라고 구분하기도 한다.

주로 조합 최적화 문제에 적용되었던 유전 알고리즘과 의사 풀림은 그 범위가 실수를 매개 변수로 취하는 함수 최적화로 확장되어 왔으며, MINLP 를 풀기 위한 도구로서 시도된 바 있다. 본 연구에서는 이를 더 확장하여 GDP 형태로 모델이 세워진 최적화 문제를 해결하기 위한 유전 알고리즘의 원형을 제시해보고, 유용한 방법론을 제안하였다.

유전 알고리즘을 전역 최적화 도구로, 의사 풀림을 국지 최적화 도구로 사용할 수 있도록 결합하였고, 다극 탐색 기법인 다중 Niche 군집 방법을 채택하였으며, 유전 알고리즘 연산자들의 확률을 진화 세대수에 맞추어 적응시켰다. 제약 조건들을 다루기 위한 동적 벌칙항을 도입하였으며, 논리 변수에 종속되는 제약 조건들을 다루기 위한 기법들을 고안하였다.

II. 다기간 최적화 문제의 정식화 (Formulation)

1. MINLP 정식화

공정 합성에서 나타나는 일반적인 MINLP 문제의 형태는 다음과 같이 표현될 수 있다.

$$\begin{aligned} \min_{x,y} Z &= f(x,y) \\ \text{s.t. } h(x,y) &= 0, g(x,y) \leq 0 \\ x &\in X \subseteq \mathbb{R}^n, y \in Y = \{0,1\}^q \end{aligned} \quad (1)$$

여기서 x 는 n 개의 실수 변수를 원소로 가지는 벡터이고, y 는 q 개의 이진 변수를 원소로 가지는 벡터이다. $h(x,y)=0$ 은 m 개의 등식 제약조건을 나타내며, $g(x,y) \leq 0$ 은 p 개의 부등식 제약조건을 나타낸다. $f(x,y)$ 는 우리가 최소화 시키고자 하는 목적함수다. 좀더 고찰하고자 하는 문제의 범위를 줄여서, 0-1 의 이진 변수에 대해서는 선형이고, 실수 변수에 대해서는 선형 혹은 비선형인 문제는 다음과 같이 정식화할 수 있다.

$$\begin{aligned} \min_{x,y} Z &= f(x) + d^T y \\ \text{s.t. } h(x) &= 0, g(x) \leq 0 \\ Ax &= a, By + Dx \leq d, Ey \leq e \\ x &\in X = \{x | x \in \mathbb{R}^n, x^l \leq x \leq x^u\}, y \in \{0,1\}^q \end{aligned} \quad (2)$$

이 모델들에 대한 해결 알고리즘으로는 GBD (Generalized Benders Decomposition), GOA (Generalized Outer Approximation), 그리고 GCD (Generalized Cross Decomposition) 등이 있으며, 모두 MILP 를 풀기 위해 branch and bound 방법을 사용한다. 다기간 최적화 문제를 정식화할 경우, 각 기간마다 매개변수 값들의 범위와 방정식을 이루는 계수의 값이 달라지므로, 목적함수는 기간, 장소 등의 인덱스를 가지는 식들의 총합의 형태를 띠게 된다.

2. GDP 정식화

MINLP 로 정식화된 모델은 내재적인 문제점을 안고 있다. 공정 합성에서 자주 사용되는 공정도를 구성할 때, 한가지 목적을 위해 여러 대안들도 모두 포함하게 된다. 이 경우 각 대안에 대한 제한식들이 잇따르게 되는데, MINLP 의 경우 선택되지 않은 대안에 종속되는 제한식들을 원래의 모델에서 고려하지 않도록 완전히 배제하는 방법이 존재하지 않는다. 이러한 예는 이진 변수를 포함하는 공정 흐름도의 구조 최적화 문제에서 쉽게 볼 수 있는데, 이것은 자칫 특이점을 야기할 수도 있다[12]. 이것을 극복하기 위해 제안된 것이 GDP 이다. 공정 합성에서 나타나는 GDP 의 일반적인 형태는 다음과 같다.

$$\begin{aligned} \min_{x,Y} Z &= \sum_{k \in K} c_k + f(x) \\ \text{s.t. } r(x) &\leq 0, \bigvee_{j \in J_k} \begin{bmatrix} Y_{jk} \\ g_{jk}(x) \leq 0 \\ c_k = \gamma_{jk} \end{bmatrix}, k \in K \\ \Omega(Y) &= \bigwedge_{s=1,2,\dots,S} \left[\bigvee_{(j,k) \in P_s} (Y_{jk}) \bigvee_{(j,k) \in Q_s} (\neg Y_{jk}) \right] = \text{True} \\ x &\geq 0, x \in \mathbb{R}^n, c_k \geq 0, Y_{jk} \in \{\text{True}, \text{False}\} \end{aligned} \quad (3)$$

모델을 살펴보면, 불린 변수인 Y_{jk} 의 값이 참일 때, $g_{jk}(x) \leq 0$ 과 $c_k = \gamma_k$ 가 강제된다. 그렇지 않으면 그에 상응하는 제약조건들은 무시된다. CNF (Conjunctive Normal Form)으로 표현된 논리식에서 P_s 는 참인 Y_{jk} 의 집합이고, Q_s 는 거짓인 Y_{jk} 의 집합이다.

3. Hybrid MINLP/GDP 정식화

MINLP 와 GDP 는 모델의 대상에 대한 지식만 있으면, 서로간에 바꿔 쓸 수 있다. 하지만 각각의 모델만으로 모든 문제를 자연스럽게 표현하기에는 한계가 있다. Vecchiotti 와 Grossmann[9]은 이진 변수와 불린 변수들을 모두 아우를 수 있는 MINLP 와 GDP hybrid 모델을 제시한 바 있다. 그 형태는 다음과 같다.

$$\begin{aligned}
 \min_{x,y,Y} Z &= \sum_{k \in K} c_k + f(x) + d^T y \\
 \text{s.t. } &g(x) \leq 0, r(x) + Dy \leq 0, Ay \geq a \\
 &\begin{pmatrix} Y_k \\ h_k(x) \leq 0 \\ c_k = \gamma_k \end{pmatrix} \vee \begin{pmatrix} -Y_k \\ B^k x = 0 \\ c_k = 0 \end{pmatrix} \quad (4) \\
 \Omega(Y) &= \text{True}, x \in \mathbb{R}^n, y \in \{0,1\}^q \\
 c_k &\geq 0, Y \in \{\text{True, False}\}^m
 \end{aligned}$$

III. 시스템의 구축

1. 코딩 방법 및 문제의 구성

다기간 최적화 문제는 이진 변수들, 불린 변수들, 그리고 상한과 하한값을 가지는 실수 변수들이 뒤섞여 있는 경우가 대부분이다. 불린 변수들은 0-1 의 이진 변수로 적당히 바꿔 쓸 수 있으며, 이들로 표현되는 논리식들은 표 1 과 다음의 규칙들을 바탕으로 방정식 혹은 부등식으로 바꿔 표현할 수 있다[13].

① Implication 은 그에 상응하는 disjunction 으로 바꾼다.

$$P_1 \Rightarrow P_2 \Leftrightarrow \neg P_1 \vee P_2 \quad (5)$$

② DeMorgan 의 정리를 이용하여, negation 은 최대한 안쪽에 위치시킨다.

$$\begin{aligned}
 \neg(P_1 \wedge P_2) &\Leftrightarrow \neg P_1 \vee \neg P_2 \\
 \neg(P_1 \vee P_2) &\Leftrightarrow \neg P_1 \wedge \neg P_2
 \end{aligned} \quad (6)$$

③ 다음의 등가를 이용하여 “OR” 연산자를 “AND” 연산자 안쪽으로 위치시킨다.

$$(P_1 \wedge P_2) \vee P_3 \Leftrightarrow (P_1 \vee P_3) \wedge (P_2 \vee P_3) \quad (7)$$

이렇게 하여 얻어진 이진 변수와 실수 변수가 최적화하고자 하는 변수의 집합을 이룬다. 이처럼 변수들이 섞여 있는 경우, 한 가지의 코딩으로 변수들을 표현하고자 하는 것은 바람직하지 못한 방법이다. 본 시스템에서는 실제 변수들을 코딩하지 않고 그대로 사용하였다. 이렇게 함으로써 이진 변수는 이진 코딩을 한 것처럼 사용할 수 있었고, 실수 변수는 실수 코딩을 한 것처럼 사용할 수 있었다. 또한 인코

딩과 디코딩에 걸리는 시간을 다른 더 중요한 유전 연산자들에 배분할 수 있었다.

표 1. 논리식들의 제약조건 표현.

Table 1. Constraint Representation of Logic Propositions and Operators.

Logical Relation	Comments	Logical/Boolean Expression	Representation as Linear Equal/Inequalities
Logical “OR”		$P_1 \vee P_2 \vee \dots \vee P_r$	$y_1 + y_2 + \dots + y_r \geq 1$
Logical “AND”		$P_1 \wedge P_2 \wedge \dots \wedge P_r$	$y_1 \geq 1; y_2 \geq 1; \dots; y_r \geq 1$
Implication	$P_1 \Rightarrow P_2$	$\neg P_1 \vee P_2$	$1 - y_1 + y_2 \geq 1$ or $y_1 - y_2 \leq 0$
Equivalence	P_1 if and only if P_2 $(P_1 \Rightarrow P_2) \wedge (P_2 \Rightarrow P_1)$	$(\neg P_1 \vee P_2) \wedge (\neg P_2 \vee P_1)$	$y_1 - y_2 \leq 0; y_2 - y_1 \leq 0$ or $y_1 = y_2$
Exclusive OR(EOR)	Exactly one of the variables is true	$P_1 \oplus P_2 \oplus \dots \oplus P_r$	$y_1 + y_2 + \dots + y_r = 1$
Classification	$Q = \{P_1, P_2, \dots, P_r\}$ Q is true if any of the variables inside the bracket are true		$y_q = y_1 + y_2 + \dots + y_r$

2. 교배 및 변이 연산 확률의 적응

교배 연산자가 탐색에서 기본적으로 가지는 의미는 이미 가지고 있는 정보의 이용이다. 이에 반해 변이 연산자는 새로운 정보의 탐사의 의미를 가진다. 초반에는 탐색 공간에 대한 탐사가 중시되므로, 교배 확률은 낮은 수준에서 유지시키고, 변이 확률을 높이 잡아준다. 하지만 탐색이 계속되고 세대가 진행될수록, 새로운 영역의 탐사보다는 이미 가지고 있는 정보의 이용이 더 중요해지게 되므로, 교배 확률은 조금씩 높아나가고, 변이 확률은 조금씩 낮추어간다. 이런 아이디어를 바탕으로 본 시스템에서는 다음의 간단한 확률 적응을 채택하였다.

$$\begin{aligned}
 P_c(i) &= P_{c0} + (1 - P_{c0}) \times \frac{i}{\max_gen} \\
 P_m(i) &= P_{m0} \times \left(1 - \frac{i}{\max_gen} \right)
 \end{aligned} \quad (8)$$

이때, $P_c(i)$, $P_m(i)$ 는 각각 i 세대의 교배 연산과 변이 연산 확률이고, P_{c0} , P_{m0} 는 각각 초기에 주어지는 교배 연산과 변이 연산 확률이다. \max_gen 은 최종 세대수를 의미한다. (8)로부터 세대가 지날수록 교배 연산 확률은 1로, 변이 연산 확률은 0으로 적용되어감을 쉽게 알 수 있다. 이 최종 적응값은 다른 상수를 도입하여 변경시킬 수도 있다.

3. 다극 탐색기법

본 시스템에서는 다극 탐색 기법으로서 다중 niche 군집 방법을 채택하였다. 다중 niche 군집 방법을 채택하여 FPR 선택압을 줄임으로써 적합도 스케일링이 불필요해졌고, 연속 교체 전략을 사용함으로써 정예 선택을 하지 않아도 뛰어난 개체들이 쉽게 제거되지 않게 되었다. 하지만 이 방법은 많은 메모리와 연산 시간을 필요로 하였다.

4. 국지최적화 방법

본 시스템에서 사용한 국지 최적화 기법은 의사 풀림

로, 유전 알고리즘의 각 세대마다 적당한 반복 횟수만큼의 사 풀림 기법이 적용되었다. 이때 국지 최적화를 겪을 개체들로는 적합도가 높은 상위의 일정 비율의 개체들을 선별하였다. 또한 국지 최적화된 결과는 Lamarckian 방법으로 대체되었다.

5. 제약조건의 처리

본 시스템에서 제약조건을 처리하기 위해 사용한 방법은 동적 penalty 방법이다. 사용된 벌칙항의 형태는 기본적으로 다음의 형태에 기반을 두었다.

$$f' = f - f_{avg} \times penalty$$

$$penalty = \exp\left(\max\left(\frac{\max_gen}{ref_gen}, gen - \frac{\max_gen}{ref_gen}\right)\right) \times \left(\sum_{i=1}^{no_h} \Phi_{hi} + \sum_{i=1}^{no_g} \Phi_{gi}\right) \quad (9)$$

여기에서 f는 음이 아닌 값을 갖는 최대화 문제로 변형한 목적 함수의 값을 나타내고, f_{avg} 는 그 세대 개체군의 f의 평균 값, f'는 벌칙항을 도입하였을 때의 적합도를 나타낸다. gen은 현재 세대수, max_gen은 종료 세대수, ref_gen은 벌칙항의 크기를 결정하는 매개변수이다. 이 값은 벌칙항이 상대적으로 작은 초기에, 개체군이 목적함수 값 자체는 높지만 실현 불가능한 개체들로 조기 수렴되는 일이 없도록 적절히 잡아주어야 한다. 벌칙항의 앞부분인 지수 함수 부분은 벌칙항의 세대수에 대한 적응을 나타낸다. 어느 정도까지의 세대수까지는 적당한 수준의 벌칙항이 적용되다가, 어느 정도 이상의 세대수에서는 점점 벌칙항이 커져서, 결국 중국에는 제약조건을 만족시키지 않는 개체들을 모두 제외할 수 있게 된다. 각각 등식과 부등식 제약조건의 위반정도를 나타내는 함수 Φ_{hi} 와 Φ_{gi} 는 (10)과 같이 구성하였다.

$$\Phi_{hi} = \begin{cases} \frac{h_i}{h_{i,min}}, & \text{if } h_i < 0, h_{i,min} \neq 0 \\ 0, & \text{if } (h_i < 0, h_{i,min} = 0) \vee (h_i > 0, h_{i,max} = 0) \\ \frac{h_i}{h_{i,max}}, & \text{if } h_i > 0, h_{i,max} \neq 0 \end{cases} \quad (10)$$

$$\Phi_{gi} = \begin{cases} 0, & \text{if } g_{i,max} = 0 \\ \frac{\max(g_i, 0)}{g_{i,max}}, & \text{if } g_{i,max} \neq 0 \end{cases}$$

여기에서 $h_{i,min}$ 은 등식 제약조건 i에 대해 그 개체군 내에서 개체가 가지는 가장 작은 음수값을 말하고, $h_{i,max}$ 는 등식 제약조건 i에 대해 그 개체군 내에서 개체가 가지는 가장 큰 양수값을 말한다. 또 $g_{i,max}$ 는 부등식 제약조건 i에 대해 그 개체군 내에서 개체가 가지는 가장 큰 값을 일컫는다. 식 (10)에서 나타나는 g와 h에는 불린 변수들에 종속되거나 종속되지 않는 등식 및 부등식 제약조건들이 모두 포함되어 있다. 이 경우 그 종속성에 따라 0 혹은 적당한 값이 g와 h에 각각 부여되게 된다. 또한 강제되어야 할 논리식은 그 논리

식을 나타내는 제약조건 항을 여러 번 중복해서 더해서 그 중요성을 높일 수 있게 시스템을 설계하였다. 또한 본 연구에서는 이들 선형 논리식의 중요성을 높이기 위한 또 다른 방식으로 Schoenauer와 Xanthakis[14]가 제안한 방법을 변형하여 적용해 보았다.

- ① 무작위로 생성된 초기 개체군으로부터 출발.
- ② j=1 (j는 제약조건 counter)
- ③ j 번째로 강제되어야 하는 선형 논리식 제약조건을 만족시키는 개체들이 총 개체군의 일정 비율을 넘을 때 (소위 flip threshold)까지 개체군을 진화시킴. 이때 1, 2, ..., j-1 번째의 논리식 제약조건 중 하나라도 만족시키지 못하는 개체는 제거함.
- ④ j가 강제되어야 하는 선형 논리식 제약조건의 수보다 작으면, j=j+1로 놓고, c를 반복.
- ⑤ j가 강제되어야 하는 선형 논리식 제약조건의 수와 같으면 목적함수를 최적화하는 값을 찾을 때까지 진화시킴. 이때 이들 제약조건을 만족시키지 못하는 개체들은 제거함.

본 연구에서는 ②에서 ④까지의 절차를 매 세대마다 반복함으로써 선형 논리식 제약조건을 개체군의 어느 비율 이상에게 강제해 보았다. 하지만 이 경우 너무 많은 시간이 소요되고, 특별한 경우를 제외하고는 크게 효율을 볼 수 없었으므로, 테스트만 해 보았을 뿐 실제 문제에는 적용하지 않았다.

6. 제안된 알고리즘

앞에서 설명된 기법들을 채택함으로써 시스템을 구축하는데 사용된 알고리즘을 도식화해 보면 그림 1과 같다.

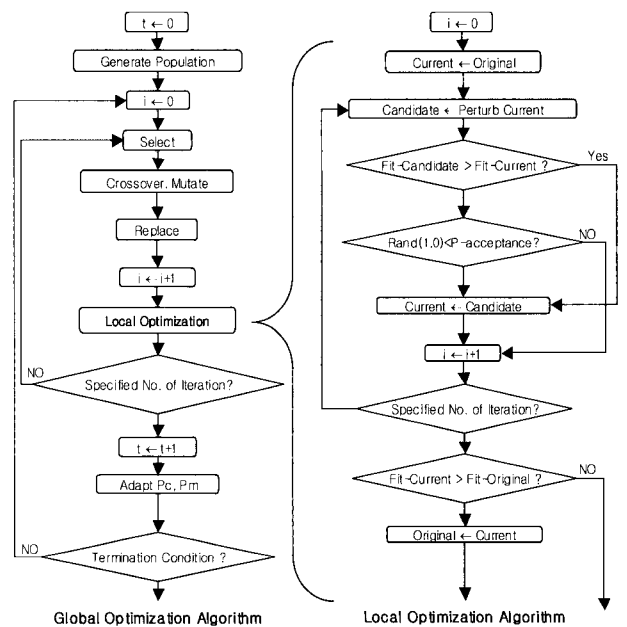


그림 1. 제안된 알고리즘.
Fig. 1. The proposed algorithm.

IV. 적용 예제

1. MINLP 테스트 예제

예제 1 : (원유 수송배관 설계 [15])

서로 다른 유량과 성질을 가진 n 개의 원유를 수송하기 위해 그림 2 과 같이 수평으로 놓이 새관을 설계 한다. 원유의 시간 할당과 최소 출구 압력이 주어질 때 연간 총 비용을 최소화하는 배관 지름 (u), 펌프의 출력 (v) 을 결정해야 한다.

난류라고 가정하고, Fanning friction factor $f=0.04/Re^{0.16}$ 이라고 하면, 각 crude 에 대해 필요한 마력 bHPⁱ 는 다음 식과 같이 주어진다.

$$bHP^i = \frac{Q_i^i}{33000\eta_i} \left[\frac{144(P^i - P_i^i)}{\rho_i} + \frac{K_i^i}{D_i^{4.84}} \right], \quad i = 1, 2, 3, 4 \quad (11)$$

또한,

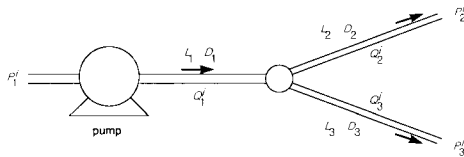


그림 2. 다중 원유 수송배관도.

Fig. 2. Illustration of multi-crude pipeline.

$$K_j^i = 0.1076675(Q_j^i)^{1.84} L_j \left(\frac{\mu_j}{\rho_j} \right)^{0.16}, \quad i = 1, 2, 3, 4; j = 1, 2, 3 \quad (12)$$

$$P_j^i = \frac{\rho_j}{144} \left[\frac{144P^i}{\rho_j} - \frac{K_j^i}{D_j^{4.84}} \right], \quad i = 1, 2, 3, 4; j = 2, 3 \quad (13)$$

여기에서 Q 는 유량 (ft³/min), L 은 파이프의 길이 (ft), D 는 파이프의 지름 (in.), ρ는 밀도 (lb/ft³), μ는 점도 (cP), η는 펌프의 효율, P 는 압력 (psia)이다. 펌프의 마력과 출구 압력은 다음 제약조건을 만족해야 한다.

$$\overline{bHP} \geq bHP^i, \quad i = 1, 2, 3, 4 \quad (14)$$

$$P_j^i \geq P_m^j, \quad i = 1, 2, 3, 4; j = 2, 3 \quad (15)$$

여기에서 P_m 은 출구에서 설정된 최소 압력이다. 정리하면 다음과 같은 문제를 얻을 수 있다.

$$\begin{aligned} \min OBJ &= \sum_{j=1}^3 c_D L_j D_j + c_B \overline{bHP}^{1.6} + \sum_{i=1}^4 C_{Ri} \left[\frac{144(P^i - P_i^i)}{\rho_i} + \frac{K_i^i}{D_i^{4.84}} \right] \\ \text{s.t. } P_m^j &\leq \frac{\rho_j}{144} \left[\frac{144P^i}{\rho_j} - \frac{K_j^i}{D_j^{4.84}} \right], \quad i = 1, 2, 3, 4; j = 2, 3 \\ \overline{bHP} &\geq \frac{Q_i^i \rho_i}{33000\eta_i} \left[\frac{144(P^i - P_i^i)}{\rho_i} + \frac{K_i^i}{D_i^{4.84}} \right], \quad i = 1, 2, 3, 4 \\ D_j^i &\leq D_j \leq D_j^i, \quad j = 1, 2, 3 \end{aligned} \quad (16)$$

여기에서,

$$C_{Ri} = \frac{Q_i^i \rho_i c_e}{44253\eta_i}, \quad i = 1, 2, 3, 4 \quad (17)$$

θ_i는 crude i 에 대해 1년 중 할당된 시간 (hr)이고, c_e는 전

력 비용 (\$/kWh)이며, D_j^L, D_j^U는 각각 파이프 지름의 하한, 상한값 (in.)이다. c_D, c_B, γ는 비용 상수이다.

결과

GA/SA 파라미터 : Population Size: 100
 Generation Number: 200
 Number of runs: 10
 SA Iteration Number: 50

최적해 : bHP=301.809
 D=(26.392, 15.221, 16)
 P=(105.582, 102.177, 105.616, 101.941)
 최적값=129891 \$
 문헌값 : bHP=309.1 ← 450
 D=(24.6, 15.6, 16.0) ← (28, 14, 14)
 P=(105.6, 102.2, 105.6, 101.9) ← (110, 110, 110, 110)
 최적값=130555 \$

결과에서 볼 수 있듯이 제안된 알고리즘으로 문헌치 보다 약간 우수한 해를 구할 수 있었다.

예제 2

(다중생산 공정설계 [16])

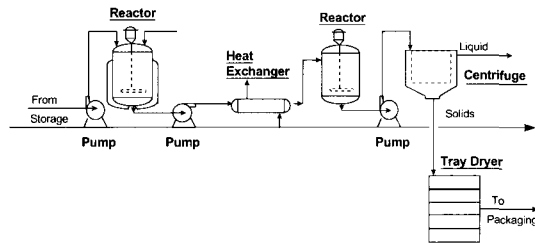


그림 3. 다중생산 공정도.

Fig. 3. Illustration of multi-product process.

그림 3의 공정은 세 개의 회분식 공정과 다섯 개의 연속 공정으로 이루어져 있고 1년 (8000 시간)에 걸쳐 세 가지의 제품을 생산하는 공정이다. 공정에 대한 자세한 설명과 필요한 데이터는 참고문헌 [16]를 참조하기 바란다.

공정 설계의 목적은 비용이 최소가 되도록 장치의 크기를 결정하는 것으로 최적화 문제는 다음과 같은 형태로 주어진다.

$$\begin{aligned} \min OBJ &= a_1 V_1^{\alpha_1} + a_2 V_2^{\alpha_2} + a_3 V_3^{\alpha_3} + b_1 R_1^{\beta_1} + b_2 R_2^{\beta_2} + b_3 R_3^{\beta_3} + b_4 R_4^{\beta_4} + b_5 R_5^{\beta_5} \\ \text{s.t. } 8000 &\geq \frac{400000}{R_1} T_1 + \frac{300000}{R_2} T_2 + \frac{100000}{R_3} T_3 \\ V_1 &\geq 1.2B_1, \quad V_1 \geq 1.5B_2, \quad V_1 \geq 1.1B_3 \\ V_2 &\geq 1.4B_1, \quad V_2 \geq 1.2B_3 \\ V_3 &\geq 1.0B_1, \quad V_3 \geq 1.0B_2, \quad V_3 \geq 1.0B_3 \\ R_3 &\geq R_2, \quad R_3 \geq R_4, \quad R_3 \geq R_5 \\ T_1 &\geq \frac{1.2B_1}{R_1} + 3 + \frac{1.2B_1}{R_2}, \quad T_1 \geq \frac{1.2B_1}{R_3} + 1 + \frac{1.4B_1}{R_4}, \quad T_1 \geq \frac{1.4B_1}{R_5} + 4 \\ T_2 &\geq \frac{1.5B_2}{R_1} + 6 + \frac{1.5B_2}{R_2}, \quad T_2 \geq \frac{1.5B_2}{R_3} + 8 \\ T_3 &\geq \frac{1.1B_3}{R_1} + 2 + \frac{1.1B_3}{R_2}, \quad T_3 \geq \frac{1.1B_3}{R_3} + 2 + \frac{1.2B_3}{R_4}, \quad T_3 \geq \frac{1.2B_3}{R_5} + 4 \end{aligned} \quad (18)$$

결과

GA/SA 파라미터 : Population Size: 200
 Generation Number: 100
 Number of runs: 10
 SA Iteration Number: 50

최적해 : $V=(1262.31, 1285.66, 913.035)$
 $R=(1301.08, 361.040, 388.819, 413.024, 413.069)$
 $B=(912.410, 840.854, 909.898)$
 $T=(7.13369, 11.4213, 7.07147)$
 최적값=157789 \$

문헌값 : $V=(1181.4, 1250.6, 893.3) \leftarrow (2000, 2000, 2000)$
 $R=(753.1, 422.1, 422.1, 422.1, 422.1) \leftarrow (1000, 1000, 1000, 1000, 1000)$
 $B=(893.3, 787.6, 892.9) \leftarrow (2000, 2000, 2000, 2000)$
 $T=(6.693, 10.799, 6.865) \leftarrow (5.0, 5.0, 5.0)$
 최적값=159483 \$

역시 문헌값보다 우수한 해를 구할 수 있었다. 위 두 예제에 대하여 박과 윤[17]에 의해 개발된 APROGA 시스템, 이와 윤[18]의 APROGA-II 시스템의 결과와 비교한 결과를 표 2에서 보여주고 있다. 본 연구의 방법론이 평균적으로 더 나은 성능을 보임을 확인할 수 있으며, 최적해에 도달하는 데 걸리는 시간도 의사 풀림과의 결합을 통해 상당히 개선할 수 있었다.

표 2. APROGA 시스템과의 비교.

Table 2. Comparison with APROGA systems.

예제	문헌값	APROGA	APROGA-II*	제안연구	% 향상
1	130,555	129,926	129,964 (best) 130,716 (avg)	129,891	0.5
1	159,483	158,004	156,998 (best) 159,877 (avg)	157,789	1

*APROGA-II 시스템의 경우, 3 회 평균값을 비교대상으로 취해짐

2. GDP 테스트 예제

예제 1

이 예제는 결정론적 GDP 알고리즘의 효율성을 시험하기 위해 Lee와 Grossmann[4]이 제시한 바 있다. 제안된 유전 알고리즘이 전역해를 찾고 있음을 확인할 수 있었다.

$$\min Z = (x_1 - 3)^2 + (x_2 - 2)^2 + c$$

s.t.

$$\begin{bmatrix} Y_1 \\ x_1^2 + x_2^2 \leq 1 \\ c=2 \end{bmatrix} \vee \begin{bmatrix} Y_2 \\ (x_1 - 4)^2 + (x_2 - 1)^2 \leq 1 \\ c=1 \end{bmatrix} \vee \begin{bmatrix} Y_3 \\ (x_1 - 2)^2 + (x_2 - 4)^2 \leq 1 \\ c=3 \end{bmatrix} \quad (19)$$

$$0 \leq x_1, x_2 \leq 8; c \geq 0; Y_j \in \{True, False\}, j=1,2,3$$

결과

GA/SA 파라미터 : Population Size: 100
 Generation Number: 100

Number of runs: 10
 SA Iteration Number: 50

최적해 : $Y=(F, T, F)$
 $x=(3.293, 1.707)$
 $Z=1.172$

문헌값 : $Y^*=(F, T, F)$
 $x^*=(3.293, 1.707)$
 $Z^*=1.172$

예제 2

이 문제는 Duran 과 Grossmann[19]에 의해 MINLP 문제로 제시되었으며, Turkey 와 Grossmann[20]에 의해 GDP 문제로 변환되었다. 이 공정의 흐름도는 그림 4 와 같다.

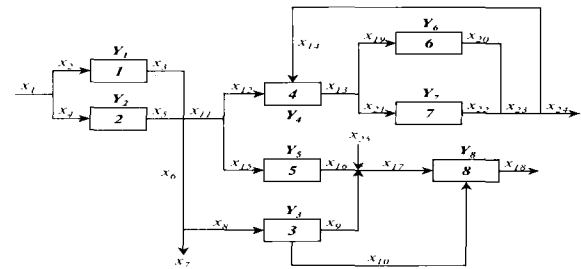


그림 4. 공정 흐름도.

Fig. 4. Process network.

$$\min Z = a^T x + \sum_{i=1}^8 c_i + 122$$

s.t.

$$\begin{aligned} x_1 - x_2 - x_4 &= 0, & x_6 - x_7 - x_8 &= 0, & x_3 + x_5 - x_6 - x_{11} &= 0, & x_{13} - x_{14} - x_{21} &= 0 \\ x_{17} - x_9 - x_{16} - x_{24} &= 0, & x_{11} - x_{12} - x_{15} &= 0, & x_{23} - x_{20} - x_{22} &= 0, & x_{23} - x_{14} - x_{24} &= 0 \\ x_{10} - 0.8x_{17} &\leq 0, & x_{10} - 0.4x_{17} &\geq 0, & x_{12} - 5x_{14} &\leq 0, & x_{12} - 2x_{14} &\geq 0 \end{aligned}$$

$$\begin{bmatrix} Y_1 \\ \exp(x_3) - x_2 - 1 = 0 \\ c_1 = 5 \end{bmatrix} \vee \begin{bmatrix} -Y_1 \\ x_3 = x_2 = 0 \\ c_1 = 0 \end{bmatrix}$$

$$\begin{bmatrix} Y_2 \\ \exp(x_7/1.2) - x_4 - 1 = 0 \\ c_2 = 8 \end{bmatrix} \vee \begin{bmatrix} -Y_2 \\ x_5 = x_4 = 0 \\ c_2 = 0 \end{bmatrix}$$

$$\begin{bmatrix} Y_3 \\ 1.5x_9 - x_8 + x_{10} = 0 \\ c_3 = 6 \end{bmatrix} \vee \begin{bmatrix} -Y_3 \\ x_9 = 0, x_{10} = x_8 \\ c_3 = 0 \end{bmatrix}$$

$$\begin{bmatrix} Y_4 \\ 1.25(x_{12} + x_{14}) - x_{13} = 0 \\ c_4 = 10 \end{bmatrix} \vee \begin{bmatrix} -Y_4 \\ x_{14} = x_{13} = x_{12} = 0 \\ c_4 = 0 \end{bmatrix}$$

$$\begin{bmatrix} Y_5 \\ x_{15} - 2x_{16} = 0 \\ c_5 = 6 \end{bmatrix} \vee \begin{bmatrix} -Y_5 \\ x_{16} = x_{15} = 0 \\ c_5 = 0 \end{bmatrix}$$

$$\begin{bmatrix} Y_6 \\ \exp(x_{20}/1.5) - x_{19} - 1 = 0 \\ c_6 = 7 \end{bmatrix} \vee \begin{bmatrix} -Y_6 \\ x_{20} = x_{19} = 0 \\ c_6 = 0 \end{bmatrix}$$

$$\begin{bmatrix} Y_7 \\ \exp(x_{22}) - x_{21} - 1 = 0 \\ c_7 = 4 \end{bmatrix} \vee \begin{bmatrix} -Y_7 \\ x_{22} = x_{21} = 0 \\ c_7 = 0 \end{bmatrix}$$

$$\begin{bmatrix} Y_8 \\ \exp(x_{18}) - x_{10} - x_{17} - 1 = 0 \\ c_8 = 5 \end{bmatrix} \vee \begin{bmatrix} -Y_8 \\ x_{18} = x_{17} = x_{10} = 0 \\ c_8 = 0 \end{bmatrix}$$

$$a^T = [0, 1, -10, 1, -15, 0, 0, 0, -40, 15, 0, 0, 0, 15, 0, 0, 80, -65, 25, -60, 35, -80, 0, 0, -35]$$

$$x_i, c_j \geq 0, Y_j \in \{True, False\}, i=1,2,\dots,25, j=1,2,\dots,8$$

$$\begin{aligned}
 &Y_1 \Rightarrow Y_1 \vee Y_4 \vee Y_5, \quad Y_2 \Rightarrow Y_2 \vee Y_4 \vee Y_5, \quad Y_3 \Rightarrow Y_3 \vee Y_5, \quad Y_4 \Rightarrow Y_4 \vee Y_5, \quad Y_5 \Rightarrow Y_5 \vee Y_6 \\
 &Y_6 \Rightarrow Y_6, \quad Y_7 \Rightarrow Y_7 \vee Y_8, \quad Y_8 \Rightarrow Y_8, \quad Y_9 \Rightarrow Y_9, \quad Y_{10} \Rightarrow Y_{10} \vee Y_{11} \vee (-Y_{12} \wedge -Y_{13}) \\
 &Y_{14} \vee Y_{15}, \quad Y_{16} \vee Y_{17}, \quad Y_{18} \vee Y_{19}
 \end{aligned}$$

이 문제는 다음과 같이 w-MIP criteria 를 적용하여 3 개의 disjunction 을 방정식으로 바꿀 수 있다. 이 때, 이에 해당하는 3 개의 Boolean 변수들은 binary 변수들로 바뀌게 되고, 그 결과로 얻어지는 문제는 GDP 와 MINLP 가 결합된 형태를 나타내게 된다[21].

$$\begin{aligned}
 &1.5x_9 - x_8 + x_{10} = 0; c_3 = 6y_3 \\
 &1.25(x_{12} + x_{14}) - x_{13} = 0; c_4 = 10y_4 \\
 &x_{15} - 2x_{16} = 0; c_5 = 6y_5 \\
 &y_j \in \{0,1\}, i = 1, 2, \dots, 25, j = 1, 2, \dots, 8 \\
 &-y_1 + y_3 + y_4 + y_5 \geq 0, \quad -y_2 + y_3 + y_4 + y_5 \geq 0 \\
 &y_1 + y_2 - y_3 \geq 0, \quad y_1 + y_2 - y_4 \geq 0, \quad y_1 + y_2 - y_5 \geq 0 \\
 &-y_3 + y_8 \geq 0, \quad -y_5 + y_8 \geq 0 \\
 &-y_4 + y_6 + y_7 \geq 0, \quad y_4 - y_6 \geq 0, \quad y_4 - y_7 \geq 0 \\
 &y_1 + y_2 = 1, \quad y_3 + y_4 = 1, \quad y_6 + y_7 = 1
 \end{aligned}$$

결과

GA/SA 파라미터 : Population Size: 500

Generation Number: 100

Number of runs: 30

SA Iteration Number: 50

최적해 : $Y=(F, T, F, T, F, T, F, T)$

$X=(4.428, 0, 0, 4.428, 2.03, 0.787, 0.444, 0.343, 0, 0.343, 1.243, 1.243, 2.011, 0.366, 0, 0, 0.598, 0.663, 2.011, 1.653, 0, 0, 1.653, 1.287, 0.598)$

$Z=71.485$

문헌값 : $Y^*=(F, T, F, T, F, T, F, T)$

$X^*=(4.294, 0, 0, 4.294, 2, 0.667, 0.203, 0.464, 0, 0.464, 1.333, 1.333, 2, 0.267, 0, 0, 0.58, 0.715, 2, 1.648, 0, 0, 1.648, 1.381, 0.58)$

$Z^*=68.01$

제안된 알고리즘으로 구해진 해는 결정론적 알고리즘으로 얻어진 전역해의 품질에 약간 못 미치지만, 동일한 설계변수의 값을 가지며, 연속 변수 값들도 많이 근접해 있음을 확인할 수 있다.

V. 결론

본 연구에서는 불린 변수와 이진 변수가 연속 변수와 섞여 있는 최적화 문제의 일반적인 형태를 유전 알고리즘과 의사 풀림의 기법을 사용하여 다루고자 하였다. 이러한 형태의 문제들은 특히 다기간 최적화 문제에서 많이 살펴볼 수 있으며, 가까이는 공정 합성에서도 쉽게 그 예를 찾을 수 있다. 그 방법론으로서 유전 알고리즘과 의사 풀림을 선택하게 된 동기는, 이들 최적화 기법들이 여러 형태의 문제로의 적용성이 뛰어나기 관찰되어 왔기 때문이다. 이런 복합 형태의 문제를 해결하기 위해 많은 방법들이 시도되었다. 먼저 유전 알고리즘과 의사 풀림을 결합함으로써 각자의 장점인 전역 탐색과 국지탐색 효과로 서로의 단점을 상호 보완하고자 하였다. 제약조건의 처리나 다극 탐색, 국지 최적화를 위한 유

전 알고리즘과 의사 풀림의 제반 이론들이 적용되었고, 문제의 형태에 가장 적합한 코딩 방법이 제안되었으며, 논리적으로 연결되어 있는 제약조건의 처리 문제 등 기타 많은 경험적인 기법들이 고안되었다. 그 결과로 구축된 시스템은 기존의 MINLP 문제와 최근 새롭게 제안되고 있는 GDP 문제에 대해 적용되었다. 사례 적용결과 MINLP 의 경우 문헌값보다 향상된 결과를 얻을 수 있었으며, 다른 방법과의 비교에서도 더 나은 성능을 보여줌을 확인 할 수 있었다. GDP 의 경우 변수의 개수가 늘어나고 불린 변수와 이진 변수에 종속되는 연속 변수와 제약조건의 수가 늘어나고 복잡해짐에 따라 결과값이 조금씩 전역해에 못 미침이 관찰되었다. 이 경우에는 w-MIP 조건을 통해 MINLP/GDP 의 복합 문제로 바꿈으로써 그 상황을 개선할 수 있었다. 또한 Big-M 조건으로 변환함으로써 엄격한 조건으로 인한 특이점 문제를 해결할 수 있음도 관찰되었다. 그러나 여전히 전역해로의 도달은 더 개선되어야 할 문제의 하나로 남아있다. 이 문제는 좀더 여러 경우에 대해 검증된 방법론들의 사용을 통해 해결이 가능할 것으로 사료된다.

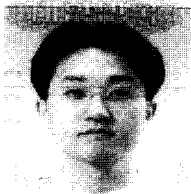
참고문헌

- [1] D. H. Wolpert and W. G. Macready, "No free lunch theorems for search," Santa Fe Institute, Santa Fe, NM, Tech. Rep. SFI-TR-05-010, 1995.
- [2] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Programming*, Vol. 1, No. 1, pp. 67-82, 1997.
- [3] J. M. Douglas, "A hierarchical decision procedure for process synthesis," *AIChE*, Vol. 31, No. 3, pp. 353-362, 1985.
- [4] S. Lee and I. E. Grossmann, "New algorithms for nonlinear generalized disjunctive programming," *Computers and Chemical Engineering*, Vol. 24, No. 9, pp. 2125-2141, 2000.
- [5] G. R. Kocis and I. E. Grossmann, "Global optimization of nonconvex mixed-integer nonlinear programming (MINLP) problems in process synthesis," *Industrial and Engineering Chemistry Research*, **27**, pp. 1407-1421, 1988.
- [6] G. R. Kocis and I. E. Grossmann, "A modeling and decomposition strategy for the MINLP optimization of process flowsheets," *Computers and Chemical Engineering*, Vol. 13, No. 7, pp. 797-819, 1989.
- [7] J. H. Holland, "Adaptation in natural and artificial systems," University of Michigan Press, Ann Arbor, 1975.
- [8] D. E. Goldberg, "Genetic algorithms in search, optimization and machine learning," Addison-Wesley, Reading, MA, 1989.
- [9] S. Kirkpatrick, C. D. Gelatt, Jr., and M. P. Vecchi, "Optimization by simulated annealing," *Science*, **220**, pp. 671-680, 1983.
- [10] D. Sirag and P. Weisser, "Toward a unified thermodynamic genetic operator," *Proceedings of the 2nd International Conference on Genetic Algorithms*, pp. 116-122, 1987.
- [11] D. Adler, "Genetic algorithms and simulated annealing: a marriage proposal," *IEEE International Conference on Neural Networks*, **2**, pp. 1104-1109, 1993.
- [12] A. Vecchiotti and I. E. Grossmann, "LOGMIP: a disjunctive 0-1 non-linear optimizer for process systems models," *Computers and Chemical Engineering*, **23**(4-5), pp. 555-565, 1999.
- [13] R. Raman and I. E. Grossmann, "Relation between MILP

modeling and logical inference for chemical process synthesis," *Computers and Chemical Engineering*, **15**(2), pp. 73-84, 1991.

- [14] M. Schoenauer and S. Xanthakis, "Constrained GA optimization," *Proceedings of the Fifth International Conference on Genetic Algorithms*, Morgan Kaufmann, San Mateo, CA, pp.573-580, 1993.
- [15] I. E. Grossmann and R. W. H. Sargent, "Optimum design of multipurpose chemical plants", *Industrial and Engineering Chemistry Process Design and Development*, Vol. 18, No. 2, pp. 343-348, 1979.
- [16] V. Reklaitis, A. Ravindran and K. M. Ragsdell, "Engineering optimization: methods and applications," Wiley-Interscience, 1983.
- [17] 박성준, 윤인섭, "개선된 유전 알고리즘을 이용한 다목적 공정의 최적화에 관한 연구", 서울대학교 응용화학부 석사학위논문, 1997.

- 석사학위논문, 1997.
- [18] 이원보, 윤인섭, "유전 알고리즘을 이용한 다극 및 다목적함수 공정최적화", 서울대학교 응용화학부 석사학위논문, 1998.
- [19] M. A. Duran and I. E. Grossmann, "An outer-approximation algorithm for a class of Mixed-integer nonlinear programs," *Mathematical Programming*, **36**, pp. 307-339, 1986.
- [20] M. Turkey and I. E. Grossmann, "Logic-based MINLP algorithms for the optimal synthesis of process networks," *Computers and Chemical Engineering*, **20**(8), pp. 959-978, 1996.
- [21] A. Vecchiotti and I. E. Grossmann, "LOGMP: a disjunctive 0-1 non-linear optimizer for process systems models," *Computers and Chemical Engineering*, **23**(4-5), pp.555-565, 1999.



송 상 옥

1997년 서울대학교 응용화학부 졸업. 1997년~현재 서울대학교 응용화학부 대학원 재학중. 관심분야는 공정최적화, 데이터 마이닝, 이상진단.



장 영 중

1997년 서울대학교 응용화학부 졸업. 2001년 서울대학교 응용화학부 석사학위 취득. 현재 UIUC 화학공학과 박사과정. 관심분야는 공정최적화.



김 구 회

1992년 광운대학교 화학공학과 (공학사). 2000년 서울대학교 응용화학부 (공학박사). 현재 서울대학교 화학공정신기술연구소 특별연구원. 관심분야는 공정시스템 및 안전.



윤 인 섭

1949년 6월 4일생. 1971년 서울대학교 화학공학과 (공학사). 1982년 MIT 화학공학과 (공학박사). 1983년~현재 서울대학교 응용화학부 교수. 관심분야는 공정시스템 및 안전.