Development of Real Coded Genetic Algorithm for Multiperiod Optimization

YoungJung Chang*, Sang Ok Song, Ji Ho Song, Dongil Shin and S.Ando**

* School of Chemical Engineering, Seoul National University Kwanak Ku, Seoul, 151-742 Korea
(Tel : 82-2-880-1581; Fax : 82-2-872-1581 ; E-mail: genjaf@pslab.snu.ac.kr)
** School of Chemical Engineering, Seoul National University Kwanak Ku, Seoul, 151-742 Korea
(Tel : 82-2-873-2605; Fax : 82-2-884-0530 ; E-mail: esyoon@pslab.snu.ac.kr)

Abstract

Multiperiod optimization is the key step to tackle the supply chain optimization problems. Taking supply and demand uncertainty or prediction into consideration during the process synthesis phase leads to the maximiziation of the profit for the long range time horizon. In this study, new algorithm based on the Genetic Algorithms is proposed for multiperiod optimization formulated in MINLP, GDP and hybrid MINLP/GDP. In this study, the focus is given 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 tried, and many heuristics are adopted for this purpose.

1. Introduction

Multiperiod optimization has been formulated in the form of MINLP (Mixed Integer Nonlinear Programming), so the solution strategies for the multiperiod optimization have been in majority based on the MINLP solution algorithms. But when the integer variables can be reduced to the binary or Boolean variables, and the system has a lot of constraints that are subject to those variables, the solution procedure is susceptible to the singularity problem. GDP (Generalized Disjunctive Programming) was proposed as an alternative formulation scheme to handle that problem. MINLP and GDP are interchangeable with constraint relaxation techniques like big-M constraint and w-MIP condition. GDP is also a good scheme to include the logically expressed heuristics within mathematical programming. The majority of previously proposed solution algorithms for GDP are based on the deterministic Branch and Bound method.

Genetic Algorithms and Simulated Annealing are representative random search methods that are used extensively for the various optimization practices. When compared to each other, GA is known to have much probability to search the entire space with the innate parallelism, so that it can easily escape from the local optimum, and the SA has the ability of fine-tuning and hill-climbing nature within the small radius of consideration. So the hybridization of two methods is thought to show the synergy effect, and there have been many attempts to combine them. Examples can be found in the previous work of Sirag and Weisser [1], and Adler's work [2].

2. Motivation

Originally, GA was developed as a tool for handling combinatorial optimization. Traditional simple GA is well suited to this application in its binary encoding and decoding system, and its operators. So it had different originality from Evolutionary Strategy (ES) which uses real parameters and mutation operator only. But in the process of sharing their techniques, GA and ES became similar to each other, so that in present time, it is very hard to tell one from the other.

The encoding of real parameters in GA was accomplished by discretizing the real parameters' upper and lower bounds with predefined resolution (step size). The resolution should be selected carefully. It cannot be too small because as gene length grows, the needed population size and termination generation should also grow. Also, the introduction of truncation error below the resolution is innate. To overcome this shortcoming, the real-coded GA was proposed for the function optimization. This real-coded GA also uses crossover and mutation operators modified to handle real-coded genes.

The problems arising in the field of chemical process synthesis have both binary and real variables. MINLP has integer variables (usually binary variables) and GDP has Boolean variables. To handle these variables within the same framework of binary encoding or real encoding may cause the loss of good building blocks during the operation. To resolve this situation, the GA with non-coding is proposed in this study, and SA is hybridized as a local optimization technique for the fast convergence and fine-tuning.

3. Genetic Algorithms

Over the last decades, Genetic Algorithms [3] have emerged as a leading tool for optimization of arbitrary functions and for guided search problems in high dimensional spaces. The major theories supporting the GA's performance are schemata theorem and building block hypothesis.

Schemata Survival Probability

When using the binary encoding, one-point crossover and allele-wise mutation, the survival probability of specific schemata H at the next generation can be expressed as:

$$\frac{p(H,t+1)}{p(H,t)} \geq \frac{f(H)}{\bar{f}}\left[1-p_c\frac{\delta(H)}{l-1}\left(1-p(H,t)\right)\right](1-p_m)^{o(H)} \tag{1}$$

Here, p(H, t) represents the proportion of schemata H in the population at time t. f(H) is the average fitness of individuals including schemata H at time t, and f is the average fitness of the population at time t. p_c and p_m are the probability of crossover and mutation operator respectively at time t. l is the length of chromosome, δ(H) is the defining length of schemata H, and o(H) is the order of schemata H.

When using the real encoding, the survival probability of specific interval schemata S at the next generation can be expressed as:

$$\frac{p(S,t+1)}{p(S,t)} \geq \frac{f(S)}{\bar{f}}\left[1-p_c\frac{\delta(S)}{l-1}\left(1-p(S,t)\right)\right](1-p_m)^{o(S)} \tag{2}$$

Operators

The main operators of GA's are crossover and mutation. Crossover has the meaning of exploiting the explored domain, and the mutation has the meaning of exploring the new domain. But when it comes to the multi-point crossover or real coded GA, the difference between the meanings of crossover and mutation becomes obscure and sometimes the interface banishes.