

실수 코딩 유전자 알고리즘을 이용한 생산 시스템의 시뮬레이션 최적화

박 경 종

광주대학교 경영학과

Simulation Optimization of Manufacturing System using Real-coded Genetic Algorithm

Kyoung-Jong Park

Dept. of Business Administration, Gwangju University

In this paper, we optimize simulation model of a manufacturing system using the real-coded genetic algorithm. Because the manufacturing system expressed by simulation model has stochastic process, the objective functions such as the throughput of a manufacturing system or the resource utilization are not optimized by simulation itself. So, in order to solve it, we apply optimization methods such as a genetic algorithm to simulation method. Especially, the genetic algorithm is known to more effective method than other methods to find global optimum, because the genetic algorithm uses entity pools to find the optimum.

In this study, therefore, we apply the real-coded genetic algorithm to simulation optimization of a manufacturing system, which is known to more effective method than the binary-coded genetic algorithm when we optimize the constraint problems. We use the reproduction operator of the applied real-coded genetic algorithm as technique of the remainder stochastic sample with replacement and the crossover operator as the technique of simple crossover. Also, we use the mutation operator as the technique of the dynamic mutation that configures the searching area with generations.

Key-words : Real-coded genetic algorithm, Simulation optimization, Binary-coded genetic algorithm

1. 서 론

시뮬레이션은 분석적인 방법으로 표현할 수 없거나 표현하기 어려운 문제를 컴퓨터상에서 최대한 현실 문제와 비슷하게 모델링하여 분석하는 what-if 분석법중의 하나이다. 시뮬레이션 기법은 주어진 제약 조건에 대한 시뮬레이션 실행 후의 결과 값을 주지만, 어떤 제약 조건일 때 최적의 시뮬레이션 결과 값을 도출하는지 아는 것은 매우 어렵다. 특히 시뮬레이션 결과 값은 확률적 특성을 내포하고 있기 때문에 최적해를 얻어내기가 더욱 어렵다. 이러한 문제점 때문에 고전적인 최적화 기법

들을 시뮬레이션에 적용하여 많이 사용하고 있다(Meketon, 1987; Ho et al., 1993).

시뮬레이션 기법과 같이 사용되는 최적화 기법들 중에서도 유전자 알고리즘은 최적해를 탐색시에 하나의 개체(변수)가 아닌 개체들의 군(pool) 단위로 탐색하기 때문에 다른 방법들에 비해서 전역적 해를 구할 가능성이 높고 효율적이라 알려져 있다. Yunker and Tew(1994)는 시뮬레이션과 유전자 알고리즘을 사용하여 최적화를 수행할 때의 효과에 대한 연구를 수행하였고, Lee et al.(2000)은 조립 라인의 시뮬레이션 최적화에 이진코딩 유전자 알고리즘을 사용하여 유전자 알고리즘이 시뮬레

이션 최적화에 좋은 효율을 보여준다고 설명하였다.

유전자 알고리즘에서 염색체를 코딩할 때 사용되는 이진코딩 유전자 알고리즘은 가장 쉽고 대중적인 반면에 여러 문제점을 때문에 실제 문제에서는 실수코딩 유전자 알고리즘이 보다 효과적이라고 알려져 있다. 실제 문제에서 실수 코딩이 선호되는 이유는 다음과 같다(진강규, 2000).

첫째, 변수의 표현형과 유전자의 유전자형 간에 일대일 일치로 프로그래밍이 간편해지고 이진코딩 유전자 알고리즘에서 요구되는 부호화 및 복호화 프로세스가 필요없게 되어 탐색 속도를 높일 수 있다. 둘째, 지역동조를 통해 해의 정밀도를 개선할 수 있다. 이진코딩 유전자 알고리즘에서는 해명 절벽 때문에 지역동조가 어렵지만 실수코딩 유전자 알고리즘에서는 더 적합하고 빠른 방법으로 동조가 일어나도록 해주는 불균등 돌연변이와 같은 연산자도 있다. 셋째, 해에 관한 사전지식이 없는 경우에는 매우 큰 정의 영역을 정하는 것도 가능하다. 넷째, 적관적으로 문제 공간과 가깝고 관련된 지식을 포함시키는 특별한 연산자들의 구현이 용이하다. 다섯째, 문제 영역과 가까우므로 복잡한 제약 조건을 다루기 위한 도구를 설계하는 것이 쉬워진다.

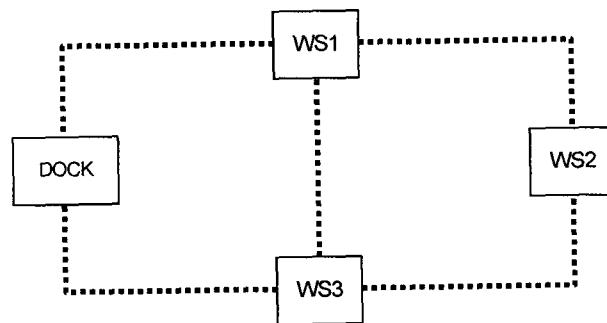
따라서 본 연구에서는 생산 시스템의 시뮬레이션 최적화에 이진코딩 유전자 알고리즘보다 효율적이라고 알려진 실수코딩 유전자 알고리즘을 사용하고 재생산 연산자는 교체를 수반한 잔여화를 분포 선택 방법을 사용한다. 교배 연산자는 단순교배 방법을 사용하고 돌연변이 연산자는 동적 돌연변이 연산자를 사용한다.

2. 시뮬레이션 모델

본 논문에서 사용되는 실험 예제는 Kelton et al.(2002)에서 제시된 모델을 참조한다. 소형생산 시스템에 삼각분포 (6, 13, 19)분의 도착 시간 간격을 가지고 부품들이 도착한다. 모든 부품들은 <그림 1>과 같이 도크로 들어와서 작업장 1, 2, 3을 각각 지나서 최종적으로 다시 도크로 돌아온다. 모든 부품들의 이동은 2대의 카트로 이루어지며 각기 분당 최소 50 피트에서 최대 70피트의 속도로 움직일 수 있다. 도크에서 작업장 1과 3까지의 거리는 각각 50피트이며, 작업장을 사이의 거리도 50 피트이다. 부품들은 작업장 1과 3에서는 하역되어 작업되지만 작업장 2에서는 카트 위에 올려진 상태에서 가공된다. 가공 시간은 작업장 1, 2, 3에서 각각 10-15, 8-11, 8-18 사이에서 변경될 수 있다. 모두 적재 시간과 하역 시간은 무시된다고 가정한다.

<그림 1>에서 부품들은 작업장 1 -> 작업장 2 -> 작업

장 3의 순서로 흘러간다. 이 때 카트는 작업장에서 부품이 흘러가는 작업장의 순서와 상관없이 이동이 가능하다. 즉, 작업장 1 -> 작업장 3으로의 이동이 가능하고 도크 -> 작업장 3으로의 이동도 가능하다.



<그림 1> 생산 시스템의 개념도

본 연구에서 사용되는 시뮬레이션 모델은 시뮬레이션 툴인 ARENA로 모델링을 하며, 유전자 알고리즘은 VBA를 사용하여 코딩한 후 시뮬레이션 모델과 연결하여 최적화 과정을 수행한다.

3. 실수코딩 유전자 알고리즘

유전자 알고리즘은 진화라는 주제에 고무되어 나타난 계산 모델의 한 종류로서 적자생존과 유전의 메커니즘을 바탕으로 하는 탐색 알고리즘이다. 유전자 알고리즘의 구조는 일반적인 진화 알고리즘의 구조와 동일하다. 기존의 최적화 알고리즘과는 다르게 유전자 알고리즘은 재생산 연산자(reproduction operator), 교배(crossover) 연산자, 돌연변이(mutation) 연산자와 적합도를 이용해서 탐색을 수행한다. 이러한 방법은 목적함수의 미분이나 특별한 수학적 연산을 필요로 하지 않는다. 또한 유전자 알고리즘은 점에 의한 탐색이 아니라 개체들이 모여 이루어지는 개체군에 의한 병렬 탐색이라는 점에서 기존의 최적화 알고리즘과는 구별된다.

유전자 알고리즘은 일정한 세대 동안에 해가 될 가능성이 있는 염색체 벡터의 개체 집단을 만들어 적합도 계산시에 각각의 해를 평가한다. 이러한 과정에서 더 적합한 개체들을 선택(selection)함으로써 새로운 개체 집단이 재생산된다. 재생산이란 적합도 함수(fitness function)를 이용하여 개체들을 복사하는 과정이다. 이 새로운 개체 집단에 있는 염색체들의 일부 유전자들은 교배와 돌연변이에 의한 변경 과정을 거쳐 새로운 해를 구성한다. 교배는 부모의 염색체의 일부분을 서로 바꿈으로써 부모의 특징을 결합하여 두개의 유사한 자손들을 구성한

다. 돌연변이는 돌연변이율과 동일한 확률을 가지고 임의로 변화시킴으로써 선택된 염색체에서 하나 또는 그 이상의 유전인자를 변경하는 것이다(공성곤 등, 2002). 즉, 환경에 잘 적응하는 유전자만을 선택하고 교배하고 때에 따라서는 돌연변이도 하며 다음 세대에 우수한 유전 형질이 전달되도록 하는 방법이다. 지금까지 설명한 유전자 알고리즘의 흐름을 정리하면 다음의 <그림 2>와 같다.

```
t; //세대
P(t); //t 세대의 모집단
Genetic Algorithm {
    t=0;
    initialize P(t); //초기 모집단 생성
    evaluate P(t); //적합도 계산
    while(the termination conditions are not met)
    {
        //종료 조건이 만족되지 않으면
        t=t+1;
        select P(t) from P(t-1); //새로운 세대 생성
        alter P(t); //유전 연산(교배, 돌연변이)
        evaluate P(t); //적합도 계산
    }
}
```

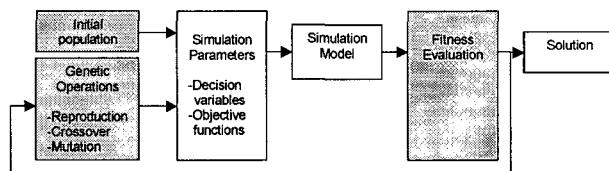
<그림 2> 유전자 알고리즘의 일반적인 절차

연속적인 탐색 공간에서 결정변수의 최적화를 다루는 문제에서는 유전자를 이진코딩하는 것보다 직접 실수로 표현해 주는 것이 더 자연스럽다. 염색체를 실수로 표현하면 문제의 결정변수와 정확하게 일대일로 일치하는 실수 유전자를 구성할 수 있다.

실수코딩 유전자 알고리즘도 이진코딩 유전자 알고리즘과 마찬가지로 적절한 스키마 정의를 도입하여 스키마 정리를 정립하려는 시도가 있었는데, Wright(1991)와 Radcliffe(1991)의 연구가 대표적이다. Wright는 실수변수 연속함수에 대해서 기존의 이진코딩 스키마의 개념을 검토하고 실수코딩 유전자 알고리즘의 스키마 정리를 체계화하는데 유용한 실수코딩 스키마를 정의하였다. Radcliffe는 임의 동치관계를 도입함으로써 암시적 병렬성의 개념을 비스트링 표현으로 확장하였는데 이 개념은 어떤 형태의 유전 연산자도 효과적으로 헤석할 수 있는 기틀을 마련하였다.

개체를 염색체의 형태로 바꾸는 것을 인코딩(encoding)이라 하고 그 반대의 경우를 디코딩(decoding)이라 하는데 이진코딩 유전자 알고리즘의 경우에는 이러한 인코딩과 디코딩 과정이 필수적이지만 실수코딩 유전자 알고리즘의 경우에는 실수를 염색체와 일대일로 매핑이 가능하기 때문에 인코딩과 디코딩 과정이 필요하지 않

다. 본 논문에서는 실수코딩 유전자 알고리즘을 사용하기 때문에 실수코딩 유전자 알고리즘이 연결된 시뮬레이션 모델의 개념도를 <그림 3>에서처럼 Lee et al.(2000)과는 다르게 구성한다.



<그림 3> RCGA와 연결된 시뮬레이션 모델의 개념도

실수코딩 유전자 알고리즘에서 각 세대를 진행하기 위한 염색체는 실수 변수와 일대일로 매핑이 가능하기 때문에 각 세대를 이루는 염색체의 유전인자를 실수 범위를 가지는 결정 변수로 구성할 수 있다. 다음의 <표 1>은 실수코딩 유전자 알고리즘에서 염색체의 유전인자로 사용되는 실험 문제의 결정 변수와 제한 범위를 보여준다. 본 연구에서는 실수코딩 유전자 알고리즘을 사용하기 때문에 염색체의 유전인자로 표시되는 결정 변수들의 값은 상한치와 하한치의 범위에서의 실수 값으로 선택된다. 선택된 결정 변수의 값들은 시뮬레이션 모델에서 입력 값으로 사용되어 적합도 값을 계산하기 위해 사용되는 목적식의 값을 도출한다.

<표 1> 염색체의 유전인자인 결정변수의 허용 구간

Gene #	Decision variable	Lower bound	Upper bound
1	Process time of WS1	10	15
2	Process time of WS2	8	11
3	Process time of WS3	8	18
4	Speed of Cart	50	70

<표 1>에서 설명된 결정 변수들의 제한 범위에서 각 유전인자를 선택하여 염색체를 구성한 후 시뮬레이션을 수행한다. 시뮬레이션 모델의 최적화를 위해서는 최적 대안을 선출할 수 있는 목적식을 고려해야 한다. 생산 시스템의 최적화를 위해 고려될 수 있는 목적식은 추구하는 목표에 따라 다양한 선택이 가능하지만 본 연구에서는 단위 시간동안 최대 생산량을 산출하는 조건을 목적식으로 고려한다. 다음의 <표 2>는 본 논문의 실수코딩 유전자 알고리즘에서 사용되는 적합도 함수와 목적식을 설명한다.

<표 2> 목적함수와 적합도 함수

Objective function	Fitness function
Maximize throughput	Simulated throughput

유전자 알고리즘을 사용하기 위해서는 앞에서도 언급한 것처럼 재생산 연산자, 교배 연산자 및 돌연변이 연산자에서 어떤 기법을 선택하느냐에 따라 최적해의 결과가 다르게 나타날 수 있다. 특히, 실수코딩 유전자 알고리즘의 경우에는 기존의 이진코딩 유전자 알고리즘과는 다르게 염색체를 구성하기 때문에 연산자 별로 다른 기법을 사용해야 한다. 지금부터는 실수코딩 유전자 알고리즘에서 사용이 가능한 기법들을 설명하고 본 연구에서 시뮬레이션 최적화를 위해 사용되는 연산자 기법을 선택한다.

3.1 재생산 연산자

실수코딩 유전자 알고리즘의 재생산 연산에 사용될 수 있는 기법은 결정표본선택, 교체를 수반한 잔여화률 분포선택, 확률균등선택, 토너먼트선택, 지역선택, 순위에 기초한 선택, 세대적선택, 정상상태선택 등이다.

일반적으로 가장 많이 사용되고 있는 루울렛선택 방법은 선택화률의 크기에 따라 이전의 집단에서 염색체를 선택하고 복제할 뿐 유전자를 변형할 수 없기 때문에 집단 내의 전체 유전자형의 변화에는 영향을 주지 못한다. 또한 확률적 속성 때문에 선택 과정에서 최적자(the fittest)를 반드시 선택하지 못하는 단점을 가진다.

결정표본선택은 교체를 수반한 확률분포선택 방법으로 루울렛선택의 단점을 보완하기 위한 방법이다. 이 방법은 각 염색체의 기대치의 정수값 만큼은 우선 배정하고 나머지 숫자만큼은 염색체들을 소수점 부분의 크기에 따라 배열하고 그 순서대로 선택하는 방법이다.

교체를 수반한 잔여화률표본선택은 결정표본선택을 개선한 것으로 자주 이용되는 연산자 중의 하나이다. 이 방법은 집단 내 각 염색체는 기대치의 정수 값 만큼은 우선 배정받고, 나머지 숫자만큼은 염색체들을 소수점 부분의 크기를 가지고 확률적으로 경쟁하도록 한 방법으로 본 연구에서는 재생산시 이 알고리즘을 사용한다(진강규, 2000).

재생산 과정을 통해서 새로운 집단을 생성할 때 선택하는 집단의 크기에 따라서 해를 찾는 속도에 차이가 발생하는데 이러한 집단의 크기는 코딩법과 코딩된 염색체의 길이에 의존한다. 이러한 개체 집단의 적절한 크기에 대한 첫 시도는 Goldberg(1985, 1989b)에 의해 연구되었지만 집단의 크기가 크다고 해서 빠른 시간에 해를

찾을 수 있다는 의미는 아니다. 일반적으로 집단의 크기는 30-200 사이면 적절하다고 연구되었다(Goldberg, 1989a). 본 연구에서는 개체 집단의 크기가 20, 40, 80, 200 일 때로 나누어 실험을 한다.

3.2 교배 연산자

재생산 과정이 끝난 후에는 생성된 개체 집단 내에서 염색체들을 검토하여 염색체들이 가지고 있는 유전인자들의 정보를 교환하는 교배 과정을 수행하는데 이를 구현하는 연산자는 코딩법에 따라 달라진다. 그러므로 이진코딩 유전 알고리즘이나 기호코딩 유전 알고리즘을 기반으로 하는 연산자는 실수코딩 유전자 알고리즘에 적용하기에는 어려움이 있기 때문에 실수코딩 유전자 알고리즘에 적합한 교배 연산자를 사용해야 한다.

실수코딩 유전자 알고리즘에 적합한 교배 연산자는 단순교배, 산술적교배, 프랫교배, Wright의 휴리스틱교배, 선형교배 등이다. Wright(1991)에 의해 제기된 단순교배는 임의로 선정된 부모 염색체를 랜덤으로 선택된 교배 점에서 교차하여 두 자손을 생성하는 방법이다. 산술적교배는 단순교배에서 일어나는 교배지점에서의 불연속성을 완화해주기 위한 방법으로 볼록 집합 이론을 기본 개념으로 하고 있다(공성곤 등, 2002). 프랫교배는 두 개의 개체 집단에서 두 개의 자손 개체 집단을 만들어내는 전통적인 교배 방법에서 벗어나 두 개의 부모 개체 집단을 교배하여 한 개의 자손 개체 집단을 생성하는 방법이다(Radcliffe, 1991). Wright의 휴리스틱교배는 부모 개체 집단으로부터 한 개의 자손 개체 집단을 생산하면서 적합도를 이용하여 탐색 방향을 결정하는 방법이다(Wright, 1990). 선형교배는 부모로부터 3개의 자손 개체 집단을 생성하는 방법이다(Wright, 1991). 본 실험에서는 간단하면서도 많이 사용되는 단순교배 알고리즘을 사용한다. 일반적으로 Lee et al.(2000)의 논문에서는 교배 확률을 60%-90% 사이에서 추천하였고, 진강규(2000)는 80%-95%사이에서 유지된다고 설명하였다. 본 연구에서는 교배 확률이 50%, 80%, 90%, 100%일 때를 가정한다.

3.3 돌연변이 연산자

돌연변이는 교배를 마친 염색체의 유전자들이 변경되는 것을 의미하는데 자연계에서는 돌연변이가 일어날 확률이 매우 작기 때문에 유전자 알고리즘에서 사용하는 돌연변이 확률도 매우 작게 된다. 염색체의 돌연변이가 너무 자주 발생하면 탐색해가 무작위로 선출되는 결과가 발생한다.

실수코딩 유전자 알고리즘에서 사용되는 돌연변이 연

산자는 균등돌연변이, 경계돌연변이, 실수크리프돌연변이, 동적돌연변이 등이 있다. 균등돌연변이는 염색체의 유전자가 돌연변이가 되면 정해진 영역내에 있는 임의의 실수로 교체하는 방법으로 전 탐색 공간을 자유롭게 이동하며 새로운 점을 찾기 때문에 초기 세대에 중요한 역할을 해주는 방법이다(공성곤 등, 2000). 경계돌연변이는 균등돌연변이의 변형된 형태로서 돌연변이가 발생하면 적합한 영역의 경계점들 중에서 하나를 임의로 취하는 방법이다. 실수크리프돌연변이 기법은 돌연변이가 발생하는 유전자에 작은 실수 값을 더하거나 빼는 방법이다(Davis, 1991). 동적돌연변이 기법은 정밀도를 높이기 위해서 미세 조정이 가능한 방법으로 세대에 따라 탐색영역을 조절이 가능하기 때문에 가장 많이 사용되는 방법이다(Janikow and Michalewicz, 1991). 본 실험에서는 세대에 따라 탐색 영역의 조절이 가능한 동적돌연변이 기법을 사용한다. 통상적으로 사용되는 돌연변이 확률은 0.5%-1%로 알려져 있으며 본 논문에서는 0.5%, 0.7%, 0.9%, 1%로 나누어 적용한다.

4. 실험 및 분석

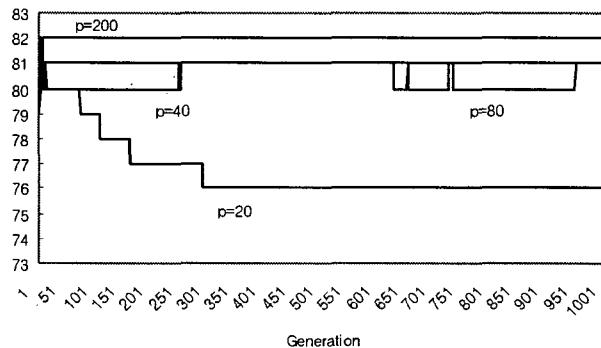
제 3장에서는 실수코딩 유전자 알고리즘을 사용하여 생산 시스템의 시뮬레이션 최적화를 추구할 때 사용되는 연산자에 대해서 설명하였다. 위에서 설명한 바와 같이 본 실험에서는 교체를 수반한 잔여확률분포 기법을 재생산 연산자로 사용하고 교배 연산자는 단순교배 기법을 사용한다. 돌연변이 연산자는 동적돌연변이 기법을 사용한다.

각 실험 결과의 비교를 위한 기준 파라미터로서 개체 집단의 크기는 40으로 하고 교배 확률은 80%로 하며, 돌연변이 확률은 0.5%로 할당한다. 즉, 개체의 재생산시 개체 집단의 크기를 20에서 200까지 변경하면서 재생산 연산자를 사용하게 되는데 이때 교배 확률은 80%로 고정되고 돌연변이 확률은 0.5%로 고정된다. 교배 확률과 돌연변이 확률을 변경하여 시험할 때도 같은 방법으로 진행된다. 재생산시 4개의 제약 조건에 대해서 1,000세대 동안 시뮬레이션 하는 것을 가정한다.

4.1 재생산 연산자 실험

재생산 연산자 변경 실험에서는 교배 연산자와 돌연변이 연산자의 파라미터는 고정하고 집단의 크기만 20, 40, 80, 200으로 변경하면서 실험한다. <그림 4>는 1,000 세대 동안 집단 크기를 변경하면서 생산 시스템의 최적화를 수행한 후 각 세대별 생산량을 보여준다. 집단크

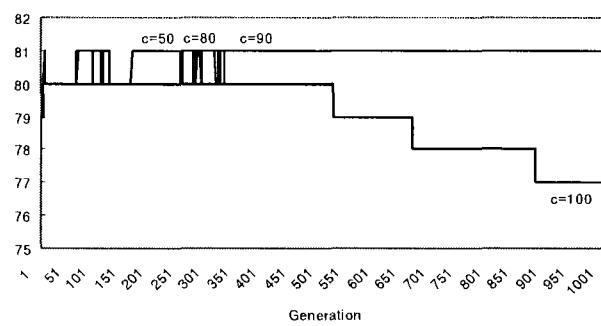
기가 20($p=20$)인 경우에는 292세대부터 76에 수렴한다. 집단크기가 40($p=40$)인 경우에는 252세대에서 81에 수렴하고, 집단크기가 80($p=80$)인 경우에는 952세대에서 81에 수렴한다. 집단크기가 200($p=200$)인 경우에는 9세대에서 82에 수렴함을 알 수 있다. 실험의 결과를 분석하면 집단의 크기가 20인 경우에는 시뮬레이션 시간은 매우 작지만 292세대 이후에야 특정 값으로 수렴하고 수렴하는 값도 매우 작다는 것을 알 수 있다. 집단의 크기가 클수록 시간은 많이 걸리지만 선택할 수 있는 초기 해가 많아 집단의 크기가 작은 경우보다 좋은 근사해에서 출발할 확률이 높기 때문에 적은 세대수를 실행하고도 좋은 결과를 준다는 것을 알 수 있다. 그러나 집단의 크기가 40인 경우와 80인 경우를 보면 집단의 크기가 작은 40에서 빠른 세대에서 81에 수렴하기 때문에 집단의 크기가 클수록 좋은 결과를 준다는 보장은 없다는 것을 알 수 있다.



<그림 4> 모집단의 크기에 따른 실험 결과

4.2 교배 연산자 실험

다음의 <그림 5>는 다른 변수들은 고정하고 교배 확률만 50%, 80%, 90%, 100%로 변경할 때의 결과이다. 교



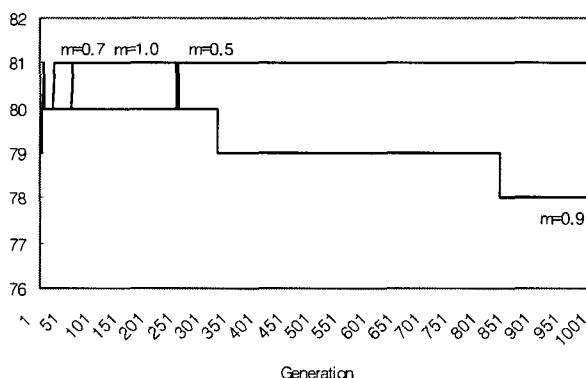
<그림 5> 교배 확률에 따른 실험 결과

배 확률이 50%(c=50)일 때는 161세대 만에 81에 수렴하고, 교배 확률이 80%(c=80) 일 때는 252세대 만에 81에 수렴한다. 교배 확률이 90%(c=90)일 때는 328세대 만에 81에 수렴하고, 교배 확률이 100%(c=100)일 때는 879세대 만에 77에 수렴한다. 교배 확률을 변경하는 실험에서는 교배 확률이 50%일 때 가장 빠른 세대에 가장 많은 생산량 81에 도달하지만 교배 확률이 100%일 때는 반대로 가장 작은 생산량 77에 도달한다. 이러한 현상은 부모 세대의 유전 형질을 자식 세대로 전달할 때 염색체를 100% 변경하는 것은 부모 세대가 가지고 있는 우성 형질을 오히려 변형시킴으로서 최적해에 도달하지 못한다는 것을 보여주기 때문에 적절한 교배 확률의 선택이 필요하다는 것을 알 수 있다.

4.3 돌연변이 연산자 실험

제생산을 위한 집단 크기를 40으로 고정하고, 교배 확률이 80%로 고정될 때 돌연변이 확률 0.5%, 0.7%, 0.9%, 1.0% 인 경우의 결과를 <그림 6>에서 볼 수 있다. 돌연변이 확률이 0.5%($m=0.5$)일 때는 251세대에 81에 수렴하고, 0.7%($m=0.7$)일 때는 24세대 반에 81에 수렴한다. 돌연변이 확률이 0.9%($m=0.9$)일 때는 836세대에서 78에 수렴하고, 1.0%($m=1.0$)일 때는 58세대에서 81에 수렴한다.

돌연변이 확률만을 변경한 실험에서는 돌연변이 확률이 0.7%일 때에 가장 빠른 세대에서 가장 많은 생산량인 81에 수렴하지만, 돌연변이 확률이 0.9%일 때는 836 세대에서 78에 수렴한다. 실험의 결과를 보면 돌연변이 확률이 높다고 해서 꼭 좋은 해를 보장하지 못한다는 것을 알 수 있다. 이러한 이유는 부모의 우성 인자를 통해서 전역 최적해를 찾아가다가도 돌연변이에 의해 지역 최적해로 수렴하는 확률이 발생하기 때문으로 판단된다.

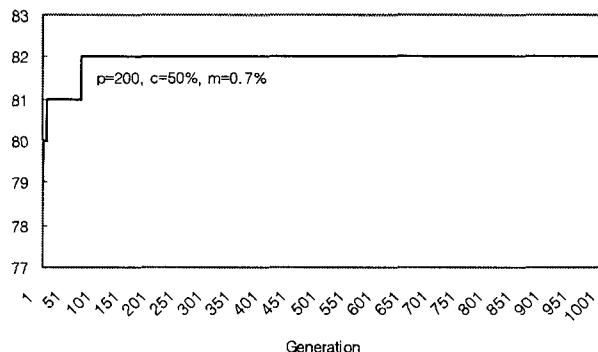


<그림 6> 돌연변이 확률에 따른 실험 결과

4.4 최적 파라미터 실험

주어진 실험 조건을 사용했을 때 재생산 과정에서의
집단크기 변경, 교배 과정에서의 교배 확률 변경 및 돌
연변이 확률 변경 실험을 통해서 가장 좋은 변수 조합
은 집단 크기가 200, 교배 확률은 50%, 돌연변이 확률이
0.7%일 때이다. 따라서 본 절에서는 실험을 통해서 가장
좋다고 결정된 최적 파라미터들을 적용하여 실험 결과
를 분석한다.

<그림 7>의 결과를 보면 최적 파라미터로 선정된 $p=200$, $c=50$, $m=0.7$ 에서는 71세대 만에 82에 도달하였다. 이 결과는 <그림 5>에서 $p=40$, $c=50$, $m=0.5$ 일 때 161세대 만에 81에 수렴한 경우와 <그림 6>에서 $p=40$, $c=80$, $m=0.7$ 일 때 24세대 만에 81에 수렴한 경우보다는 좋은 결과를 보여준다. 그러나 <그림 4>에서 $p=200$, $c=80$, $m=0.5$ 일 때 9세대 만에 82에 도달한 경우와 비교해 보면 두 경우에서 모두 82에 수렴하지만 최적으로 선정된 파라미터 조합이 아닌 경우에 훨씬 빠른 세대에 수렴함을 알 수 있다. 위의 결과를 분석해 보면, 재생산, 교배 및 돌연변이 연산자를 내에서 최적으로 선정된 파라미터라도 다른 연산자와의 조합에서는 서로 영향을 주기 때문에 최적 파라미터 조합도 최적해를 항상 보장하지 못한다는 것을 알 수 있다.



<그림 7> 최적 파라미터 적용시의 실행 결과

5. 결 론

본 연구에서는 실수 코딩 유전자 알고리즘을 생산 시스템의 시뮬레이션 최적화에 적용하였다. 실수코딩 유전자 알고리즘을 사용할 경우에는 일반적으로 많이 사용되던 이진코딩 유전자 알고리즘이 표시되는 염색체형이 다르기 때문에 실수 표현이 가능한 연산자를 사용해야 한다. 그러므로 본 논문에서는 재생산 연산자로는 교

체를 수반한 잔여화를 분포 방법을 사용하였고 교배 연산자로는 단순 교배 방법을 사용하였다. 돌연변이 연산자는 세대에 따라 탐색 영역의 조절이 가능한 동적돌연변이 방법을 사용하였다.

실수코딩 유전자 알고리즘의 실험을 위해서 카트로 이송되는 생산 시스템을 ARENA 툴을 사용하여 시뮬레이션 모델로 구축하고 유전자 알고리즘은 VBA로 구현하였다. 재생산 연산자 실험에서는 집단의 크기가 20, 40, 80, 200인 경우를 비교하였고 교배 연산자 실험에서는 교배 확률이 50%, 80%, 90%, 100%인 경우를 비교하였다. 돌연변이 연산자 실험에서는 돌연변이 확률이 0.5%, 0.7%, 0.9%, 1.0%인 경우를 비교하였고 실험을 통해 도출된 최적 파라미터 조합 결과를 비교하여 파라미터 간의 관계를 살펴보았다.

재상산 연산자 실험에서는 집단의 크기가 200($p=200$) 일 때 가장 빠른 세대에 가장 좋은 최적해에 수렴했고, 교배 연산자 실험에서는 교배 확률이 50%($c=0.5$), 80%($c=0.8$), 90%($c=0.9$) 일 때 가장 좋은 최적해에 도달했지만 교배 확률이 50%로 가장 작을 때 제일 빠른 세대에 최적해에 수렴하였다. 돌연변이 연산자 실험에서는 돌연변이 확률이 0.5%($m=0.05$), 0.7%($m=0.07$), 0.1%($m=0.1$) 일 때 가장 좋은 최적해에 수렴하지만 돌연변이 확률이 0.7%일 때 가장 빠른 세대에서 수렴하였다. 각각의 연산자 내에서의 파라미터 값을 변경한 실험에서는 집단의 크기나 교배 확률 및 돌연변이 확률이 커진다고 해서 최적해를 보장하지는 못하기 때문에 파라미터 값을 적절하게 선택하는 것이 필요하다는 것을 알 수 있다.

추후 연구 과제는 부모 세대에서 자식 세대로 유전자 형을 전파할 때 부모 세대에서 가장 우성인 형질이 자식 세대로 그대로 유지될 수 있도록 하는 엘리트증(elitism) 기법을 적용하여 적용하지 않는 경우와 효율을 비교하는 것이 필요하다고 판단된다.

참고문헌

- [1] 공성곤, 김인택, 박대희, 박주영, 신요안, 유전자 알고리즘, 그린, 2002.
- [2] 진강규, 유전알고리즘과 그 응용, 교우사, 2000.
- [3] A. Wright, Genetic algorithms for real parameter optimization, Foundations of Genetic Algorithms, 1st Workshop on the Foundations of Genetic Algorithms and Classifier Systems, Morgan Kaufmann Publishers, pp. 205-218, 1990.
- [4] A. Wright, Genetic algorithms for real parameter optimization, Foundations of genetic algorithms, G. Rawlin (Ed.), Morgan Kaufmann publishers, CA, 1991.
- [5] C. Z. Janikow and Z. Michalewicz, An Experimental comparison of binary and floating point representations in genetic algorithms, Proceedings of 4th International Conference on Genetic Algorithms, Morgan Kaufmann Publishers, 1991.
- [6] D. E. Goldberg, Optimal initial population size for binary-coded genetic algorithms, TCGA Report No. 85001, University of Alabama, 1985.
- [7] D. E. Goldberg, Genetic algorithms in search, optimization and machine learning, Addison-Wesley, N.Y., 1989a.
- [8] D. E. Goldberg, Sizing populations for serial and parallel genetic algorithms, Proceedings of 3rd International Conference on Genetic Algorithms and Their Applications, pp. 70-79, 1989b.
- [9] J. M. Yunker and J. D. Tew, Simulation optimization by genetic search, Mathematics and Computers in Simulation, 37(1), pp. 17-28, 1994.
- [10] L. Davis, Handbook of genetic algorithms, Van Nostrand Reinhold, 1991.
- [11] M. S. Meketon, Optimization in simulation : survey of recent results, Proceedings of 1987 Winter Simulation Conference, pp. 58-67, 1987.
- [12] N. C. Ho, S. S. Lee, and Y. L. Loh, A two-stage approach for optimizing simulation experiments, Annals of CIRP, 42(1), pp. 501-504, 1993.
- [13] N. J. Radcliffe, Equivalence class analysis of genetic algorithms, Complex Systems, 5(2), pp. 183-205, 1991.
- [14] S. G. Lee, L. P. Khoo, and X. F. Yin, Optimizing an assembly line through simulation augmented by genetic algorithms, International Journal of Advanced Manufacturing Technology, 16, 220-228, 2000.
- [15] W. D. Kelton, R. P. Sadowski, and D. A. Sadowski, Simulation with arena 2d, McGraw-Hill, 2002.