

# 이산사건 시물레이션을 사용한 시스템의 설계

## System Design Using Discrete Event Simulation

이 영해<sup>†</sup>, 박 경종<sup>‡</sup>, 한 관희<sup>‡</sup>

<sup>†</sup> 한양대학교 산업공학과

<sup>‡</sup> 대우정보시스템 CIM 사업부

Young-Hae Lee<sup>†</sup>, Kyoung-Jong Park<sup>‡</sup>, Kwan-Hee Han<sup>‡</sup>

<sup>†</sup> Dept. of I. E., Hanyang University

<sup>‡</sup> Dept. of CIM, Daewoo Information Systems Co., Ltd.

### Abstract

In this paper we use discrete simulation method to get the criteria of system evaluation required in the case of designing the complicated probabilistic event system having discrete probabilistic variables and to search the effective and reliable alternatives to satisfy the objective value of the given system through on-line, single run within the short time period. If we find the alternative we construct the algorithm which change values of decision variables and determining alternative by using the stopping algorithm which end the simulation in the steady state of system.

In order to prevent the loss of data when we analyze the acquired design alternative in the steady state we provide the background of the estimation of the autoregressive model and mean and confidence interval for evaluating correctly the objective function obtained by the small amount of output data through the short time period simulation.

### 1. 서론

복잡한 확률적 이산 사건 시스템의 설계 도구로서 이산 사건 시물레이션을 사용하기 위해서는 극복해야 할 몇 가지 문제가 있다. 첫째, 시스템의 성능을 평가할 목적함수와 일부 제약식의 값은 알려진 함수를 이용한 단순한 계산이 아니라 오직 시물레이션을 수행하여서만 얻을 수 있다. 둘째, 시물레이션을 수행하여 얻은 결과는 확률적 이산 사건 시스템의 경우 반드시 확률적 요소가 포함되어 있기 때문에 효과적인 통계적 분석이 요구되고, 최적화 문제도 결정적(deterministic) 최적화가 아닌 확률적 최적화 문제가 된다. 셋째, 수행하는 시물레이션의 형태가 안정상태(steady state) 시물레이션인 경우, 하나의 대안에 대하여 안정상태에서의 시스템의 평가척도를 구할 때, 출력 데이터들이 서로 자동상관(autocorrelation)을 내포하여 이를

해소하는데 많은 출력 데이터들이 필요하게 되어 과도한 실행 시간이 걸리게 된다. 넷째, 시물레이션을 이용하여 하나의 대안을 평가하는 데도 많은 시간이 걸리는데, 해 발견을 위한 탐색 공간이 클 경우에는 더욱 많은 시간이 걸리게 된다[1].

시물레이션 최적화 문제는 일반적으로, 첫째, 목적함수의 수에 의한 분류, 둘째, 변수에 의한 분류, 셋째, 제약조건에 의한 분류로 나눌 수 있으며, 사용된 방법은 Fu[6], Azadivar[1], Meketon[14], 그리고 Safizadeh[16] 등에서 참고할 수 있다.

일반적으로 많이 사용해 왔던 finite differences을 이용한 방법에서는 gradient를 추정하기 위해서 n개의 파라미터에 대해 최소한 n+1번의 시물레이션 반복 수행이 필요하다는 단점을 가지고 있다[8]. 따라서, 시물레이션 최적화를 수행할 때 발생하는 많은 반복 수행의 문제를 해결하

기 위해서는 단일 실행(single run)으로 최적화가 가능한 방법들을 필요로 한다. 단일 실행으로 최적화를 수행하기 위해 Perturbation Analysis(PA) 방법과 Score Function(SF) 방법이 제안되었지만, 이 방법들은 모두 다 연속형 결정변수를 대상으로 하고 있다.

또한, 현재까지의 시물레이션 최적화에 대한 연구는 연속형 결정변수에 대하여 주로 연구되고 발전되었으나, 이산형 결정변수에 대한 시물레이션 최적화 연구는 상대적으로 연구가 많이 진행되지 못했다. 그러나, 시스템의 버퍼 최적화나 자원 최적화 등에서 많은 관심과 필요성이 증대되어 최근에 와서 관심이 높아지고 있다[2, 12, 15, 17, 20].

기존 연구들 중에서 특히 Chen[4]은  $g(x)=\gamma$  라고 표현되는 stochastic root finding 문제에 대해 Monte Carlo sampling을 이용한 알고리즘과 retrospective approximation[5] 방법을 이용한 알고리즘을 제시하였다. Chen이 제시한 방법은 매우 많은 시물레이션의 반복 실행을 요구하므로 간단한 수치 시물레이션을 목적함수의 평가 도구로 이용하고, 목적함수가  $g(x)=\gamma$  라고 표현되는 경우에는 이용이 가능하나, 복잡하고 한 번의 시물레이션 수행동안 긴 실행시간을 요구하는 안정상태 이산사건 시물레이션에는 적합하지 않다.

Wild and Pignatiello[19]는 안정상태 이산사건 시물레이션을 이용하여 시스템을 설계할 시 주어진 목표치( $g(x) \leq \gamma$ 의 형태로 표시)를 만족하는 대안을 발견하기 위해 시물레이션 도중에 결정변수 값을 변경해 가는 reverse-simulation이라고 부르는 방법을 제시하였다. 그러나, 이 방법은 결정변수 값의 변경에 대해 시스템 목적함수 값의 변화의 방향을 알 수 있는 M/M/s로 모델링 될 수 있는 아주 단순한 경우에만 적용이 가능하고, 각 개체가 시스템을 떠날 때마다 결정 변수 값을 변경하므로 안정되고, 신뢰할 수 있는 대안을 얻기가 어렵고, 다양한 형태의 목표치가 주어진 경우에 사용하기가 불가능하다. 지금까지의 연구들과 개발된 방법들은 이산 사건 시물레이션 기법을 이용한 이산 사건 시스템의 최적 설계시 반드시 고려해야 할 긴 시물레이션 수행 시간을 간과한 경우가 많았고, 단지 시물레이션을 기초로 한 확률적 최적화 알고리즘에 국한하는 경우가 많았다.

그러므로, 본 연구에서는 이산 사건 시물레이션 기법을 이용하여 이산형 변수를 가진 복잡한 확률적 이산 사건 시스템을 설계할 경우, 요구되는 시스템 평가척도와 제약식을 충족하며 신뢰할 수 있는 대안을 효율적으로 빠른 시간 내에 온-라인으로 발견해 내는 문제를 다룬다. 이를 위해 필요한 탐색 알고리즘을 개발하고, 짧은 시물레이션의 수행으로 얻은 데이터를 이용하여 안정상태의 반응치를 추정하는 기법을 개발한다.

## 2. 해답색 알고리즘 및 종료 알고리즘

시스템 설계 대안을 단일 시물레이션 실행 환경에서 찾기 위한 기본적인 알고리즘은 다음과 같다.

### [알고리즘 2.1]

단계 1: 목적함수( $f_i(X)$ ), 결정변수( $x_j, j=1, \dots, n$ ), 목표치( $A_i$ ), 결정변수  $x_j, j=1, \dots, n$ 에 대한 증감( $\Delta x_j$ ), 목적함수를 평가하고 결정변수 값을 조정할 시간 간격( $\Delta t$ )을 설정하고, 시물레이션을 시작한다.

단계 2: 시물레이션 도중  $\Delta t$  마다 목적함수와 목표치를 비교하여 결과에 따라 바람직한 방향으로  $\Delta x_j$  만큼 결정변수 값을 조정하고, 시물레이션을 계속한다(알고리즘 2.2 참조).

단계 3: 알고리즘 종료 조건을 검사하고 만족하면, 최근 시점까지 가장 많이 방문했던 결정변수 값들의 조합을 최종해로 정하고 시물레이션을 종료한다(알고리즘 2.3 참조).

단계 4: 얻어진 최종해를 가지고 충분한 길이의 검증 시물레이션을 수행하고 필요한 자료를 출력한다

알고리즘 2.1에서 목적함수를 평가하고 결정변수 값을 조정할 시간 간격( $\Delta t$ )의 단위는 바로 앞에서 결정변수 값의 조정 행위가 일어난 뒤로부터의 경과 시간, 완성된 부품수, 지나간 고객수, 또는 시물레이션에서 처리된 사건(event) 수 등으로 설정이 가능하며, 수행하는 시물레이션의 형태가 종료형(terminating) 시물레이션인 경우에  $\Delta t$ 는 수행하는 시물레이션의 총 수행 길이가 될 수 있다.

그러나, 안정상태 시물레이션인 경우는 주어진 변수 값들의 조합인 하나의 대안에 대하여 안정상태에서의 출력을 얻을 때까지는 상당히 긴 시물레이션의 수행 길이가 요구되므로, 요구되는 최종해를 얻을 때까지 필요한 여러 대안들의 평가에 걸리는 총시간은 컴퓨터로 주어진 시간 내에 처리할 수 없는 매우 긴 시간이 될 수 있다. 따라서, 이를 줄이기 위한 노력이 절대적으로 필요하다.

일반적으로  $\Delta t$ 가 작을수록 자주 목적함수를 평가하고 결정변수 값을 조정하므로 빨리 목표치를 만족하는 결정변수 값으로 수렴할 확률이 높을 수도 있겠으나,  $\Delta t$ 가 작은 만큼 얻은 데이터의 수가 적어 목적함수 값을 평가하는 데 오차가 커질 수 있다. 그리고, 목적함수를 평가하고 결정변수 값을 조정하는데, 많은 계산 시간을 소비하게 된다. 이를 위해 다음의 제 3 장에서는 짧은 시물레이션의 수행으로 안정상태에서의 목적함수 값을 효율적으로 추정할 수 있는 방법을 기술한다.

알고리즘에서 종료 조건을 만족하더라도 확률적 시물레이션의 특성상, 종료시점 근처에서 결정변수의 값들이 어느 정도의 증감이 발생할 수 있으므로, 이런 경우에는 최근 가장 많이 방문했던 결정변수 값을 최종해로 정하고 시물레이션을 종료해야 한다.

### 2.1 결정변수 조정

알고리즘 2.1의 단계 2에서, 시물레이션 도중  $\Delta t$ 마다 목적함수와 목표치를 비교하여 결과에 따라 바람직한 방

향으로  $\Delta x_j$  만큼 결정변수  $x_j$ 의 값을 조정한다고 했는데, 각 변수에 대하여 시물레이션 출력으로 얻는 목적함수의 값이 단조 증가 함수(monotonic increasing function) 또는 단조 감소 함수(monotonic decreasing function) 인지를 파악하여 주어진  $A_i$ 의 형태  $\{f_i=c\}$ ,  $\{f_i > c\}$ , 또는  $\{f_i < c\}$ , t시점과 t-1시점의 결정변수 j의 값  $x_{j,t-1}$  와  $x_{j,t}$ , t시점과 t-1시점에서 시물레이션 모델로부터 얻어진 목적함수 i의 값  $y_{i,t-1}$  와  $y_{i,t}$ , t시점에서 시물레이션 모델로부터 얻어진 목적함수 i와 주어진 목표치의 값  $y_i$  와 c의 관계에 따라 목적함수에 대하여 필요한 각 변수 값의 증가, 감소, 그리고 불변의 횟수를 누적하여 가장 많은 횟수를 기록한 방향으로 각 변수 값을  $\Delta x_j$  만큼씩 변화를 시킨다. 이로 인해 변수 값의 변화 방향이 목적함수에 따라 충돌(conflict)이 발생하는 경우도 동시에 고려할 수 있으며, 결정변수 값의 조정을 위한 구체적인 알고리즘 2.2는 Figure 1에서 설명되고 용어 설명은 다음과 같다.

$x_{j,t}$ : 시물레이션 출력 평가가 행해지는 시점 t에서의 결정변수 j의 값

$y_{i,t}$ : 시물레이션 출력 평가가 행해지는 시점 t에서의 목적함수 i의 시물레이션 출력값

countj0: 결정변수 j에 대하여  $x_{j,t-1}$ 에서  $x_{j,t}$ 로의 변화를 관찰할 때, 불변에 해당하는 횟수의 누적수

countj+: 결정변수 j에 대하여  $x_{j,t-1}$ 에서  $x_{j,t}$ 로의 변화를 관찰할 때, 증가에 해당하는 횟수의 누적수

countj-: 결정변수 j에 대하여  $x_{j,t-1}$ 에서  $x_{j,t}$ 로의 변화를 관찰할 때, 감소에 해당하는 횟수의 누적수

$a_j$  와  $b_j$ : 결정변수 j의 하한치와 상한치

$Li++$ : 변수  $x_j$  값을 증가(감소)시키면 목적함수 i값이 증가(감소)하는 변수들의 집합

$Li+-$ : 변수  $x_j$  값을 증가(감소)시키면 목적함수 i값이 감소(증가)하는 변수들의 집합

$Li0$ : 목적함수 i에 대하여  $Li++$  나  $Li+-$ 에 포함되지 않는 변수들의 집합

## 2.2 종료 알고리즘

매 시점 t에서 수행하게 될 알고리즘의 종료조건 검사를 위해 필요한 추가적인 용어 및 알고리즘을 설명하면 다음과 같다.

$x_j^*$ : 현시점 t에서 최근 K 번의 시점에서 결정된 변수의 값들 중에서 빈도수가 가장 높은 값

K: 종료 조건 검사를 위해 기준이 될 최근의 시점 수 ( $K = 1, 2, \dots, n$ )

$\eta_j$ : 종료 조건을 위해 결정변수  $x_j$ ,  $j=1, \dots, n$ 에 대하여 허용치 ( $x_j^* \pm \eta \Delta x_j$ )를 산정하는데 사용되는 정수 ( $\eta = 0, 1, 2, \dots, n$ )

### [알고리즘 2.3]

단계 0: 정수인 K,  $\eta$  값을 부여한다.

단계 1: 결정변수  $x_j$ ,  $j=1, \dots, n$ 에 대하여 다음 과정을 수행

고, 모든 결정변수에 대하여 만족하면 단계 2로 간다.

알고리즘 2.1에 의해  $x_j^*$ 를 구하고, 최근 K 시점들에서 결정변수  $x_j$ ,  $j=1, \dots, n$ 의 모든 값이  $[x_j^* \pm \eta \Delta x_j]$  범위에 들어오면, 단계 2로 간다. 그렇지 않으면, 본 알고리즘을 중단하고 다음 시점 t까지 시물레이션을 계속한다.

단계 2: 목표치 i에 대하여 만족 여부를 확인하여, 만족하면 시물레이션을 종료한다.

그렇지 않은 경우,  $\eta = 0$ 이면, 본 알고리즘 및 시물레이션을 중단하고, '목표치를 만족하는 대안이 존재하지 않음'으로 결론을 내리고,  $\eta > 0$  이면,  $\eta$ 를  $\eta = \max[0, \eta - 1]$ 로 수정한 뒤 다음 시점  $t + \Delta t$  까지 시물레이션을 계속한다.

단계 0에서의 K값 ( $K = 1, 2, \dots, n$ )은 시물레이션을 통해 얻는 시스템 수행도의 확률성의 크기에 종속되며, 시스템 설계자에게 주어진 컴퓨터 종류와 할당된 시간을 고려하여 시스템 설계자가 결정해야 한다.  $\eta$  값( $\eta = 0, 1, \dots, n$ )은 정의된 문제 및 변수의 성질에 따라 달라져야 하는데, 보통 0 또는 1의 값이면 적당하다.

단계 2에서는 모든 결정변수 값들이 종료를 위한 허용 범위 내에 포함되어도 목표치를 만족하지 못하면 허용 범위를 줄인다. 더 이상 허용치가 없는 경우( $\eta = 0$ )는 '목표치를 만족하는 대안이 존재하지 않음'으로 결론을 내리고, 알고리즘 및 시물레이션을 중단하고, 허용치가 있는 경우( $\eta > 0$ )는 시물레이션을 계속 수행함을 의미한다.

## 3. 안정상태 목적함수 평가

앞의 제 2장에서 기술한 알고리즘 2.1에서  $\Delta t$ 가 작을수록 자주 목적함수를 평가하고 결정변수 값을 조정하므로 빨리 목표치를 만족하는 결정변수 값으로 수렴할 확률이 높을 수도 있겠으나,  $\Delta t$ 가 작은 만큼 얻은 데이터의 수가 적어 목적함수 값을 평가하는 데 시물레이션 모델의 확률적 성질에 의해 오차가 커질 수 있다. 이를 위해 본 장에서는 비교적 짧은 시물레이션의 수행으로 얻은 적은 수량의 데이터로 안정상태에서의 목적함수 값을 효율적으로 추정할 수 있는 방법을 기술한다.

일반적으로, 시물레이션 출력의 초기 편의(initial bias)의 영향을 받지 않는 안정상태에서 추정치를 얻기 위해서는 굉장히 긴 시간의 시물레이션의 수행이 필요한데, 이를 위해 시물레이션 연구자들이 많은 연구를 해왔다[11].

Voss et al.[18]은 비교적 짧은 변이기간(transient period) 동안의 시물레이션 수행으로 안정상태에서의 목적함수 값을 효과적으로 얻을 수 있는 자동회귀 모델(Autoregressive Model)을 이용한 방법을 제시하였으며, 실험을 통하여 기존의 방법인 Unweighted Batch Means[11], Weighted Batch Means[3] 방법과 비교하여 Mean Error, Root Mean Square Error, Coverage, Mean Interval Length 등의 면에서 비슷하거나 우수함을 보였다.

따라서, 본 연구에서는 비교적 짧은 시간인  $\Delta t$  동안

시뮬레이션을 수행하여 안정상태시의 목적함수 값을 추정하기 위해 Voss et al.[18]이 제시한 방법을 기초로 하여 사용하며, 알고리즘을 정리하면 다음과 같다.

**[알고리즘 3.1]**

- 단계 1: Finite Information Criterion(FIC)인 p개의 계수를 사용한 예측오차(prediction error)를 이용하여 p\*를 계산한다.
- 단계 2: 자동회귀 계수 벡터  $\Phi$ 의 추정치인 조건부 최소자승 추정치(conditional least square estimate)  $\hat{\Phi}$ 를 Fuller[7] 식에 의해 구한다.
- 단계 3:  $\Delta t$  동안에 얻은 n개의 데이터에 의한 안정상태 평균  $\mu(\Phi)$ 의 추정치  $(\hat{\mu}_n)$ 을 표준 평균과 편차의 수정을 고려하여 구한다.

**4. 실험 및 평가**

본 장에서는 제 2 장 및 3 장에서 개발된 알고리즘을 평가하기 위한 실험을 수행한다. 시뮬레이션 시스템에 대한 이론치를 구할 수 있고, 결정변수의 값을 변경함으로써 목적함수의 방향을 알 수 있는 M/M/s 대기모형으로 표현될 수 있는 단일 작업장을 가진 제조시스템을 대상으로 구현된 알고리즘의 효율을 파악하고 나타나는 결과를 분석한다.

작업물이  $\lambda=18$ 인 지수분포로 도착하여 대기 버퍼에 대기하다가  $\mu=2$ 인 지수분포로 작업을 수행한 후 시스템을 빠져나가는 모델에서, 대기 버퍼의 용량은 무한대이며, 기계의 작업물의 선택 규칙은 선입 선출 방법을 따른다고 가정한다. 목표치는 대기 버퍼에서 기다리는 작업물의 대기 시간이 10이하가 되도록 하는 최소 기계의 대수를 찾는 것인데, Hillier & Lieberman[9]와 Little[13]의 공식을 사용하여 목표치를 만족하는 안정상태에서의 이론치를 구하면 기계의 수가 최소 11대가 되면 작업물의 대기시간 10이하를 만족한다.

알고리즘 2.2에서 기계 대수 x1의 증감값( $\Delta x1$ )을 1로 하고, 목적함수를 평가하고 결정변수 값을 조정할 시간 간격( $\Delta t$ )은 작업을 마친 작업물의 수가 30, 50, 100, 200, 그리고 300일 때로 한다. 또한 가용한 기계 대수의 범위를  $[a1=1, b1=20]$ 로 설정하며, 본 절에서 실험하는 단일 작업장 문제는 기계 대수가 증가(감소)하면 작업물이 대기하는 대기 시간이 감소(증가)하기 때문에 L1+- = {x1}에 속하는 문제이다.

알고리즘 2.3에서 시스템을 만족하는 기계 대수의 변화 구간을 검사하는 회수를  $K=10$ 으로 하고 기계 대수의 허용되는 변화 폭  $\eta$ 를 0으로 한다. 또한, 알고리즘 3.1에서 pmax를 20으로 하고 결과를 관찰한다.

알고리즘 2.1, 2.2, 2.3, 그리고 알고리즘 3.1을 사용하여 각각의  $\Delta t$ 에 따라 목적함수( $w \leq 10$ ) 조건을 만족할 때까지 진행되는 수렴 방향은 다음의 Table 1을 분석하면 알

수 있다.

Table 1. The machine vs. time at  $\Delta t$  in single workstation

$\Delta t$	# of machines	# of parts	Ending time
30	20	1650	5227.28
50	11	2550	8246.61
100	11	4800	15807.87
200	11	9200	30381.43
300	11	13800	45323.06

Table 1에서, 각각의  $\Delta t$ 에 따라 시뮬레이션을 수행하는 동안 선택되는 기계 대수의 변화를 살펴보면 초기에는 필요한 기계 대수가 계속해서 증가하다가 선택되는  $\Delta t$ 의 크기에 따라 기계 대수가 변하는데,  $\Delta t$ 가 30인 경우에는 선택된 최소 기계 대수가 20일 때 시뮬레이션이 종료되며, 구해진 기계 대수 20은 이론적인 최소 기계 대수 11과 많은 차이가 난다. 이러한 원인은  $\Delta t$ 의 크기가 시스템의 상태를 적절하게 판단하여 결정 변수인 기계 대수를 변경하여 만족하는 시스템 설계 대안을 제공할 수 있을 만큼 충분히 크지 못하기 때문이다.

$\Delta t$ 가 50 이상인 경우에는 시뮬레이션이 진행됨에 따라 이론적인 최소 기계 대수 11에 근접하면서 가능해 영역을 설정하기 때문에 단일 작업장 문제의 경우에는  $\Delta t$ 가 50 이상이 되면 적당하다고 설명할 수 있지만, 다른 유형의 문제에서는 시스템의 특성에 따라 만족하는  $\Delta t$ 의 크기도 다를 것으로 예측된다.

**5. 결론 및 추후 과제**

본 연구에서는, 이산형 결정변수를 갖는 확률적 시스템을 대상으로 단일 실험으로 빠른 시간에 온-라인으로 최적의 시스템의 설계 대안을 발견하고, 얻어진 설계 대안을 안정상태에서 분석할 때 데이터의 손실을 방지하기 위해 짧은 시뮬레이션 수행으로 적은 수의 출력 데이터로 주어진 목적 함수의 평가를 위해 자동회귀 모델링 방법을 사용하였다.

추후 연구 과제는 시스템의 결정변수로 정성적인 인자-시스템의 운영 룰(FIFO, LIFO 등), 부품의 작업 순서, 또는 설비 배치 문제-가 고려될 때 제안된 해 탐색 알고리즘과 안정 상태 시뮬레이션을 종료하는 알고리즘의 개선이 필요할 것으로 생각된다.

**참고문헌**

[1]Azadivar, F., "A Tutorial on simulation optimization", Proceedings of the 1992 Winter Simulation Conference,

198-204, 1992.  
 [2] Barry, D.A., and B. Fristedt, Bandit problems, Chapman and Hall, London, 1985.  
 [3] Bischak, D. P., W. D. Kelton, and S. M. Pollock, "Weighted batch means for confidence intervals in steady-state simulations", Management Science, 39, 1002-1019, 1993.  
 [4] Chen, H., "Stochastic root finding in system design", working paper SMS94-8, School of Industrial Engineering, Purdue University, U.S.A, 1994.  
 [5] Fu, M. C., and K. J. Healy, "Techniques for optimization via simulation: an experimental study on (s,S) inventory system", IIE Transactions, 29, 3, 191-200, 1997.  
 [6] Fu, M. C., "Stochastic optimization using simulation: a review", Annals of Operations Research, 53, 199-248, 1994.  
 [7] Fuller, W. A., Introduction to statistical time series, John Wiley and Sons, NY, 1996.  
 [8] Heidergott, B., "Sensitivity analysis of a manufacturing workstation using perturbation analysis techniques", International Journal of Production Research, 3, 611-622, 1995.  
 [9] Hillier, F. S., and G. J. Lieberman, Introduction to Operations Research, 5th ed. McGraw-Hill, 1990.  
 [10] Jacobson, S. H., and L. W. Schruben, "Techniques for simulation response optimization", Operations Research Letters, 8, 1-9, 1989.  
 [11] Law, A. M., and W. D. Kelton, Simulation Modeling and Analysis, McGraw-Hill, 1991.

[12] Lee, Y. H., and K. Iwata, "Part ordering through simulation-optimization in a FMS", International Journal of Production Research, 29, 7, 1309-1323, 1991.  
 [13] Little, J. D. C., "A proof for the queueing formula:  $L = \lambda w$ ", Operations Research, 9, 3, 383-387, 1961.  
 [14] Meketon, M. S., "Optimization in simulation: a survey of recent results", Proceedings of 1987 Winter Simulation Conference, 58-67, 1987.  
 [15] Pierreval, H., and L. Tautou, "Using evolutionary algorithms and simulation for the optimization of manufacturing systems", IIE Transactions, 29, 181-189, 1997.  
 [16] Safizadeh, M. H., "Optimization in simulation: current issues and the future outlook", Naval Research Logistics, 37, 807-825, 1990.  
 [17] Shi, L., and O. Sigurdur, "Nested partitions method for stochastic optimization", Technical Report, Dept. of I. E., University of Wisconsin-Madison, 1997.  
 [18] Voss, P. A., J. Haddock, and T. R. Willemain, "Estimating steady state mean from short transient simulations", Proceedings of the 1996 Winter Simulation Conference, 222-229, 1996.  
 [19] Wild, R. H., and J. J. Pignatiello, "Finding stable system designs: a reverse simulation technique", Communications of the ACM, 35, 10, 87-98, 1994.  
 [20] Yakowitz, S., and E. Lugosi, "Random search in the presence of noise with application to machine learning", SIAM Journal on Scientific Statistical Computing, 11, 702-712, 1990.

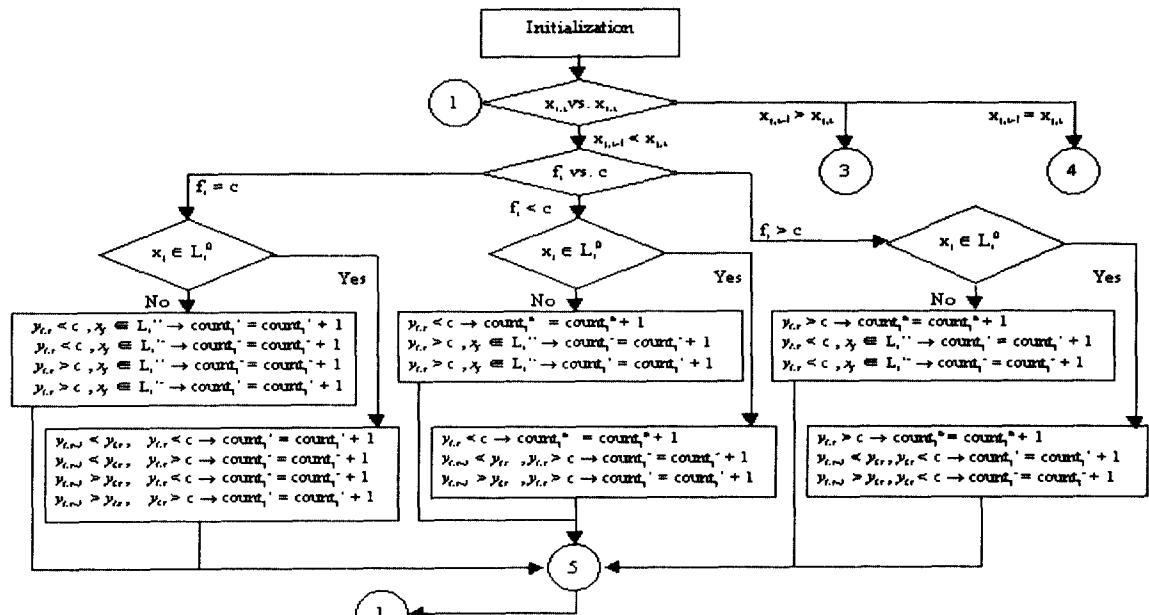


Figure 1 The algorithm to change decision variable (a)

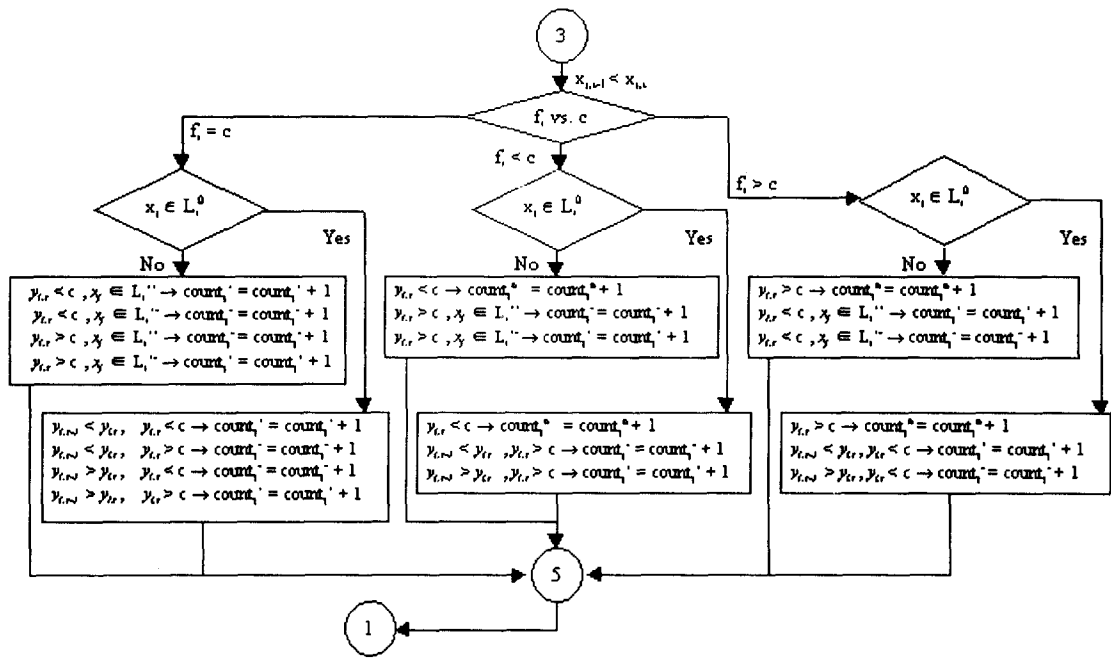


Figure 1 The algorithm to change decision variable (b)

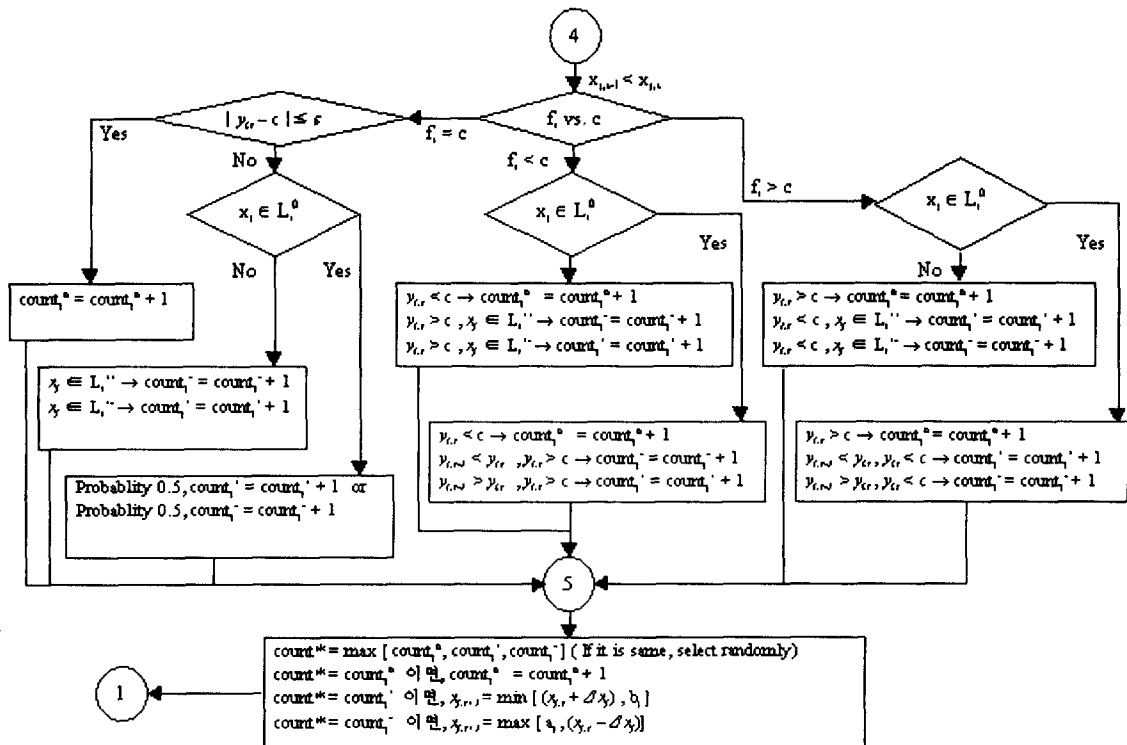


Figure 1 The algorithm to change decision variable (c)