

이산사건 시물레이션에서의 우선순위 큐 성능분석을 위한 다단계 마코브 프로세스 모델: 창조 모델에 대한 사례연구

임동순^{1†}

A Multi-stage Markov Process Model to Evaluate the Performance of Priority Queues in Discrete-Event Simulation: A Case Study with a War Game Model

Dong-Soon Yim

ABSTRACT

In order to evaluate the performance of priority queues for future event list in discrete-event simulations, models representing patterns of enqueue and dequeue processes are required. The time complexities of diverse priority queue implementations can be compared using the performance models. This study aims at developing such performance models especially under the environment that a developed simulation model is used repeatedly for a long period. The developed performance model is based on multi-stage Markov process models; probabilistic patterns of enqueue and dequeue are considered by incorporating non-homogeneous transition probability. All necessary parameters in this performance model would be estimated by analyzing a results obtained by executing the simulation model. A case study with a war game simulation model shows how the parameters defined in multi-stage Markov process models are estimated.

Key words : Priority queue, Discrete event simulation, Markov process

요약

이산사건 시물레이션에서의 미래사건 리스트 관리에 요구되는 우선순위 큐의 성능을 평가하기 위하여 사건의 삽입과 삭제 패턴을 묘사한 성능 모델이 필요하다. 성능 모델을 이용하여 다양한 우선순위 큐 구조를 시간 복잡성 측면에서 비교 평가할 수 있다. 본 연구는 대상이 되는 시물레이션 모델이 반복적으로 운용되고, 실행 시간이 유한적인 경우에 보다 정확한 성능모형을 작성하는 방안을 제시한다. 제안된 성능모형은 다단계 마코브 프로세스 모델에 기반을 두어 확정적인 순서에 의한 삽입과 삭제를 하기 보다는 확률적인 패턴에 의해 연산 순서를 결정한다. 대한민국 육군의 전쟁 연습 모델인 창조 모델을 운영한 결과를 바탕으로 다단계 마코브 프로세스 모델을 작성한 사례연구를 포함하였다.

주요어 : 우선순위 큐, 이산사건 시물레이션, 마코브 프로세스

1. 서론

이산 사건 시물레이션의 구성요소는 크게 모델과 엔진

으로 나눌 수 있다. 특정의 시스템을 표현한 모델은 다수의 이벤트 루틴으로 구성되어 있어 예정된 시각에 사건이 발생되면, 그 사건 발생에 따라 정의된 이벤트 루틴이 실행된다. 그 결과 시스템의 상태가 바뀌고, 미래에 발생할 새로운 사건들이 계획된다. 계획된 미래 사건들은 미래사건 리스트(Future Event List: FEL)에 저장되고, 시물레이션의 엔진은 이러한 FEL을 관리하여 정해진 시간에 정의된 미래 사건들이 발생할 수 있도록 한다. 다음 사건 시각 진전(next-event time-advance) 방식에 의한 엔진은 현재의 시물레이션 시각에서 더 이상의 발생될 사건이 없

* 이 논문은 2008년도 한남대학교 학술연구조성비 지원에 의하여 연구되었음.(2008A118)

2008년 9월 23일 접수, 2008년 11월 18일 채택

¹⁾ 한남대학교 산업경영공학과

주 저 자 : 임동순

교신저자 : 임동순

E-mail; dsyim@hnu.kr

을 때 FEL에 있는 사건 중 가장 빨리 발생되어야 할 사건의 예정된 발생 시각으로 시뮬레이션 시각을 진전 시키고, 그 사건을 발생시킨다. 따라서, FEL에 저장된 미래사건들은 발생될 시각 순에 의하여 차례대로 삭제된다.

대규모 시뮬레이션 모델에서 총 실행 시간의 적지 않은 부분이 FEL의 처리에 소요된다²¹. 시뮬레이션의 실행 시간이 성공적인 프로젝트 수행을 위한 중요한 요소라면 FEL의 처리 시간을 효율적으로 할 수 있는 방안을 강구하는 것이 바람직하다. FEL은 사건들의 발생 시각을 키 값으로 하는 우선순위 큐(priority queue)를 이용하여 구현된다. 우선순위 큐 구조에는 비교적 단순한 연결 리스트(linked list) 뿐 만 아니라, 다소 복잡한 힙 트리(heap tree), 스프레이 트리(splay tree)⁸⁾, 칼렌다 큐(calendar queue)¹¹⁾, MList³⁾ 등 다양한 구조들이 있다. 우선순위 큐 구조들은 두 가지 중요한 연산인 자료의 삽입과 삭제가 어떤 패턴으로 수행되는가에 따라 시간 복잡성이 달라진다. 이러한 연산이 수행되는 패턴은 시뮬레이션 모델에 따라 다를 것이므로 시뮬레이션 엔진의 효율성을 위해서는 FEL를 구현시키기 위한 자료구조를 대상이 되는 모델의 특성에 따라 결정해야 한다.

본 논문에서는 특정의 이산사건 시뮬레이션 모델에서 어떠한 우선순위 큐 구조들이 FEL로서 적절한지를 평가하는 방법론을 제시하려고 한다. 구체적으로는 대상이 되는 시뮬레이션 모델의 특성을 반영한 성능 실험 모델을 생성하는 방안을 소개한다. 제안된 방안은 유효적인 실행 시간을 갖는 시뮬레이션 모델이 장기간에 걸쳐 반복적으로 운용되고, 사건 발생의 특성이 매 운용 마다 거의 동일하게 유지되는 경우에 유용하다. 국방 훈련용 그리고 분석용 모델은 이러한 반복적 운용의 좋은 예가 된다. 이러한 경우 시뮬레이션 실행 결과를 분석하여 모델의 특성을 파악할 수 있고, 이를 바탕으로 가장 적합한 성능 실험 모델을 생성할 수 있다. 본 논문에서는 대한민국 육군의 위게임 모델인 창조 모델을 대상으로 사례연구를 수행한 결과를 제시한다.

2. 성능분석을 위한 실험모델

우선순위 큐 구조의 성능 분석 대상이 되는 시뮬레이션 모델의 특성은 크게 사건의 삽입, 삭제 패턴과 사건의 발생시각인 타임 스탬프(time stamp)의 분포로 구성된다. 이러한 모델의 특성을 반영한 성능 실험 모델 하에서 우선순위 큐 구조의 성능 분석을 위한 실험을 수행하고, 실험 결과를 수집하여야 한다.

성능 분석을 위한 실험 모델은 크게 확정적인 모델과 확률적인 모델로 구분할 수 있다. 확정적인 모델은 사건의 삽입, 삭제 순서가 미리 정해져 있는 모델로 안정적 모델(hold model)과 상하향 모델(up down model)이 있다. 확률적인 모델은 사건의 삽입, 삭제가 확률적으로 수행되는 특성을 가지며, 마코브 과정(Markov process)에 기반을 둔 모델이 이에 속한다.

2.1 안정적 모델

우선순위 큐의 성능 분석을 위한 대부분 실험들은 전통적으로 안정적 모델에 기초하고 있다^{5,6)}. 안정적 모델에서는 초기에 고정된 수의 사건들을 우선순위 큐에 넣은 후 한 사건을 삭제하고 연이어 새로운 사건을 삽입하는 삭제-삽입 과정을 반복한다.

안정적 모델을 이용하여 11개의 우선순위 큐의 성능을 분석한 Jones(1986)의 결과는 큐 크기가 큰 경우에는 스프레이 트리가 우수하고, 10개 이하의 큐 크기에서는 링크드 리스트가 우수함을 나타내었다. Brown(1988)은 그 자신이 제안한 칼렌다 큐를 스프레이 트리와 비교하여 칼렌다 큐가 우수하다고 보고하였다.

2.2 상하향 모델

모든 이산사건 시뮬레이션 모델이 안정적인 특성을 갖고 있지는 않다. 어떤 모델에서는 시뮬레이션 실행 초기에 사건의 삭제가 거의 없고, 삽입만이 반복 수행된다. 최대 큐 크기를 이룬 후에는 이와 반대로 사건의 삽입은 없고, 삭제만이 반복 수행되어 큐 크기가 0에 다다른다. 이러한 성장단계와 쇠퇴 단계는 한 사이클을 이루고, 다수의 사이클이 반복된다. 이러한 실험 모델을 상하향 모델이라고 한다⁷⁾.

2.3 마코브 안정모델

안정적 모델과 상하향 모델은 사건의 삽입, 삭제 순서가 확정되어 있는 특성을 갖는다. 안정적 모델에서는 한 번의 삽입과 한 번의 삭제를 반복 수행하고, 상하향 모델에서는 삽입만을 연속 수행한 후에 삭제만을 연속적으로 수행한다. 마코브 안정 모델은 이러한 삽입, 삭제의 확정성을 탈피하여 확률적으로 연산 순서를 결정하려는 목적에서 개발되었다.

Chung 등(1993)의 마코브 안정 모델은 시스템의 상태를 삽입, 삭제 두 가지로 간주하여 다음과 같은 균일 전이 확률 행렬(homogeneous transition probability matrix)을 도출한다. 이러한 확률을 기반으로 다음 상태를 결정한다.

$$Q = \begin{bmatrix} \alpha & 1-\alpha \\ 1-\beta & \beta \end{bmatrix}$$

만약, 이전의 상태가 삭제이었다면 다음 상태가 삭제일 확률은 α 이고, 삽입일 확률은 $1-\alpha$ 이다. 반면, 이전의 상태가 삽입이었다면 다음 상태가 삭제일 확률은 $1-\beta$ 이고, 삽입일 확률은 β 이다.

안정적 모델에서와 같이 초기에 고정된 수의 사건을 큐에 넣은 후 균일 전이확률에 의해 다음연산(삽입 또는 삭제)이 결정된다.

3. 창조모델의 특성분석

다음 장에서는 대한민국 육군의 전쟁 연습 모델인 창조 모델을 대상으로 우선순위 큐 구조들에 대한 성능 분석을 위해 개발된 실험 모델을 소개한다. 전 장에서 소개한 세 가지 실험 모델이 포함하고 있는 고유의 특성은 창조 모델에 적용하기에 적절하지 않다. 창조 모델의 실행은 종료시점 또는 종료시각을 가지고 있는 유한적인 시뮬레이션에 속한다. 서로 다른 입력으로 인해 반복된 실행마다 서로 다른 결과를 가져오지만, 실행 중 발생하는 사건의 생성 시점과 수는 전체 실행 시간을 통해 유사하다. 때문에, 보다 정확한 성능 분석을 위한 실험 모델을 생성하기 위해 실행 자료로부터 사건의 삽입, 삭제 패턴과 타임 스탬프 분포의 추정이 선행되어야 하고, 이러한 분석 결과를 바탕으로 실험 모델을 개발하여야 한다. 이 장에서는 이러한 일련의 분석 방법과 결과를 설명한다. 분석 대상이 되는 데이터는 시뮬레이션 시각으로 5,300분 동안

의 창조 모델을 실행시킨 결과이다.

3.1 사건 삽입/삭제 패턴

FEL에 삽입되는 사건은 특성에 따라 25 가지의 타입으로 구분되어 총 964,781건의 자료가 발생되었다. 그림 1은 시뮬레이션 시각의 진행에 따른 큐 크기 추이를 나타낸다. 큐 크기가 시뮬레이션 시각 1140분부터 급격히 증가하여 최대 20,997개의 미래 사건이 FEL에 있음을 알 수 있다. 큐 크기의 증가는 시뮬레이션 시간으로 0.4 분 동안에 이루어지고, 그 후 14,500까지 급격히 감소한 후 소강상태를 보인다. 다시 큐 크기는 감소하여 7,290까지 떨어지고, 그 후 1,200까지 완만하게 감소하는 양상을 보인다. 시뮬레이션 시각 2750분 이후로는 큐 크기의 변동이 크지 않는 안정적인 상태를 보인다.

이러한 큐 크기의 변화 패턴은 표 1과 같이 6 가지 단계로 구분할 수 있다. 단계 1에서는 삽입이 대부분 발생하고, 단계 2에서는 삭제가 대부분 발생한다. 단계 3과 단계 6은 삽입과 삭제 횟수가 비슷한 안정 상태로 간주 될 수 있다. 단계 4와 단계 5는 삽입 보다 삭제 횟수가 약간 많다. 단계 4 보다는 단계 5에서 큐 크기가 더 완만히 감소한다. 단계 별 삽입/삭제 연산 횟수로는 단계 6이 전체 연산의 53%를 차지한다. 이는 반 이상의 연산이 단계 6에서 이루어지어 이 단계에서 발생하는 사건의 삽입/삭제 특성이 우선순위 큐 구조의 성능을 좌우 할 것으로 예상된다. 표 2는 25가지 사건 타입에 대한 삽입/삭제 특성을 나타낸다. 사건 타입 3에 속하는 사건 만의 삽입/삭제를 FEL에서 처리한다면 총 사건 발생 수는 6,322이고, FEL에 있는 최대 큐 크기 역시 6,322이다. 이는 6,322개의 사

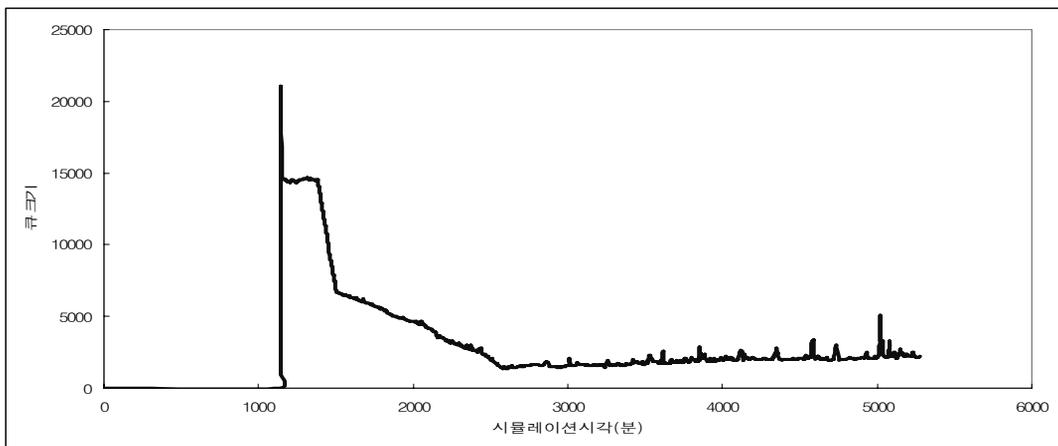


그림 1. 시뮬레이션 시각에 따른 큐 크기 추이

표 1. 단계별 특성

단계	시간(분)	삽입횟수	삭제횟수	총 연산횟수
1	~ 1140.00048	21,255(2%)	3(0%)	21,258(1%)
2	~ 1140.43823	3(0%)	6,658(1%)	6,661(0.3%)
3	~ 1386.00100	95,626(10%)	96,018(10%)	191,644(10%)
4	~ 1504.67000	60,377(6%)	67,933(7%)	128,310(7%)
5	~ 2614.31000	275,500(29%)	280,728(29%)	556,228(29%)
6	~ 5290.00000	512,020(53%)	511,174(53%)	1,023,194(53%)
계		964,781(100%)	962,514(100%)	1,927,295(100%)

건이 연속적으로 삽입되고, 그 후 차례로 삭제되는 상하향 모델(UD)과 동일한 형태를 의미한다. 이와 비슷하게 3가지 사건 타입에 대해 시간에 따른 큐 크기를 나타낸

표 2. 사건 타입별 특성

사건타입	발생수	최대 큐크기	구분
3	6,322	6,322	UD
4	9,975	67	HO
5	357,628	326	HO
7	89,969	140	HO
15	14,210	406	HO
17	9,303	7,373	UD
22	340,068	5	HO
32	436	33	HO
34	4,871	3,450	UD
35	4,987	3,518	UD
36	304	304	UD
72	97,549	2,964	HO
73	134	14	HO
74	50	40	UD
78	837	837	UD
79	177	177	UD
86	4,995	151	HO
90	2,826	101	HO
91	11,526	32	HO
94	785	13	HO
95	418	102	-
107	139	2	HO
131	2,406	32	HO
135	3,685	35	HO
141	847	288	-

그림 2에서 보듯이 사건 타입 17은 삽입과 삭제가 대부분 분리되어 발생하는 상하향 모델의 형태에 근사한 양상을 보인다.

반면, 사건 타입 7 만의 삽입/삭제를 FEL에서 처리한다면 최대 큐 크기는 140으로 사건 발생 수 89,969에 비해 상대적으로 작은 수치를 나타낸다. 이는 삽입과 삭제가 거의 비슷한 확률로 발생하여 큐 크기가 증가하지 않아 안정 모델(HO)에 근사함을 의미한다. 사건 타입 35의 경우는 상하향과 안정적 모델의 특징을 모두 갖고 있는 양상을 보인다. 처음에는 사건 삽입만이 발생되다가 시각 1200에서 3518의 최대 큐 크기에 다다른 후 서서히 사건 삭제가 증가한다. 그 후 2600분 이후부터 삽입, 삭제가 비슷하게 발생하는 안정적인 상태를 나타낸다.

가장 많은 발생을 보이는 사건 타입 5와 22는 각각 전체 사건의 37%와 35%로 이 둘은 전체의 72%를 차지한다. 이 둘 사건들의 큐 크기는 안정상태를 나타내어 전형적인 안정 모델에서의 삽입/사건 패턴과 유사하다.

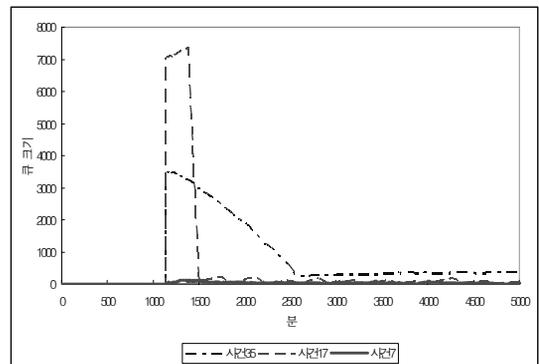


그림 2. 사건 타입 35, 17, 7의 큐 크기 추이

3.2 타임 스탬프 증가시간 분포

새로운 사건이 FEL에 삽입될 때 사건이 가지고 있는 발생 시각인 타임 스탬프를 분석하였다. 타임 스탬프에서 현재 시뮬레이션 시각을 뺀 순 증가시간을 모든 사건에 대해 구한 분포는 그림 3과 같이 대부분의 증가 시간이 10분 이내에 속한다는 것을 나타낸다. 그러나, 단계별로 구분하여 평균 증가시간을 분석한 표 3은 각 단계 별로 조금씩 다른 분포를 가지고 있음을 의미한다. 특히, 단계 1의 경우에는 평균 335분으로 다른 단계에서의 평균 보다 상당히 큰 값을 가진다. 분포 패턴 역시 단계 1과 다른 단계 사이에 상이한 차이가 있음을 나타낸다. 단계 1에서의 분포를 나타낸 그림 4는 증가시간이 0분인 경우가 많고, 그 외 시간은 0부터 1440분 사이에 거의 동등한 빈도수를 가지나, 다만 230분 ~ 280분 사이에 많은 빈도수를 가진다는 것을 나타낸다.

각 단계에서 사건 타입별로 분석한 결과에 따르면 사건 별로 서로 다른 특성을 가지고 있음을 알 수 있다. 단계 1에서 35, 17, 36의 3개 사건 타입에 대해 누적분포를 나타낸 그림 5는 각 사건 타입들이 서로 상이한 분포를 가지고 있다는 것을 잘 보여준다. 따라서 보다 정확한 타임 스탬프 난수 생성을 위해서는 단계와 사건 타입에 따른 분포를 고려해야 함을 의미한다.

4. 다단계 마코브 프로세스 모델에 의한 성능분석

4.1 다단계 마코브 프로세스 모델

전 장의 분석에서 보듯이 창조모델에서는 사건의 삽입, 삭제 패턴의 불확실성이 우선순위 큐 성능 분석의 중요요소로 간주된다. 이에 따라 확률적 모델에서의 마코브 과정에 의한 모델을 기반으로 실험 모델을 생성하였다. 기존의 마코브 안정적 모델은 균일 전이확률에 의존하여 삽입, 삭제 순서를 확률적으로 결정하는 한 단계 만으로 구성된 실험방법이다. 그러나 시뮬레이션 실행 중 발생하는 사건의 삽입, 삭제 패턴은 시간에 따라 다르게 변화한다. 이를 감안하여 본 실험에서는 시뮬레이션의 실행 기간을 특성에 따라 여러 단계로 나누어 각 단계에서의 전이 확률을 다르게 설정하는 방법을 사용하였다. 이를 구체적으로 설명하기 위해 다음과 같은 기호를 사용하기로 한다.

α_j : j 번째 단계에서의 전이확률 P[삭제/삭제]

β_j : j 번째 단계에서의 전이확률 P[삽입/삽입]

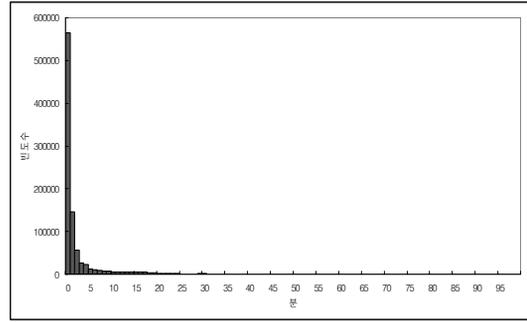


그림 3. 타임 스탬프 증가시간 빈도

표 3. 단계별 평균 타임 스탬프 증가시간

단계	평균 증가시간
1	334.89840 분
2	0 분
3	10.37658 분
4	6.09868 분
5	6.95704 분
6	8.55784 분

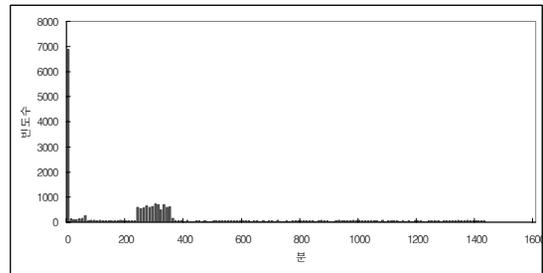


그림 4. 단계 1에서의 타임 스탬프 증가시간 빈도

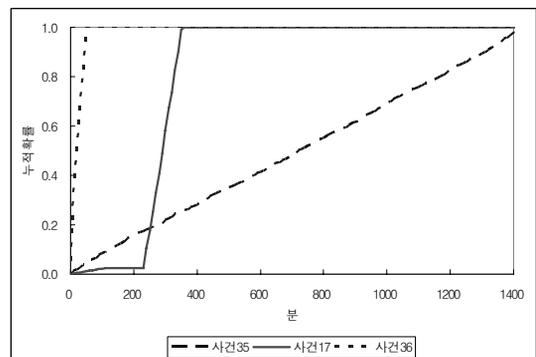


그림 5. 단계 1에서 사건 35, 17, 36의 누적 확률

n_j : j 번째 단계에서의 연산(삽입과 삭제) 수

p_{ij} : j 번째 단계에서 사건이 삽입될 때 이 사건이 i 번째 사건 타입에 속할 확률

$F_{ij}(t)$: j 번째 단계에서 i 번째 사건 타입의 스템프 증가 시간 t 에 대한 누적 확률 분포

실험 모델은 표 1에서 구분된 6단계의 마코브 프로세스 모델에 기반을 두었고, 표 4는 데이터 분석 결과 추정된 각 단계에서의 전이확률과 연산수이다. 전이확률은 각 단계에서 4가지 조건부 상태전이(삽입삭제, 삭제삭제, 삽입삽입, 삭제삽입) 빈도수를 측정하여 평균치로 추정하였고, 연산수는 단계별 특성을 나타내는 표 1의 연산수와 같다.

표 5는 각 사건에 대해 단계별로 추정된 삽입 확률(p_{ij})의 일부이다. 25가지 사건 타입 중 중요하다고 고려되는 5가지 사건 타입만을 나타내었다. 삽입 확률의 의미는 다음과 같다. 예를 들어, 단계 1에서 마코브 프로세스 전이 확률에 의해 한 사건의 삽입이 발생되었다면 이 사건이 사건 타입 3에 속할 확률은 0.2974이다. 이러한 방식으로 사건 타입을 확률적으로 결정하는 것은 사건의 증가 시간 분포가 단계와 사건 타입에 따라 서로 상이함에 기인한다.

증가 시간 분포인 $F_{ij}(t)$ 는 그림 5에서와 같은 누적 분포함수를 단계별, 사건 타입별로 구하여 이를 경험적 분포로 간주하였다.

표 4. 단계별 전이확률과 연산 수

단계	α_j	β_j	n_j
1	0.667	1.000	21,258
2	1.000	0.667	6,661
3	0.604	0.603	191,644
4	0.649	0.605	128,310
5	0.610	0.602	556,228
6	0.640	0.641	1,023,194

표 5. 사건 타입별 삽입확률

단계	사건 3	사건 5	사건 7	사건 17	사건 22
1	0.2974	0.0000	0.0000	0.3316	0.0000
2	0.0000	0.3333	0.3333	0.0000	0.3333
3	0.0000	0.4285	0.1117	0.0054	0.4111
4	0.0000	0.4331	0.1165	0.0025	0.4185
5	0.0000	0.4013	0.0977	0.0021	0.3806
6	0.0000	0.3514	0.0885	0.0020	0.3336

우선순위 큐로 구현된 FEL에서의 사건 삽입과 삭제는 다음의 두 절차에 따른다. 사건들의 우선순위는 미래 발생시간인 타임 스템프에 따르고, 가장 빠른 타임 스템프의 사건이 가장 높은 우선순위를 갖는다.

Procedure: deleteEvent (FEL)

```
delete the highest priority event from FEL;
set clock_time be this event's time of occurrence;
return clock_time;
```

Procedure: insertEvent (FEL, scheduled_time)

```
encode scheduled time in new_event;
insert new_event in FEL;
```

분석 자료를 바탕으로 다단계 마코브 프로세스 모델에 의해 사건의 삽입/삭제 패턴을 시뮬레이션 하는 절차는 다음의 **multiStageMarkovProcessModel**과 같다.

Procedure: multiStageMarkovProcessModel

```
clock_time = 0; j = 1;
state = DELETE; done = FALSE; event_count = 0;
begin total_time measure;
while (not done) do {
  get a U (0, 1) random variate u;
  if (state = DELETE) then {
    if (u <  $\alpha_j$ ) then state=DELETE;
    else state = INSERT;
  }
  else {
    if (u <  $\beta_j$ ) then state=INSERT;
    else state = DELETE;
  }
  if (state = DELETE) then {
    if (FEL is empty) continue;
    clock_time = deleteEvent (FEL);
  }
  else {
    determine an event type i according to  $p_{ij}$ ;
    determine an increment time t according to  $F_{ij}(t)$ ;
    insertEvent (FEL, clock_time + t);
  }
}
```

표 6. 우선순위 큐 평균 실행시간 (단위: 10^{-3} 초)

우선순위큐	단계 1	단계 2	단계 3	단계 4	단계 5	단계 6	계
칼렌다큐(정적)	60.64	4.44	159.88	118.12	401.80	658.96	1403.84
칼렌다큐(동적)	90.08	4.32	173.80	133.72	416.92	669.88	1488.72
힛트리(정적)	36.32	9.32	205.64	156.32	517.52	894.88	1820.00
MList	29.88	53.2	167.44	144.52	467.44	1363.80	2226.28
스프레이트리	64.80	4.40	273.28	199.92	710.76	1195.00	2448.16
힛트리(동적)	35.56	16.92	287.44	197.72	702.88	1210.08	2450.60
링크드리스트	5934.36	3.16	3890.60	3460.72	4305.44	5602.60	23196.99

```

event_count = event_count + 1;
if (event_count =  $n_j$ ) then  $j = j + 1$ ;
if ( $j > 6$ ) then done = TRUE;
}
end of total_time measure;
    
```

4.2 평가대상 우선순위 큐

다단계 마코브 프로세스 모델을 이용한 성능분석의 대상이 되는 우선순위 큐 구조는 링크드 리스트, 스프레이트리, 힛 트리, 칼렌다 큐, 그리고 MList로 한정하였다. 칼렌다 큐는 기본적으로 버킷을 정적인 배열로 정의하여 사용한다. 그러나, 일반적으로 이산 사건 시뮬레이션에서 최대 큐 크기를 미리 예측할 수 없는 특성으로 인하여 배열의 크기를 시뮬레이션 실행 이전에 결정하기는 매우 어렵다. 따라서, 본 연구에서는 버킷의 메모리 할당을 동적으로 하는 경우를 추가하여 실험하였다. 힛 트리에서 노드는 기본적으로 동적인 메모리 할당에 의해 생성된다. 그러나, 칼렌다 큐의 경우와 같이 노드의 최대 크기를 안다면, 노드들을 정적인 배열로 정의하여 사용할 수 있다. 이러한 정적인 배열을 사용한 힛 트리의 성능 분석을 추가 수행하였다. 창조 모델의 분석 결과 최대 큐 크기는 20,997이므로 최대 배열의 크기를 40,000으로 하였다.

스프레이트리는 노드들을 배열이 아닌 동적인 메모리 할당에 의하는 것만을 대상으로 실험하였다. MList는 3단계의 구조로 구성되어 있고, 단계 1과 단계 3은 링크드리스트로 구현된다. 단계 2는 칼렌다 큐와 동일한 구조이다. 본 실험에서는 단계 2에서의 버킷 메모리 할당을 동적으로 하는 경우만을 대상으로 하였다.

4.3 실험결과

실험은 Intel core 2CPU 1.86Ghz의 CPU를 갖는 PC에서 수행되었고, 우선순위 큐 구조들은 JAVA 프로그램

밍 언어를 이용하여 코딩되었다.

각 우선순위 큐 구조에 대해 6 단계로 이루어진 마코브 프로세스 모델에 의한 실험을 25회 반복하여 이에 따른 평균 계산시간을 10^3 초 단위로 측정하였고, 그 결과가 표 6에 나타나 있다. 큐 크기가 급격히 증가하는 단계 1에서는 MList, 큐 크기가 급격히 감소하는 단계 2에서는 링크드 리스트가 상대적으로 우수한 성능을 보인다. 비교적 안정상태에 속하는 단계 3, 4, 5, 6에서는 칼렌다 큐의 성능이 매우 우수한 반면 링크드 리스트의 성능은 가장 열악하다. 이미 언급했듯이 단계 6에서 가장 많은 연산이 이루어지고 있어 이 단계에서의 성능이 우선순위 큐의 전반적인 성능 결정에 매우 중요한 역할을 한다. 안정상태에 속하는 단계 6에서는 칼렌다 큐의 성능이 월등하여 전체적인 성능에서도 칼렌다 큐가 가장 우수하다.

배열을 이용한 정적인 메모리 할당으로 구현된 칼렌다 큐 나 힛 트리가 동적인 메모리 할당을 하는 것에 비해 우수한 성능을 보이는 것은 자연스러운 결과일 것이다. 칼렌다 큐의 경우 정적으로 메모리 할당을 했을 경우에 비해 동적인 메모리 할당은 평균 5.7%의 실행 시간 감소를 가져왔으나, 힛 트리의 경우에는 평균 25.7%의 향상을 가져왔다. 칼렌다 큐에서 메모리 할당은 버킷 수를 증가시켜야 하는 경우에 발생한다. 이러한 버킷 수의 조정은 큐 크기가 증가할 때 가능하다. 그러나, 큐 크기가 안정되어 있는 단계 6과 같은 경우에는 메모리 할당 빈도가 많지 않다. 따라서, 창조 모델에서와 같이 안정 상태가 주가 되는 시뮬레이션 모델에서는 정적인 배열의 칼렌다 큐가 동적인 메모리 할당의 칼렌다 큐에 비해 그다지 큰 향상을 가져다주지 못한다.

향후 창조 모델의 구성요소가 병력 및 무기의 추가 등으로 인해 확장되어, 시뮬레이션 실행 중 발생하는 사건이 증가될 수 있는 가능성이 있어 이를 고려한 부가적인 실험을 수행하였다. 즉, 이전 실험 모델에서의 6단계 특성

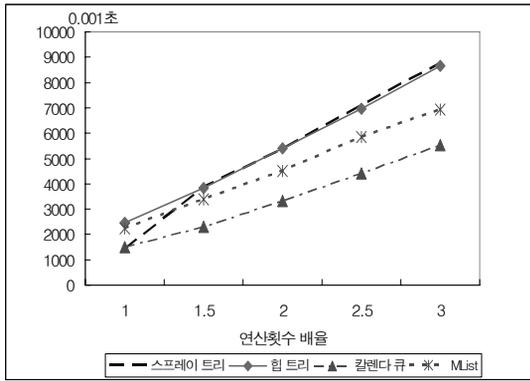


그림 6. 연산횟수 증가에 따른 평균 실행시간

과 사건 타입의 비율은 그대로 유지된다는 가정 하에서 각 단계에서의 연산 횟수를 현재의 1.5배, 2배, 2.5배, 3배로 하여 이에 따른 우선순위 큐의 성능을 분석하였다. 정적인 배열을 사용하는 우선순위 큐들과 링크드 리스트는 분석에서 제외하였다. 그림 6의 결과는 칼렌다 큐, MList, 힙 트리, 스프레이 트리의 순으로 우수함을 나타낸다. 그러나, 1.5배 이상의 연산 횟수에서 스프레이 트리와 힙 트리는 거의 동일한 실행시간을 유지하여 두 구조 간에 큰 차이가 없음을 나타낸다. 칼렌다 큐는 가장 적은 실행시간이 소요되고, 더욱이 연산 횟수의 증가에 따라 스프레이 트리, 힙 트리와 격차가 더욱 커지는 특성을 보여준다.

5. 결 론

본 논문에서는 이산사건 시뮬레이션 모델을 대상으로 시뮬레이션 실행 속도를 향상시키기 위한 방안을 연구하였다. 구체적으로는 시뮬레이션 엔진의 중요한 부분을 차지하는 미래사건 리스트의 관리에 초점을 맞추어, 다양한 우선순위 큐의 성능을 비교하는 실험 모델을 제안하였다. 특히, 대한민국 육군의 전쟁 연습모델인 창조 모델을 응용한 결과를 바탕으로 가장 효과적인 FEL 구조를 제안하는데 역점을 두었다.

창조 모델과 같은 국방 훈련, 분석용 모델은 한 번의 실행으로 모델의 수명이 다하는 것이 아니라 장기간에 걸쳐 반복적으로 운용되는 특징을 갖는다. 더욱이 종료시점 또는 종료시각을 갖는 유한 시뮬레이션으로 실행 마다 유사한 사건 발생 패턴을 나타낸다. 이러한 특성의 모델에서는 실험 결과를 분석하여 가장 효율적인 미래사건 리스트 관리를 도모하는 것이 바람직하다.

이러한 목적에서 본 연구는 창조 모델을 대상으로 다단계 마코브 프로세스 모델에 기반을 둔 실험 모델을 개발하였으며, 각 단계에서의 전이확률과 사건 타입별 타임스탬프 증가시간을 추정하였다.

개발된 다단계 마코브 프로세스 모델로 수행된 실험 결과, 삭제 보다는 삽입이 월등히 많이 발생하는 경우에는 칼렌다 큐의 성능이 저조하고, 힙 트리의 성능이 우수하였다. 삽입 보다 삭제가 많이 발생하는 경우에는 대체로 스프레이 트리의 성능이 우수하였다. 그러나, 삽입, 삭제가 거의 동등하게 발생하는 안정상태에서는 어느 우선순위 큐도 칼렌다 큐의 성능을 추월하지 못하였다.

창조 모델을 대상으로 한 실험에서는 대체로 칼렌다 큐의 성능이 우수함을 나타내었다. 이는 창조 모델의 실행 특성상 사건의 삽입, 삭제가 동등하게 발생하는 안정상태의 비중이 삽입만이 발생하는 상태와 삭제만이 발생하는 상태에 비해 월등히 크다는 것에 기인한다.

참 고 문 헌

1. Brown, R. (1988), "calendar queues: A fast O(1) priority queue implementation for the simulation event set problem", Comm. of the ACM, Vol. 24, No. 12, pp. 825-829.
2. Chung, K., Sang, J. and Rego, V. (1993), "A performance comparison of event calendar algorithms: an empirical approach", Software-Practice and Experience, Vol. 23, No. 10, pp. 1107-1138.
3. Goh, R.S.M. and Thng, I.L. (2003), "MLIST: An Efficient Pending Event Set Structure for Discrete Event Simulation", International J. of Simulation, Vol. 4, No. 5/6, pp. 66-77.
4. Henriksen, J. O. (1977), "An improved event list algorithm", in Proceedings of the 1977 Winter Simulation Conference, pp. 547-557.
5. Jones, D. W. (1986), "An empirical comparison of priority-queue and event-set implementations", Comm. of the ACM, Vol. 29, No. 4, pp. 300-311.
6. McCormack, W. M. and Sargent, R. G. (1981), "Analysis of future event-set algorithms for discrete event simulation, Comm. ACM, Vol. 24, No. 12, pp. 801-812.
7. Ronngren, R. and Ayani, R. E. (1997), "Parallel and sequential priority queue algorithms", ACM Trans. On Modeling and Computer Simulation, Vol. 2, pp. 157-209.
8. Tarjan, R. E. and Sleator, D. D. (1985), "Self-adjusting binary search trees", J.of ACM, Vol. 32, No. 3, pp. 652-686.



임 동 순 (dsyim@hnu.kr)

1983 한양대학교 산업공학과 학사
1986 한국과학기술원 산업공학과 석사
1991 아이오와 주립대학 산업공학과 공학박사
1992~현재 한남대학교 산업경영공학과 교수

관심분야 : 모델링&시뮬레이션, 네트워크 알고리즘