

# 시뮬레이션과 유전 알고리즘의 하이브리드 기법을 이용한 정보시스템 용량 산정 및 선택 방안

†민재형\* · 장성우\*\* · 신경식\*\*\*

## A Hybrid Approach to Information System Sizing and Selection using Simulation and Genetic Algorithm

†Jae H. Min\* · Sungwoo Chang\*\* · Kyung-shik Shin\*\*\*

### ■ Abstract ■

The purpose of this paper is to develop a new method for information system sizing and selection based on a hybrid mixture of simulation and genetic algorithm, and to show its cost-effectiveness by applying it to a real world problem. To serve this purpose, we propose an operational model which identifies a set of system alternatives using simulation, and determines the optimal one using genetic algorithm. Specifically, with simulation, we generate probability distributions describing real data gathered from actual system, which can overcome the major weakness of the existing methodology that normally employs point estimates of the actual data and constant correction factors without theoretical rationale. We next search for the optimal combination of H/W, the number of CPUs, and S/W, which meets both of our business goals of incurring low TCO (total cost of ownership) and maintaining a good level of transaction processing performance. Experimental result shows the proposed method in this paper saves the cost while it preserves the system's capacity within allowable performance range.

Keywords : System Sizing and Selection, Simulation, Genetic Algorithm, Hybrid Approach, TCO

논문접수일 : 2007년 08월 06일    논문게재확정일 : 2007년 11월 03일

\* 서강대학교 경영학과

\*\* 한국 오라클

\*\*\* 이화여자대학교 경영학과

† 교신저자

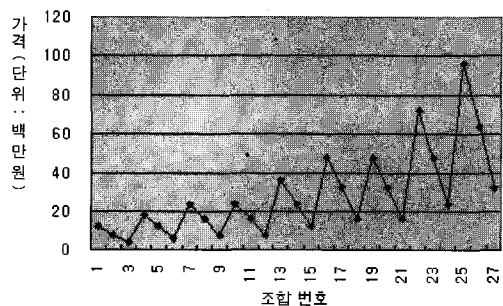
## 1. 서 론

최근 정보 기술의 발달에 힘입어 기업들은 비즈니스 문제 해결을 위하여 다양한 정보시스템을 구축하고 있으며, 기업마다 정도의 차이는 있지만 상당한 비용을 정보시스템 도입에 지출하고 있는 상황이다. 그러나 막대한 비용이 지출되는 정보시스템 구입의 실상을 들여다보면 어느 정도가 정보시스템의 적정한 규모인지를 논리적으로 평가하지 않은 채, 경험과 관습에 기반을 둔 추상적인 판단에 근거해서 정보 시스템의 용량을 산정하고 있음을 알 수 있다. 예를 들면, 일반적으로 정보시스템을 선정하기 위해서는 우선, 시스템을 사용할 당시 사용자의 수, 1인당 평균 처리 업무의 수 등을 기본 데이터로 하여 기본 용량을 산정한 후, 여기에 밀집 시간(peak time) 여유율, 시스템 여유율, 애플리케이션 여유율 등의 다양한 보정계수(correction factor)를 곱해 줌으로써 도입 시스템의 용량을 산정한다[4]. 하지만 이러한 일반적인 정보시스템 용량 산정 방법론에서 사용되는 기본 데이터가 정확한 계산이 아닌 추정으로 산출되고 있고, 보정 계수 또한 이론적인 근거가 없는 경험적인 상수로 이루어져 있어, 많은 비용이 소요되는 정보시스템의 선정 및 도입 작업이 과학적인 근거가 미약한 상태로 이루어지고 있음을 알 수 있다.

또한 적절한 정보시스템의 선택을 위해서는 기본 성능 요구 조건과 함께 가격 조건을 동시에 고려하여야 한다. 즉, 적절한 시스템 성능 요구 조건을 만족하는 대안 시스템들 중에서 저렴한 총 소유 비용(이하 TCO : Total Cost of Ownership)을 지불하면서 동시에 우수한 성능을 보이는 시스템을 구매하여야 하는, 성능과 가격의 상충 관계(trade-off)를 감안한 의사결정을 수행하여야 한다. 여기서, 중요한 특기 사항은 일반적인 비즈니스 환경에서는 정보시스템의 가격 조건이 비선형적인(non-linear) 특성을 보인다는 점이다. 즉, 정보시스템의 기본적인 판매가(list price)는 회사마다 정해져 있으나 실제 납품 가격을 결정짓는 할인율(discount

rate)은 전략적인 판단에 따라 수시로 달라지게 된다. 또한 시스템의 성능을 좌우하는 주요 요소인 CPU(central processing unit)에 듀얼 코어(dual core) 기술이 적용됨에 따라 각 코어에 대한 가격 적용 비율이 S/W 업체마다 서로 달리 적용되고 있다. 예를 들면, 오라클(Oracle)의 경우 Intel Itanium과 AMD의 코어에는 0.5, IBM Power Series에는 0.75, 나머지는 1을 적용하지만, IBM은 다른 비율을 적용하는 정책을 가지고 있으며, Microsoft는 코어 개수에 상관없이 CPU 개수만 계산하는 정책을 유지하고 있다[7, 9, 10].

가격의 비선형적인 변화 특성을 좀 더 구체적으로 설명하면 다음과 같다. 먼저 가격은(CPU당 판매가)\*(CPU 개수)\*(CPU 종류에 따른 가중비율)\*(1-할인율)에 의해 산정된다. 여기서, CPU당 판매가는 업체별로 고정되어 있지만 이를 제외한 나머지 3개의 값은 상황에 따라 변하는 변수가 된다. 즉, (1) 실제 비즈니스 환경에서는 성능 요구 사항을 함께 검토해야 하기 때문에 CPU 개수를 사전에 결정할 수 없고, (2) 일반적으로 S/W와 H/W는 독립적으로 구매하기 때문에 어떤 종류의 CPU를 가지는 서버를 구입할지를 S/W 가격 산정 시 사전에 알 수 없으며, (3) 할인율은 영업 상황에 따라 전략적으로 벤더사에 의해 제안되기 때문에 이 또한 사전에 고정될 수 없으므로 가격의 변화는 비선형성을 띄게 된다.



〈그림 1〉 가격의 비선형적인 변화

예를 들어, 오라클 DBMS를 구입한다고 가정해

보자. 상황을 간단히 하기 위해 CPU당 판매가는 2천만원이고, CPU 개수는(2, 4, 8), CPU 종류에 따른 가중비율은(Intel = 0.5, IBM = 0.75, Other = 1), 할인율은(0.4, 0.6, 0.8) 중 하나의 값을 갖는다고 하면, 이러한 상황에서 가능한 가격의 분포는 <그림 1>과 같이 비선형적으로 변화됨을 알 수 있다. <그림 1>에서 횡축은 CPU 개수, 가중비율, (1-할인율) 등 세 가지 변수의 조합을 나타내는 번호로 위 예의 경우, 모두 27개의 조합이 있으며, 종축은 각 조합에 상응하는 가격을 나타낸다.

이와 같이 상황에 따라 가변적이고 복합적인 가격결정체제와 성능/가격의 상충 관계로 인해 적정 시스템의 선정은 더욱 어려운 문제로 대두되고 있는데, 이를 해결하기 위한 의사결정지원방안의 제공이 절실한 상황이다. 이러한 문제를 해결하기 위하여 본 연구에서는 시뮬레이션(simulation)과 유전 알고리즘(genetic algorithm)을 복합적으로 사용하여 효율적인 정보시스템 용량 산정 및 적정 솔루션 선택 방안을 제시하고자 한다.

구체적으로, 우선 시뮬레이션 기법을 이용하여 정보시스템이 갖추어야 할 적정 성능의 범위를 선정하고, 이를 만족시키는 가능한 대안들을 판별한다. 여기서 중요한 것은 기존의 정보시스템 용량 산정 시 관습적으로 사용하던 보정 계수가 아닌 실제 발생한 데이터에 기반을 둔 확률분포를 사용한다는 점이다. 즉, 상수 대신에 확률변수를 사용하여 현실적으로 발생할 수 있는 값의 변이를 수용한다는 점이 기존의 용량 산정 방법과 다른 점이 된다. 다음으로, 유전 알고리즘을 이용하여 주어진 성능 범위 내의 대안들 중에서 다양한 비즈니스 목표 및 제약조건을 만족시키는 최적의 시스템을 선정하는 것이다. 앞서 지적한 바와 같이 가격결정구조가 할인율과 가격정책의 조합에 의해 비선형적인 특성을 가지기 때문에 본 연구에서는 LP모델이 아닌 유전 알고리즘을 이용하여 최적 대안을 선정하는 접근방법을 선택하였다. 본 연구에서는 정해진 예산을 초과하지 않는 범위 내에서 TCO가 적으면서 동시에 좋은 성능을 가지는 시스템을 선정

하는 것을 비즈니스 목표로 하는데, 그 이유는 실제 비즈니스 환경에서 이러한 기준을 가지고 정보 시스템을 선정하기 때문이다.

## 2. 관련 연구

정보시스템 용량 산정에 관한 연구는 상당히 미흡한 실정이지만 그 중 대표적인 것으로는 한국전산원에서 수행한 연구[4,5]를 들 수 있다. 한국전산원의 초기 연구[5]에서는 과거 정보화 사업의 사례 및 국내 SI 업체의 산정 기준을 참조하여 정보시스템 구축 비용 중에서 가장 중요한 CPU, 메모리, 디스크의 용량 산정 방식과 절차를 제시하고 있다. 그러나 이 연구는 사내(한국전산원 내)에서 정보화 사업의 검토와 예산 조정에 활용할 목적으로 작성되었기 때문에 다른 자료와 차이가 나는 부분이 많아, 후속 연구[4]에서는 공공부문 정보화 사업 시 실질적인 지침이 될 수 있도록 용량 산정식을 고도화 하고 용량 산정 항목 및 적용 값에 대한 적정성과 객관성을 확보하고자 노력하였다. 하지만, 앞서 지적한 것처럼 두 연구 모두 적용 값을 상수로 규정하고 있어 실제 운용 데이터의 변이를 반영하지 못한다는 한계점을 가지고 있다. 정해용 등[3]에서는 이러한 상수 적용의 문제점을 보완하고자 탐색적 방법을 통해 용량 산정에 사용되는 보정 계수들의 타당성을 확보하고자 시도하였다. 하지만 이 방법 역시 각 시스템마다 독특하게 가지고 있는 요구조건의 변화 분포를 정확하게 반영해 주지 못한다는 한계를 여전히 가지고 있다.

본 연구에서는 이러한 기존 연구의 한계를 극복하기 위하여 시뮬레이션을 활용하고자 한다. 시뮬레이션은 실제 또는 가상 시스템의 행위를 컴퓨터를 이용하여 모사(imitation)하는 작업을 말하며, 다양한 분야의 불확실성을 내포하는 문제상황을 컴퓨터 상에서 기술하는데 활용되고 있다. 앞서 지적한 바와 같이 기존의 정보시스템 용량 산정 방법은 주요 변수에 대해 경험적으로 추정된 하나의 값만을 고려하는 확정적 접근 방법(deterministic ap-

proach)인데 반해, 본 연구에서는 시뮬레이션을 이용하여 주요 변수가 취할 수 있는 모든 값과 그 분포를 고려하는 확률적 접근 방법(stochastic approach)을 취한다.

다음으로 유전 알고리즘은 자연선택과 유전 매커니즘에 기반한 탐색 알고리즘으로서 생태계의 선택(selection), 교배(cross over), 돌연변이(mutation) 등을 수리적으로 모형화한 방법이다. 유전 알고리즘은 확률적 검색 방법을 이용하여 다양한 제약조건을 포함한 문제상황에서 목적함수를 최적화하는 모수(parameter) 값을 추정하는데 널리 활용되고 있다. 특히, LP모델과 달리 최적해를 구하는데 있어 모델의 선형성을 가정하지 않는 것이 큰 차이점이 된다. 본 연구에서는 적정 정보시스템의 선정에 대해 가격 모델의 비선형성에 대한 고려가 필요하기 때문에 유전 알고리즘에 기반한 탐색 방법을 이용하여 가격과 성능의 상충관계를 고려한 최적 시스템을 선정하고자 한다.

한편, 현실 세계에 존재하는 다양한 불확실성을 수용하면서 동시에 최적해를 찾기 위한 방법으로 시뮬레이션과 유전 알고리즘을 결합하여 의사결정을 수행하는 방법이 기존 연구에서 드물게 제시되고 있지만 아직도 특정 분야에 한정된 상황이다. 예를 들어, Fujimoto et al.[6], Jeong et al.[8], 이용균, 김경섭[2]은 생산 스케줄링 문제 해결을 위해 두 기법을 통합적으로 사용하였고, Suiadee and Tingsanchali[11]는 저수지의 최적 운용 규칙을 결정하기 위해 통합기법을 사용하였다. 하지만 정보시스템의 용량 산정 및 최적 대안 선정에 관한 방법으로는 아직까지 적용된 사례가 없는 실정이다.

### 3. 연구모형

#### 3.1 문제의 정의

본 연구에서 해결하고자 하는 문제는 “TCO가 적으면서 동시에 적정 성능을 가지는 정보 시스템을 구매”하는 것이다. 여기서, TCO는 H/W, S/W

모두의 구매가와 유지 보수 비용의 합으로 계산된다. 구매가는 규정 판매가와 할인율의 곱으로 계산된다. S/W의 경우는 규정 판매가가 단위 CPU당 가격으로 정의된다. 따라서 S/W의 규정 판매가는 S/W가 사용되는 H/W의 CPU 개수와 단위 CPU당 가격의 곱으로 결정된다. 그런 후 여기에 할인율이 곱해져서 최종 구매가가 결정된다. 유지 보수 비용은 최종 구매가에 유지 보수 요율을 곱해서 구한다. 이와 같은 과정을 거쳐 TCO를 계산하고, TCO가 가장 적은 시스템을 구입하는 것이 기업의 이익에 도움이 되는 것은 당연한 이치이다.

하지만 이러한 최소 TCO 기준은 선정하고자 하는 시스템이 필요한 성능 요구 조건을 적절히 만족한다는 전제 하에서만 의미를 가진다. 특정 시스템이 적정 성능을 가지고 있는지를 정확히 판별하기 위해서는 어떠한 기준으로 시스템의 성능을 평가할 것인지를 결정하는 것이 중요한 이슈가 된다. 즉, 정보시스템의 성능을 객관적으로 입증할 수 있는 성과지표가 필요한데, 본 연구에서는 TPC(Transaction Processing Performance Council)에서 발표하는 TPC-C 벤치마크 성능 결과인 tpmC(Transactions per Minute by TPC-C)[12]를 성과지표로 선택하였다. TPC는 대표적인 성능평가 기관으로 원래 DBMS를 운용하는 시스템의 성능을 측정하기 위해 1988년 8월 설립되었는데, 이후 애플리케이션 서버 및 웹 서버의 성능 측정까지 그 영역을 확대하고 있다. TPC에서 발표하는 성능 벤치마크로는 TPC-C, TPC-H, TPC-E 등이 있으며, 이 중 TPC-C 벤치마크는 OLTP 시스템의 성능 평가 기준으로 업계에서 준거(reference) 자료로서 많이 활용하고 있기 때문에 본 연구에서도 이를 성과지표로 사용하기로 한다.

다음으로 위의 문제에서 고려해야 할 제약조건을 기술하면 다음과 같다. 첫째, 기본적인 성능 요구 조건을 만족시켜야 한다는 것이다. 즉, 정보시스템이 수행해야 할 업무를 고려하여 산정되는 적정 성능 범위(최소 및 최대 tpmC 범위) 내에 해당 정보시스템의 공인 tpmC값이 위치해야만 한다. 둘

제, 다양한 H/W 및 S/W의 조합을 고려할 수 있어야 한다는 점이다. 실제 업무에서는 H/W와 S/W가 상호 종속되지 않고 고객의 판단에 따라 적정 H/W와 S/W를 조합해서 사용하게 되는데 이를 감안한 대안 집합을 제시할 수 있어야 한다. 셋째, 확장성을 감안한 선택을 수용할 수 있어야 한다. 일반적으로, 시스템 선택 시 주어진 성능 범위 내에 속하는 정보시스템을 그대로 쓸 수도 있고, 향후 확장성을 고려하여 보다 높은 성능을 가지는 시스템을 CPU를 줄여서 도입하기도 한다. 따라서 이러한 상황을 의사결정에 반영할 수 있어야 한다. 마지막으로, 규정 가격 대비 실제 공급 시 할인율 및 솔루션별 가격 정책을 반영할 수 있어야 한다. 서론에서 언급한 것처럼 일반적으로 납품 가격은 미리 규정된 반면 상황에 따라 최종 할인율이 달라지고, 여기에 S/W의 경우는 회사마다 서로 다른 듀얼 코어 가격 정책[7,9,10]을 가지고 있는데, 이를 TCO 산정 공식에 반영할 수 있어야 한다.

이와 같은 사항들을 고려하여 수립한 논리적 모형은 <표 1>과 같다.

앞서 서론에서 언급한 바와 같이 <표 1>의

TCO를 산정하는 식은 3개 변수(CPU 개수, CPU 종류에 따른 가중치, 전략적 할인율)의 곱을 포함한 비선형 식으로 표현되므로 이 모델은 LP로 해결할 수 없게 된다.

### 3.2 기존 방법의 한계

현재 다양한 정보시스템 용량 산정 방법이 사용되고 있지만, 본 연구에서는 한국전산원에서 제시한 정보시스템 용량 산정 기술[4]을 기본 틀로 사용하기로 한다. 이 연구에서 제시한 정보시스템 용량 산정 방안을 요약하면 <표 2>와 같다.

<표 2>에 나타난 기존 방안의 주요 문제점을 기술하면 다음과 같다. 첫째, 주요 인자 및 보정 계수가 상수로 이루어져 있는데, 이들 값은 논리적으로 설명할 수 없는 경험치 혹은 가정치일 뿐이며 이론적, 현실적 합리성에 근거하지 않는다는 점이다. 이로 인해 용량 산정 결과 역시 단지 추정치일 뿐 실제 비즈니스 현황에 근거를 둔 과학적 계산 값이 아니라는 한계가 있다. 따라서 단순 추정치와 실제 시스템 현황으로부터 획득한 분포 정보의 차이만

<표 1> 논리적 모형

<p>의사결정변수 : 선택하고자 하는 H/W 서버 및 포함된 CPU 개수, 그리고 S/W 제품</p> <p>(1) HW : 선정하고자 하는 H/W 서버 모델</p> <p>(2) CPU : 선정된 H/W 내에서 사용할 CPU 개수</p> <p>(3) SW : 선정하고자 하는 S/W 제품</p> <p>목적함수 : TCO를 최소로 하면서 동시에 성능을 최대화</p> <p>Max (-TCO + Perf)</p> <p>여기서, <math>TCO = (HW\_TCO + SW\_TCO)</math></p> <p><math>HW\_TCO = (HW\_price - (Max\_CPU - CPU) * CPU\_price) * (1+SR)</math></p> <p><math>SW\_TCO = (SW\_price * CPU * Factor * (1-DC)) * (1+SR)</math></p> <p>Factor : CPU 종류별 가중치, DC : discount rate, SR : support rate</p> <p>Perf : 해당 HW가 TPC-C 결과에서 가지는 tpmC 값</p> <p>제약조건</p> <p>(1) 성능 만족 조건 : <math>a \leq Perf \leq \beta</math>, <math>a, \beta</math> : 요구 성능 최소 및 최대 tpmC값</p> <p>(2) 예산 만족 조건 : <math>TCO \leq Budget</math></p> <p>(3) 확장성 고려 조건</p> <p>(a) CPU 개수는 HW가 제공하는 최대 CPU 개수보다 작아야 함</p> <p><math>0 \leq Max\_CPU - CPU</math></p> <p>(b) CPU 개수는 HW가 제공하는 최대 CPU 개수의 절반보다는 크거나 같아야 함</p> <p><math>Max\_CPU / CPU \leq 2</math></p>
--

<표 2> 기존의 정보시스템 용량 산정 방안

구 분	산정 근거	산정 사례
일 트랜잭션 처리 수	사용자수 * 개인입력건수/일 * (매 3년 사용자 증가 보정율 : 30%)	$1,500 * 3 * (1.3 * 1.3 * 1.3) = 9,887$ 건/일
일일 업무집중시간	일일 처리량의 70%를 4시간 사이 처리 가정 분당 트랜잭션(transactions)	$9,887 * 0.7 / 4 / 60 = 29$ 건/min
트랜잭션 복잡도	TPC와 트랜잭션 복잡성 차이(1.5배)	$29 * 1.5 = 44$ /min
응답시간 보정	LAN/WAN을 통한 클라이언트 응답 시간 보정(1.5배)	$44 * 1.5 = 66$ /min
Peak Time 보정	Peak Time시 이를 위한 6배의 보정을 적용	$66 * 6 = 396$ /min
시스템 부하율	OS(50%), 기타 시스템 S/W(DB40%, 기타 40%)	$396 * 1.5 * 1.4 * 1.4 = 1,164$ /min
I/O 부하율	I/O 부하율 70%	$1,164 * 1.7 = 1,979$ /min
시스템 여유율	40% 정도의 시스템 여유율	$1,979 * 1.4 = 2,771$ /min
트랜잭션 증가율	3년간 트랜잭션 증가율(30%)	$2,771 * 1.3 * 1.3 * 1.3 = 6,088$ tpmC

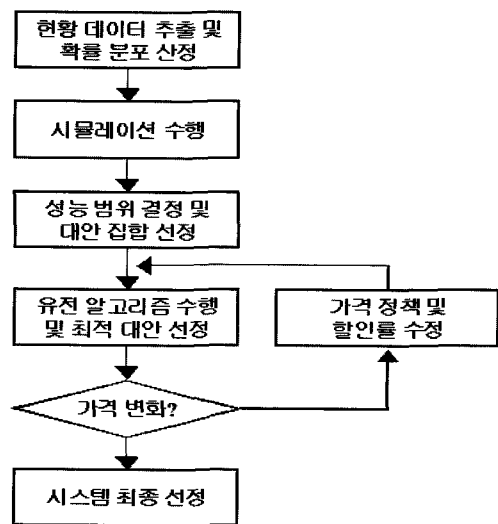
컴 오류가 발생하여, 추정치가 클 때에는 과도 용량 산정으로 인한 과다 비용 계상이라는 문제가 발생할 수 있으며, 추정치가 작을 때에는 용량 산정 부족으로 인한 향후 성능 문제 발생 가능성을 안게 되는 것이다.

둘째, 기존 방법의 경우 용량 산정 값이 단일 수치로 나오지만 실제로는 이를 평균으로 하여 그 주변 적정 영역을 가능한 성능 범위로 활용하게 되며, 보통은 제시된 용량 산정 값보다 좋은 성능을 가지는 시스템을 구매하게 된다. 하지만 어디까지가 수용 가능한 적정 범위인지를 객관적으로 제시하지 못하게 되어 결국 적정 성능 범위 역시 직관에 의존하여 결정할 수 밖에 없게 된다.

셋째, 기존의 용량 산정의 경우 최종적인 시스템 선정을 위한 의사결정 방안이 함수로 표현되고 있지 않다는 점이다. 따라서 다양한 가격 관련 기준과 성능의 상충관계(trade-off)를 함수로 제시하지 못하고 수행자의 주관적, 인지적 판단에만 의존하게 됨으로써 제시된 성능 값에 매핑 되는 시스템만 선택할 뿐, 경쟁상황에 따라 변하는 할인율, 듀얼 코어 기반 가격정책, 대용량 시스템에서 CPU를 줄여 구매하는 경우 등 다양한 비즈니스 환경을 감안한 시스템 선정 의사결정을 지원하지 못하고 있다는 점이다.

### 3.3 하이브리드 수행 방안

기존 방법의 문제점을 해결하기 위한 방안으로 본 연구에서는 시뮬레이션과 유전 알고리즘을 결합하여 정보시스템의 용량을 산정하고 최선의 정보 시스템을 선택하는 방법을 제안한다. 이러한 하이브리드(hybrid) 방법의 전체적인 수행 프로세스를 도시하면 <그림 2>와 같다. <그림 2>의 각 단계를 설명하면 다음과 같다.



<그림 2> 정보시스템 용량 산정 및 선택 프로세스 개요

3.3.1 사전 단계 : 사용 데이터 준비

TPC는 주요 정보시스템을 대상으로 성능 테스트를 수행하여 성능 결과를 공인하는 기관이다. 이 기관의 TPC-C 벤치마크 성능 결과는 성능 테스트를 위한 다양한 질의를 이용하여 처리된 분당 트랜잭션 수를 성능 값으로 정의하고, 이러한 성능을 내기 위해 조합한 H/W, S/W, Disk의 총 실제 구매 비용까지 알려 주고 있다. 이러한 TPC-C 성능 보고서는 크게 H/W 및 S/W 벤더, 서버 모델 및 구조, CPU 정보, M/M 크기, 디스크 크기 등과 같은 기본 정보와 TPC가 공인한 성능 척도로서의 tpmC 정보, 그리고 이러한 성능을 내기 위해 구성된 주요 H/W 및 S/W의 가격 정보 등을 포함하고 있다. 여기서 가격 정보는 기본 납품 가격 및 유지 보수 비용을 포함하고 있다.

이 외에도 종합적인 의사결정을 위해서는 할인율 및 듀얼 코어 가격 정책이 필요한데, 최종 할인율은 제품 납품 직전에 최종적으로 제안되므로 이는 사용자 입력으로 지정하고, 듀얼 코어 가격 정책은 S/W 제품 회사 홈페이지에 공시되어 있는 것을 일반적으로 이용한다[7, 9, 10].

본 연구에서는 이러한 정보들 중 <표 3>의 데이터들을 의사결정에 활용하기로 한다.

<표 3> 의사결정에 사용되는 데이터 내역

TPC-C 성능 정보	H/W 정보	벤더명, 모델명, CPU 타입 및 개수, 성능 정보, 납품 가격, CPU 가격, CPU 축소 가능성
	S/W 정보	벤더명, 제품명, 단위 가격
개별 정보	H/W 정보	다양한 CPU 타입 정보
	S/W 정보	듀얼 코어 가격 정책

3.3.2 단계 1 : 시물레이션을 이용한 대안 범위 선정

시물레이션의 수행 목표는 적정 tpmC 범위의 설정 및 이를 만족하는 대안 시스템의 집합을 판별하는 것이다. 본 연구에서는 시물레이션 수행을 위해 @Risk를 이용하였다. 시물레이션의 수행을 위해서는 먼저 확률변수를 지정하여야 하는데, 본 연구에서는 기본 데이터인 “동시 접속자 수”와 “접속자가 실행한 트랜잭션의 수”, 보정 계수인 “밀집 시간 여유율” 등을 확률변수로 지정한다. 그리고 분석 SQL을 통해 주요 확률변수의 실제 현황 데이터를 기존 정보시스템에서 추출한 후, 이들을 이용하여 적합한 대응 확률분포를 결정한다. 정보 추출 SQL의 예제는 <표 4>와 같다.

본 연구에서는 확률변수의 분포를 결정하기 위

<표 4> 정보 추출 SQL의 예제

```

select c.instance_number inst_no, to_char(c.snap_time, 'mmdd hh24mi') time,
       round(decode((b.snap_time-a.snap_time) * 60 * 24, 0, 0, (b.trans-a.trans)/ ((b.snap_time-a.snap_time)
* 60 * 24)), 0) "Trans/Min",
       b.trans - a.trans "trans", (b.snap_time-a.snap_time)*60*24 "Min"
from (select s.snap_id, sn.snap_time, s.dbid, s.instance_number
      , sum( decode ( name, 'user rollbacks', value, 'user commits', value, 0 ) ) trans
from stats$sysstat s, stats$snapshot sn
where name in ( 'user rollbacks', 'user commits' ) and s.snap_id = sn.snap_id
and s.instance_number = 2
group by s.snap_id, sn.snap_time, s.dbid, s.instance_number) a,
(select s.snap_id-1 snap_id, sn.snap_time, s.dbid, s.instance_number, sum (value) trans
from stats$sysstat s, stats$snapshot sn
where name in ( 'user rollbacks', 'user commits' ) and s.snap_id = sn.snap_id
and s.instance_number = 2
group by s.snap_id, sn.snap_time, s.dbid, s.instance_number) b,
stats$snapshot c
where a.snap_id = b.snap_id and a.instance_number = b.instance_number and a.snap_id = c.snap_id - 1
order by a.instance_number, to_char(c.snap_time, 'mmdd hh24mi')
    
```

해 @Risk의 'Define Distribution' 기능을 활용하였는데, 이 기능은 해당 데이터를 입력으로 받아들이며 데이터를 가장 잘 묘사하는 분포를 지정하여 준다. 여기서, 데이터의 크기는 2500으로 하였다.

기본 데이터인 동시 사용자 수와 실행된 트랜잭션 수는 추출된 데이터를 그대로 이용한다. 하지만 밀집 시간 여유율은 수집된 정보를 변형하여야 한다. 즉, 정보시스템의 CPU 활용률 정보를 수집하여 오름차순으로 정렬한 후 90퍼센타일(90<sup>th</sup> percentile)에 해당하는 밀집 시간 정보들만을 골라 이들을 보통 수행 시간의 평균 활용률 값으로 나누어 준다. 그러면 보통 시간 대비 밀집 시간의 밀집 비율 값들을 얻을 수 있는데, 이 값들에 대응하는 확률분포를 정의하도록 한다. 이제 확률분포의 정의가 끝나면 지정된 분포를 이용하여 실제 시뮬레이션 작업을 수행한다. 시뮬레이션 수행 시 입력자료(input)는 앞서 지정된 확률분포가 되며, 출력자료(output)로는 tpmC성능 값을 지정한다.

시뮬레이션 실행을 통해 얻은 출력자료(tpmC 값) 분포에서 평균 tpmC를 기준으로 상하위 각 1개의 표준편차에 위치한 tpmC를 각각 최소 및 최대 tpmC로 지정하여 범위( $\mu - \sigma \sim \mu + \sigma$ )를 설정한다. 이렇게 성능 기준의 범위가 마련되면 자동적으로 이러한 범위를 만족하는 정보시스템 대안들을 찾을 수 있는데, 해당 성능 범위를 만족하는 공인 시스템들을 TPC-C 결과표에서 찾아서 이들을 사용자가 선택할 수 있는 대안 시스템으로 지정한다. 이러한 과정에 따라, 결과적으로 수용 가능한 tpmC의 범위와 이를 만족하는 공인 시스템 집합을 얻게 된다.

또한 앞서 비즈니스 제약조건에서 언급한 것처럼 향후 시스템 확장에 유연하게 대응하기 위하여 해당 성능 값보다 고성능의 시스템을 CPU를 줄여서 구매하는 경우를 지원하기 위하여 최대 tpmC보다 1.5배 높은 성능을 제공하는 시스템까지를 대

안 시스템에 포함시키도록 한다.

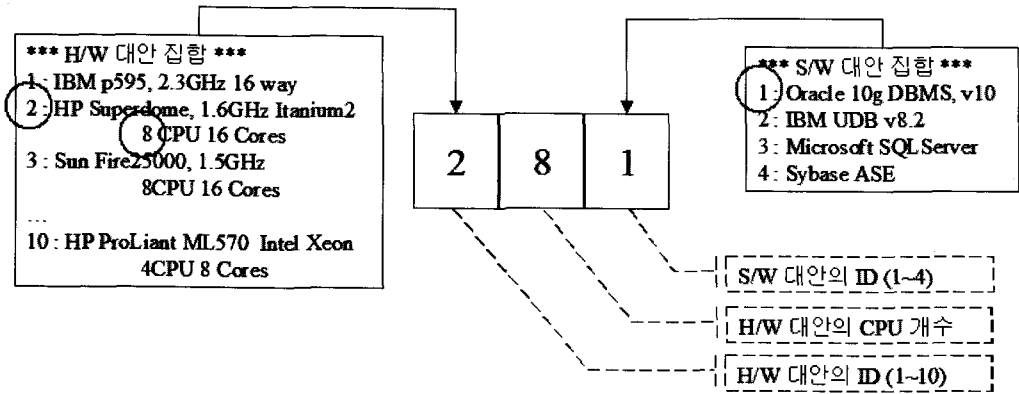
### 3.3.3 단계 2 : 유전 알고리즘을 이용한 최적 대안의 선정

앞서 "3.1 문제의 정의"에서 TCO를 산정하는 식이 3개 변수의 곱을 포함한 비선형 식으로 표현되어 LP 문제로 해결할 수 없음을 설명하였다. 따라서 목적함수를 만족시키는 의사결정변수의 값을 최적화 방식으로 구하지 못하고 탐색적 방식으로 구하게 된다. 즉, 의사결정변수가 가질 수 있는 다양한 값의 조합을 탐색하면서 목적함수를 가장 만족시키는 만족해를 구해야만 한다. 이를 위해 본 연구에서는 유전 알고리즘을 이용한 만족해의 탐색을 시도하였다. 따라서 단계 2에서는 유전 알고리즘을 이용하여 단계 1에서 선택한 대안 시스템들 중에서 목적함수를 최대로 만족시키는 정보시스템은 무엇인지 찾아내도록 한다. 본 연구에서는 유전 알고리즘의 수행을 위해 Evolver를 사용하였다.

유전 알고리즘의 적용을 위한 의사결정변수는 <표 1>에 나타난 바와 같이 H/W 모델, 적정 CPU 개수, S/W 제품 등 3가지이다. 유전 알고리즘의 적용을 위해서는 이 3가지 의사결정변수의 조합(combination)을 이용하며, 각 변수가 가질 수 있는 다양한 값의 조합을 탐색하여 목적함수를 최대로 만족시키는 변수 값의 조합을 구하게 된다. 여기서 H/W 모델 변수가 가질 수 있는 값은 앞의 단계 1에서 선택된 대안 시스템 집합 전체가 범위가 된다. 다음으로, CPU 개수가 가질 수 있는 값은 1부터 H/W 모델이 가질 수 있는 최대 CPU 개수까지이다. S/W 제품은 선택된 H/W 모델과 함께 사용될 수 있는 데이터베이스 S/W 전체가 된다. 각각의 의사결정변수가 가질 수 있는 값의 범위 내에서 임의로 값을 변화시키면서 목적함수를 만족시키는 의사결정변수 값의 조합을 찾는 방법이 Evolver의 'recipe' 모드이다. 이 기능을 이용하여 <그림 3>과 같이 H/W 모델 대안 시스템의 수, 각 H/W 모델별 최대 CPU 개수, 그리고 가능한 S/W 제품 대안의 수 범위 내에서 유전 알고리즘에 의해 의

1) 이는 박성민[1]이 제시한 확률분포의 모수 추정을 위한 적정 표본크기 범위인 최소 200, 최대 400을 만족한다.





〈그림 3〉 의사결정변수의 조합

사결정변수의 값(정수)을 바꾸어 가면서 목적함수를 최대로 만족시키는 조합 대안을 찾아내게 된다.

한편, 의사결정의 목표는 대안으로 판별된 시스템 조합 중에서 적은 TCO를 가지면서 동시에 좋은 성능을 가지는 최적 대안을 찾는 것이다. 이를 나타내는 목적함수는 <표 5>와 같다.

〈표 5〉 의사결정의 목적함수

$$f = \max(a * TCO + b * Perf)$$

where  $a = -(\min \text{ tpmC} / \text{Budget})$ ,  $b = 1$

<표 5>에서 TCO는 적을 수로 좋으므로 계수 a

는 음수로 표시하였으며, TCO와 성능의 가중치를 균등하게 가져가기 위해 계수 값을 (min tpmC/Budget)으로 설정하였다. 반면, 성능치는 높을수록 좋으므로 성능 계수  $b = 1$ 로 지정하였다. 여기서, 만약 의사결정 목표가 바뀌어 가격만 고려하는 경우에는  $a = -1$ ,  $b = 0.001$ , 성능만 고려하는 경우에는  $a = -0.001$ ,  $b = 1$ 로 설정하도록 한다.

다음으로, TCO를 결정하는 식은 <표 6>과 같다.

그리고 성능 Perf를 결정하는 식은 <표 7>과 같다.

제약조건들은 다음과 같이 설정하였다. 첫째, 목적함수 내의 변수 Perf는 단계 1에서 파악된 최소/최대 tpmC 범위 안에 들어야 하므로 min tpmC

〈표 6〉 TCO 결정 식

If (CPU 축소 가능 = 'No') H/W 실제 납품가 = H/W 규정 가격 \* (1 - 할인율)  
 Else H/W 실제 납품가 = (H/W 규정 가격 - (축소된 CPU 개수 \* CPU 가격)) \* (1 - 할인율)

If (Dual Core 사용 = 'No')  
 S/W 실제 납품가 = 사용된 CPU 개수 \* 단위 CPU당 가격 \* (1 - 할인율)  
 Else S/W 실제 납품가 = 사용된 CPU 개수 \* 단위 CPU당 가격 \* CPU당 Core 개수 \* 회사별 Core Factor \* (1 - 할인율)

TCO = H/W 실제 납품가 + S/W 실제 납품가

〈표 7〉 성능 결정 식

$$Perf = \text{공인 성능 tpmC} * (\text{실제 사용한 CPU 개수} / \text{공인 기록 상의 CPU 개수})$$

$\leq \text{Perf} \leq \text{max tpmC}$ 로 설정하였다. 둘째, 목적 함수 내의 TCO는 배정된 예산 보다 작아야 하므로  $\text{TCO} \leq \text{Budget}$  조건을 설정하였다. 셋째, 확장성을 고려하여 고성능의 시스템을 CPU 개수를 줄여 구매하는 경우에도 CPU 개수는 원래 서버 규격 대비 50%까지만 줄일 수 있다. 이를 반영하기 위해(공인 기록 CPU 개수 - 실제 사용 CPU 개수)  $\geq 0$  조건과 (공인 기록 CPU 개수/실제 사용 CPU 개수)  $\leq 2$  조건을 설정하였다.

본 연구에서 효율적인 의사결정을 위해 기본적으로 가정하고 있는 사항들은 다음과 같다. 첫째로, H/W와 S/W의 선택은 독립적이라는 점이다. 이는 각각의 가격을 각 벤더사가 서로 다르게 독립적으로 제시하는 상황을 수용하기 위해 반드시 필요한 가정이다. 둘째, CPU의 감소는 그 감소비율만큼 tpmC 성능을 감소시킨다는 가정이다. 이는 높은 성능의 시스템에서 CPU를 줄여서 사용하는 것을 선택 대안에 포함시키기 위하여 필요한 가정이다.

### 4. 사례 적용

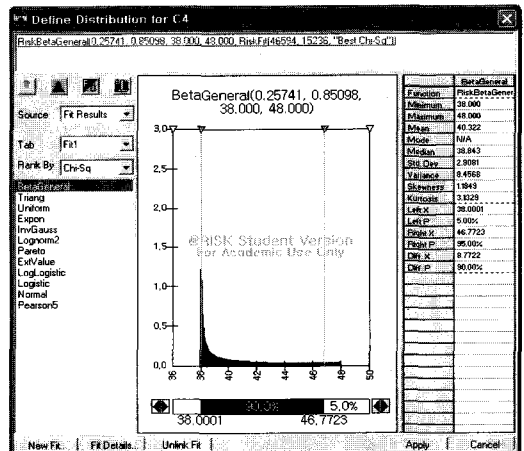
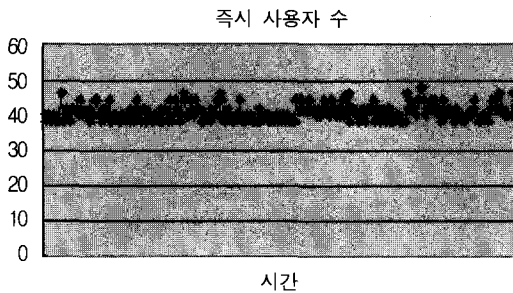
앞서 제시한 연구 모델을 실제 시스템 용량 산정 및 선택에 적용한 사례를 살펴 보기로 한다. 한 제조회사의 MES(Manufacturing Execution System)에 대하여 현재 운영 현황을 기반으로 차후 증설될 시스템의 용량 산정 및 적정 시스템 선정 작업을 수행한 사례이다.

#### 4.1 업무 적용 사례

선택된 대안 시스템들에 대해서 규정된 가격 이외에 더 이상의 가격 변화가 없다는 것을 가정하고 적정 시스템을 선정하는 방법은 다음과 같다. 먼저 SQL을 통해 기존 시스템의 활용 데이터를 수집한 후 이를 묘사할 수 있는 확률분포를 선정하였다. 사용한 확률변수와 데이터로부터 추정된 확률분포를 요약하면 <표 8>과 같다.

<표 8> 확률변수의 분포 추정

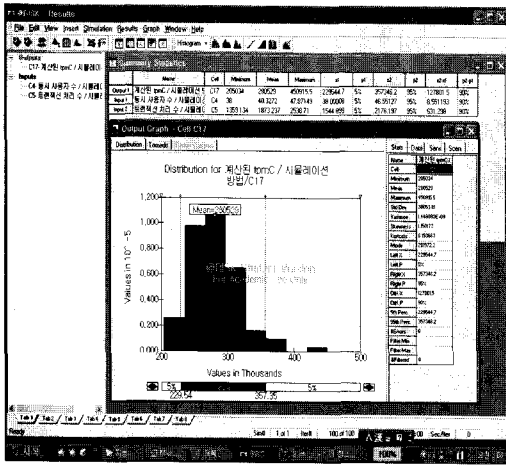
확률변수	확률분포의 추정
동시 접속 사용자 수	= RiskBetaGeneral (0.25741, 0.85098, 38, 48, RiskFit (56899, 35037, "Best Chi-Sq"))
사용자별 트랜잭션 처리 수	= RiskLogistic (1871.3, 106.28, RiskFit (56899, 12340, "Best Chi-Sq"))
밀집 시간 여유율	= RiskGamma (1.2654, 0.29097, RiskShift (1.50537), RiskFit (74251, 70944, "Best Chi-Sq"))



<그림 4> 동시 접속 사용자 현황 데이터 및 확률분포의 추정

예를 들어, 동시 접속 사용자의 현황 데이터와 이 데이터를 가장 잘 묘사하는 확률분포를 @Risk의 'Define Distribution' 기능을 이용하여 추정한 결과는 <그림 4>와 같다.

다음으로 <표 8>의 확률분포를 <표 2>의 기존 정보 시스템 용량 산정 방법에 대입하여 실제 시뮬레이션을 수행하도록 한다. <그림 5>는 시뮬레이션 작업 수행결과(tpmC)의 분포를 보여 주고 있다.

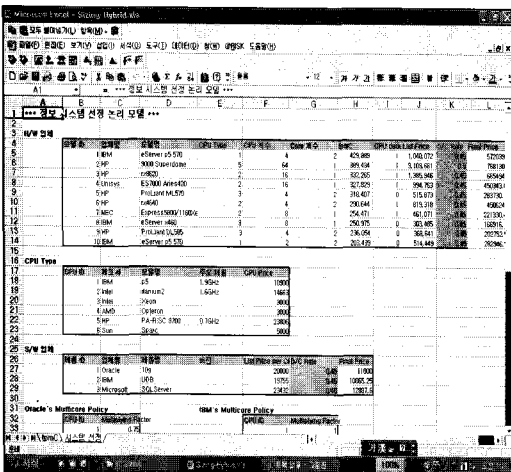


<그림 5> 시뮬레이션 수행 결과의 분포

280,529 tpmC, 표준편차는 38,053 tpmC로 나타나고 있다. 따라서 적정 tpmC의 범위는 최소 241,908 tpmC에서 최대 318,012 tpmC로 설정하기로 한다. 이러한 성능 조건을 만족하는 대안들을 TPC-C 성능 리포트에서 검색한 결과는 <그림 6>과 같다.

이제 적정 성능 범위 및 관별된 대안 집합을 활용하여 최적 시스템을 선정하는 작업을 수행한다. 의사결정의 기준이 되는 시스템 구입 예산은 500,000(천원)으로 설정되었다. 따라서 <표 5>에 나타난 바와 같이 목적함수의 TCO 계수는  $-(\min \text{tpmC} / \text{Budget}) = -0.484$ 가 된다. 그런 후, <그림 3>과 같이 H/W 모델, CPU 개수, S/W 제품의 조합을 의사결정변수로 하여 Evolver 상에서 유전 알고리즘을 통해 최적 대안을 모색한 결과, 8번 모델의 H/W에 8개의 CPU를 사용하고 2번 제품의 S/W를 사용하는 것이 가장 적절한 시스템으로 제시되었다. 이 때, TCO는 253,838(천원)이고 성능 값은 250,975 tpmC이다.

다음으로 기존의 방법처럼 성능만을 고려한 의사결정을 위하여 목적함수의 TCO 계수를  $-0.001$ 로 최소화하여(즉, 가격은 고려하지 않고 성능만을 목적함수의 주요 요소로 하여) 다시 최적 대안을 구해 보았다. 그 결과, 이번에는 6번 모델의 H/W에 4개의 CPU를 사용하고 2번 제품의 S/W를 사용하는 것이 최적 대안으로 제시되었다. 이 경우, TCO는 494,085(천원)이고 성능 값은 290,644 tpmC로 나타나, 성능과 가격을 모두 고려하여 선택한 최적 대안보다 가격 대비 성능이 열등한 것으로 나타났다.



<그림 6> 성능 값 범위를 만족하는 대안 집합 및 가격 정책

#### 4.2 본 연구 방법의 성과 평가

<표 9>는 기존의 용량 산정 방법을 적용했을 경우, 본 연구에서 제시한 방법에 따라 성과와 가격을 동시에 감안한 경우, 그리고 성능만을 고려했을 경우의 성과를 성능 요구치(tpmC), 결정된 성능 값(tpmC), 도입가격 차원에서 비교한 결과이다.

<표 9>를 보면 기존 방법의 결과와 본 연구에

<그림 5>에서 보듯이 시뮬레이션 결과, 평균은

〈표 9〉 본 연구 방법의 성과 평가

적용 방법	성능 요구치 (tpmC)	결정된 성능 값 (tpmC)	도입가격 (천원)	기존 방법과의 비교
기존 방법	285,469	290,644	494,085	-
본 연구 방법 (성능과 가격을 동시 고려)	241,906~318,012	250,975	253,838	성능 요구 만족 비용 효과 극대화
본 연구 방법 (성능만 고려)	241,906~318,012	290,644	494,085	기존 방법과 유사한 결과

서 성능만을 고려한 경우의 결과가 매우 유사함을 알 수 있다. 그러나 이 결과와 본 연구에서 제안한 성능과 가격을 동시에 고려한 경우의 결과를 비교하면, 본 연구 방법을 적용하여 결정된 성능 값 250,975tpmC는 성능 요구치 범위를 만족하는 동시에 기존 방법의 성능 값 290,644tpmC와 비교적 큰 차이를 보이지 않으면서도 실제 도입가격은 253,838(천원)으로 기존 방법의 494,085(천원)의 거의 절반 수준에 해당할 만큼 비용 대비 효과 측면에서 상당한 우위를 보임을 알 수 있다.

따라서, 본 연구에서 제안한 방법은 성능 요구치와 TCO와의 상충 관계를 감안하여 시스템을 선정하도록 함으로써 성능 요구치는 크게 차이가 나지 않으면서 도입가격은 상당히 절약할 수 있는 시스템을 선정할 수 있는 방법임을 알 수 있다.

한편, 본 연구에서는 정보시스템 선정을 위하여 시뮬레이션과 유전 알고리즘을 결합하는 접근 방식을 취하였다. 이 중 시뮬레이션 기법은 기존의 용량 산정 방법에서 뚜렷한 근거 없이 경험적인 상수로 제시되어온 주요 기준치(예를 들어, 동시 접속자 수, 접속자가 실행한 트랜잭션의 수, 밀집 시간 여유율 등)의 값을 실제 데이터에 기반한 확률분포로 추정함으로써 요구 성능치의 가능 범위와 이 범위를 만족하는 정보시스템의 선택 대안을 구하는데 활용하였다. 그러나 만일 다른 방법을 통하여 주요 기준치의 값을 정확히 산정할 수 있거나 또는 정보시스템의 선택 대안이 사전에 다른 이유로 이미 결정되어 있는 경우라면 시뮬레이션 단계를 굳이 거치지 않아도 될 것이다. 즉, 시뮬레이션 단계는 주요 기준치 및 대안 정보시스템이 신뢰할 수

있는 방법으로 사전에 결정된 경우에는 생략될 수 있을 것이다. 그러나 이러한 경우라 하더라도 가격 결정의 비선형적인 특성은 그대로 상존하므로 유전 알고리즘을 통하여 목적함수를 최대로 만족시키는 대안 조합의 탐색 작업은 반드시 수행되어야 할 것이다.

## 5. 결 론

본 연구에서는 기업의 투자활동 중에서 많은 비용이 들어가는 정보시스템의 적정 용량 산정 및 선택의 문제를 보다 과학적으로 해결하기 위한 방안을 제시하였다. 본 연구에서 제안하는 시뮬레이션과 유전 알고리즘의 하이브리드 기법은 기존에 업계에서 관행적으로 이루어지던 경험적 시스템 선정 방안에 이론적인 근거를 제공할 뿐만 아니라 실제로 활용됨으로써 기업의 비용 절감에 도움이 될 수 있도록 설계되었다.

본 연구의 기여는 첫째, 보정 상수를 확률변수로 대체함으로써 선택의 범위를 수용하기 위한 이론적인 토대를 제공하였다는 점이며, 둘째, 비선형적으로 변하는 가격체계를 반영함으로써 가격정책과 할인율의 변화를 반영한 비용 효과적인 시스템을 선택할 수 있도록 지원할 뿐만 아니라 이를 활용하여 고객들이 역으로 할인율을 제시함으로써 가격 협상에 활용할 수 있도록 도움을 주게 되었다는 점이다.

본 연구에서 제안한 시스템 선정 모델은 비즈니스 상황 및 고려 대안에 따라 확률분포와 가격조건을 바꾸어가면서 실험할 수 있는 융통성을 갖고 있

다. 예를 들면, 실제 비즈니스 상황의 변화를 반영하여 밀집 시간 분포 등에 변화를 줄 수 있으며, 또한 가격조건을 변경하여 벤더사의 제안 내역 변화 및 가격정책 변화를 수시로 반영하여 검증하는 것을 가능하게 한다.

향후 추가적인 연구를 통해 보다 많은 보정 상수들을 확률변수로 대체하는 것이 요구된다. 이를 위해서는 확률분포 추정치 타당한 근거를 마련할 수 있도록 실제 운용 데이터를 더 많이 확보할 수 있도록 해야 할 것이다. 또한 CPU 용량 산정뿐만 아니라 메모리 및 디스크의 크기 산정을 추가로 고려하여 다양한 측면의 상충관계와 보합관계를 감안한 최적의 시스템 선정 노력이 계속적으로 필요할 것이다.

## 참 고 문 헌

- [1] 박성민, "시뮬레이션 입력 모형화 - 확률분포 모수 추정을 위한 표본 크기 설정", 「한국경영과학회지」, 제31권, 제1호(2006), pp.15-24.
- [2] 이용균, 김경섭, "유전 알고리즘과 시뮬레이션을 이용한 유연생산시스템에서의 최적 버퍼 할당", 「한국시뮬레이션학회지」, 제10권, 제4호(2001), pp.65-75.
- [3] 정해용, 나종희, 최광돈, "공공 부문 정보 시스템의 하드웨어 용량 산정 방식 설정을 위한 실증적 연구", 「정보화 정책」, 제12권, 제3호(2005), pp.48-66.
- [4] 한국전산원, "정보 시스템 용량 산정 기술 및 프레임워크 연구", NCA IV-RER-03060, 2003.
- [5] 한국전산원, "H/W 용량 산정에 관한 연구", 2002.
- [6] Fujimoto, H., C. Lian-Yi, Y. Tanigawa, and K. Iwahashi, "FMS scheduling by hybrid approaches using genetic algorithm and simulation," *Genetic Algorithms in Engineering Systems : Innovations and Applications*, Vol.12, No.14(1995), pp.442-447.
- [7] IBM 사의 듀얼 코어 가격 정책, <http://www-03.ibm.com/software/sla/sladb.nsf>.
- [8] Jeong, S.J., S.J. Lim, and K.S. Kim, "Hybrid approach to production scheduling using genetic algorithm and simulation," *IJAMT*, Vol.28(2006), pp.129-136.
- [9] Microsoft 사의 듀얼 코어 가격 정책, <http://www.microsoft.com/sql/howtobuy/process-or.mspx>
- [10] Oracle 사의 듀얼 코어 가격 정책, <http://www.oracle.com/corporate/pricing/pricelists.html>.
- [11] Suiadee, W. and T. Tingsanchali, "A combined simulation-genetic algorithm optimization model for optimal rule curves of a reservoir : a case study of the Nam Oon Irrigation Project," *Hydrological Processes*, Vol.21(2007), published online.
- [12] TPC-C 벤치마크 보고서, Transaction Processing Performance Council, [http://tpc.org/tpcc/spec/tpcc\\_current.pd](http://tpc.org/tpcc/spec/tpcc_current.pd).