

# Heuristic Search Method for Cost-optimized Computer Remanufacturing

Hong-Bae Jun<sup>†</sup> · Gapsu Sohn

Department of Industrial Engineering, Hongik University

## 복수의 중고 컴퓨터 재조립 비용 최소화를 위한 휴리스틱 탐색 알고리즘

전홍배<sup>†</sup> · 손갑수

홍익대학교 산업공학과

Recently, the optimisation of end-of-life (EOL) product remanufacturing processes has been highlighted. In particular, computer remanufacturing becomes important as the amount of disposed of computers is rapidly increasing. At the computer remanufacturing, depending on the selections of used computer parts, the value of remanufactured computers will be different. Hence, it is important to select appropriate computer parts at the reassembly. To this end, this study deals with a decision making problem to select the best combination of computer parts for minimising the total remanufacturing computer cost. This problem is formulated with an integer nonlinear programming model and heuristic search algorithms are proposed to resolve it.

**Keywords** : Remanufacturing, Reuse, Computer, Heuristic Search Algorithm

### 1. 서 론

최근 많은 전자제품들이 사용에 큰 불편이 없음에도 불구하고 폐기되는 경우가 많아지고 있다. 전자제품 폐기물이 많아짐에 따라 환경에 악영향을 미칠 뿐만 아니라, 가용할 수 있는 자원의 양이 점점 줄어들고 있다. 이런 환경 문제 및 자원고갈에 대비하기 위해서 많은 나라가 폐기된 제품의 재사용 및 재활용을 장려하고 제도화하고 있다. 예

를 들어 유럽에서는 ELV(End-of-Life Vehicles)와 WEEE(Waste Electrical and Electronic Equipment)와 같은 환경 법규들을 제정하여 시행하고 있으며, 이 때문에 제품의 재사용 및 재활용에 대한 관심이 더욱더 증가하고 있다.

제품 중 컴퓨터는 특히 사람들이 일상생활에서 많이 사용하는 제품으로써 IT 기술 발전에 힘입어 제품 사이클이 점점 짧아지고 있다. 즉, 충분히 사용 가능 한데도 소비자의 취향 및 유행의 변화 때문에 사용되지 않고 버려지는 컴퓨터들이 많아지고 있다. 이러한 컴퓨터들을 수거해서 잘 활용만 한다면 추가적인 자원의 낭비와 환경에 피해를 주지 않고도 충분히 재사용 가능한 제품을 만들 수 있으며, 이를 재판매함으로써 적지 않은 경제적 이득을 거둘 수 있다.

하지만 지금까지 컴퓨터와 전자제품 재제조에 관한 연구들 대부분은 각 부품에 들어있는 유해 물질을 제거하고 원자재의 재활용을 하는 방식에 관한 연구나 제품

Received 9 August 2012; Finally Revised 15 October 2012;  
Accepted 22 November 2012

<sup>†</sup> Corresponding Author : hongbae.jun@hongik.ac.kr

© 2012 Society of Korea Industrial and Systems Engineering

This is Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.  
(<http://creativecommons.org/licenses/by-nc/3.0>).

의 수집에 관한 역물류(reverse logistics) 및 재제조되는 부품의 성능 평가에 관한 연구들이 대부분이었다. 사용 후 버려진 컴퓨터들의 중고부품들을 고객의 요구사항에 맞으면서도 경제적으로 효율성 있게 조합하여 컴퓨터를 재조립하는 방법에 대한 연구는 아직 많이 미흡하다. 컴퓨터는 사용되는 부품이 제한적이고 다른 제품들에 비해서 부품간의 호환 및 조합이 쉽기 때문에 부품의 재사용이 훨씬 원활하다. 또한, 컴퓨터 부품은 원재료 단위에서 재활용되기 위해서는 재처리 과정을 거쳐야 하고, 재처리 시 유해물질의 발생도 일어나게 된다. 하지만 부품 단계에서의 재사용은 재처리 비용을 절감하고 환경적으로도 좋은 결과를 가져올 수 있다. 위와 같은 이유로 본 연구에서는 원자재 레벨의 재활용이 아닌 컴퓨터 부품을 재사용하는데 초점을 맞추고자 한다.

중고 컴퓨터를 재조립할 때 기업의 수익을 극대화하기 위해서는 적절한 중고부품을 활용해 고객이 원하는 요구사항에 맞으면서도 동시에 컴퓨터 재조립 비용은 최소화할 수 있는 방안이 필요하다. 또한, 중고컴퓨터 조립시 고객이 원하는 보증기간까지 제품이 고장이 나지 않게 부품을 선별해 조립하는 것도 중요하다. 많지 않은 컴퓨터들을 재조립할 경우는 크게 문제가 안 되겠지만, 한번에 다량의 컴퓨터를 재조립해 납품해야 하는 대량 수주 납품 계약에는, 어떻게 중고부품을 선별해 컴퓨터를 재조립했느냐에 따라 비용차이가 크게 날 수가 있다.

이러한 문제점을 인식하고 본 연구에서는 버려진 컴퓨터들의 중고부품들을 활용하여 고객의 요구사항에 맞추면서 판매 수익을 극대화할 수 있는 컴퓨터 부품 선별방안에 대한 연구를 수행하고자 한다. 본 연구의 구성은 다음과 같다. 제 2장에서는 본 연구에서 다루고 있는 컴퓨터 재조립/재제조에 관한 기존 연구들을 살펴본다. 제 3장에서는 본 연구에서 다루고 있는 문제를 수학적으로 정의한다. 제 4장에서는 본 연구문제를 효율적으로 해결할 휴리스틱 해법을 설명하고, 제 5장에서는 이를 검증하기 위한 컴퓨터 모의실험 및 그 결과를 소개하며, 끝으로 제 6장에서 전체적인 결론을 맺고자 한다.

## 2. 기존연구

재제조는 일반적으로 “사용 후의 제품을 신제품과 동등한 수준으로 제조하는 자원 순환의 이상적 방법”으로 정의된다[9]. 재제조의 장점으로는 첫째 중고 부품으로 제품을 만들기 때문에 제품가격이 원제품 가격보다 훨씬 저렴하다. 또한, 제품의 폐기량이 줄어들기 때문에 폐기 비용을 절감할 수 있고, 재제조 제품의 판매로 추가적인 수익을 올릴 수 있다[1]. 따라서 재제조를 통하여 적은

비용으로 새 제품과 품질이나 기능 면에서 유사한 제품을 생산할 수 있다면 에너지 및 자원의 절감과 같은 효과를 기대할 수 있다[2]. 둘째로 재제조를 하면 오히려 품질이 신제품보다 좋아질 수도 있다. 이미 실사용으로 안정화 단계까지 이른 제품들이기 때문에 그 제품을 재제조함으로써 훨씬 낮은 고장률을 보일 수 있다. 또한, 안정화된 제품에 재제조를 통해 최신의 기술이 접목되면 품질에서도 훨씬 좋은 제품이 만들어질 수도 있다[9]. 셋째로 재제조 산업은 많은 고용 효과를 가져온다. 재제조 산업의 고용 효과는 소비재 산업과 거의 비슷한 수준으로 절감, 제약 등 단일 제조 업종과 비교시 거의 2배 수준으로 알려졌다[9].

전자제품들의 재제조에 대한 연구들은 전자제품과 관련한 다양한 환경규제들에 대응하는 차원에서 최근 활발히 이루어지고 있다. 중고 컴퓨터의 재제조 및 재활용에 대한 움직임은 컴퓨터 제조업체를 중심으로 먼저 생겨났다. 예를 들어 IBM에서는 독자적으로 Computer Recycling System을 운영하여 제품의 재활용 비중을 높이고 있다. Computer Recycling System에서는 부품을 종류별로 수집하고 재활용할 수 있는 부품과 불가능한 부품으로 나누는 작업이 컴퓨터에 의해 관리가 되고 있으며, 재활용이 가능한 부품들도 재료의 종류와 성분 함량에 따라 철저히 분류되어 관리되고 있다. 특히, IBM은 1992년에 폐컴퓨터 회수 프로그램을 도입해서 유럽 8개국에서 12,000톤의 폐컴퓨터를 회수해서 그 중 82%를 재활용하였다. 국내의 컴퓨터 재활용 업체들의 현황을 살펴보면 폐컴퓨터를 분해한 후 재사용 가능한 부품을 모아 재조립해서 중고품으로 판매하고 있다. 재조립된 컴퓨터들은 국내 공공기관 및 동남아시아, 인도, 파키스탄, 스리랑카 등지로 수출되고 있다.

그동안 컴퓨터 재제조에 대한 연구는 중고 컴퓨터 부품 중에서 가치가 있는 원재료들을 획득하는 방법이나 폐컴퓨터에 대한 회수 흐름 등에 중점을 두었으며, 컴퓨터 재제조에 대한 연구들은 상대적으로 관심이 적었다. 컴퓨터 재제조에 대한 관련 연구들로는 예를 들어, Melissen and De Ron[8]은 컴퓨터 부품 각각의 재처리 의사결정(Reuse, Shredding, Disposal)에 대한 비용을 분석해서 어떤 의사결정이 가장 많은 순이익을 내는지를 알아보는 모델에 대한 연구를 수행하였다. 연구자들은 컴퓨터 부품 종류를 8가지(motherboard, HDD, controller, video card, CD driver, floppy disc driver, processor, cables)로 분류해서 재처리 의사 결정 시 각 부품의 순이익을 구하였다. 총이익은 재사용되는 부품들의 이익과 Shredding 시 생성되는 원재료들의 이익, shredding시 남은 부품들의 이익의 합을 총이익으로 하고, 분해비용, Shredding 비용 및 인력비용 등을 총비용으로 정해서 순이익을 정하였다. 이

<Table 1> Comparison of Previous Studies

Authors	Product	Objective	Decision variable	Constraint	Approach
Lee et al.[4]	Computer	EOL option selection optimization	Decision on disposal computers	Environmental(Hazardous materials) and cost	Guideline proposal
Melissen and de Ron[5]	Computer	Maximization of profit	EOL options on parts	Administration cost, building cost, energy cost, financial cost	A decision making model
Bufardi et al.[7]	Telephone	Optimization of EOL scenarios	EOL scenario	Economic, Social and Environmental constraints	MCDA(multiple criteria decision aid)
Bufardi et al.[8]	Vacuum cleaner	Optimization of EOL scenarios	EOL scenario	Economic and Environmental constraints	MCDA(multiple criteria decision aid)
Lee et al.[9]	Refrigerator and television	Maximization of reuse rate	Reuse rates of electronic products	WEEE(Waste Electrical and Electronic Equipment)	Expert survey

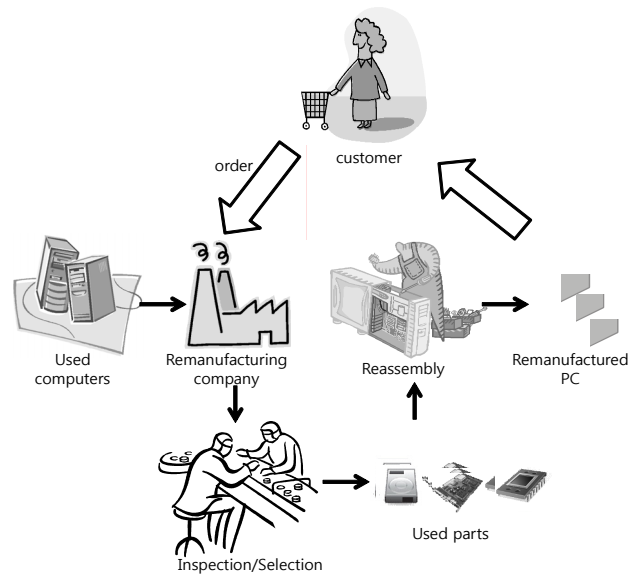
때 제한조건으로 한 부품을 얻기 위해서는 그전에 분해해야 하는 부품들이 있기 때문에 분해 시에 기술적인 제한을 두었다. 이런 기술적인 제약에 맞춰서 각 의사결정 시에 각 부품의 순이익을 정하는 decision 모델을 제시하였다. 최병철 외 3인[6]은 컴퓨터의 수명주기를 분석하고, 재활용 비율을 연구하였다. 연구결과 컴퓨터 폐기물에서 특히 인간에 해로운 성분이 많이 배출된다고 한다. 연구자들이 발표한 바로는 현재 우리나라에서 폐기되는 컴퓨터 중에 46% 정도의 컴퓨터가 재활용된다고 한다. 하지만 이 정도의 재제조율이라면 환경에 해로운 상황이며, 최소 이 재활용 비율이 63% 이상이 되어야만 환경적인 유해성이 사라진다고 주장하였다. 이러한 점들을 강조하면서 컴퓨터 재활용 부품의 비율을 끌어올리기 위해서는 재활용 기술의 향상 및 적절한 의사결정, 정부의 정책이 필요하다고 주장했다. <Table 1>은 컴퓨터를 포함한 전자제품에 재제조/재조립과 관련한 의사결정 문제에 대한 기존 연구들을 정리한 것이다.

이처럼 컴퓨터 재제조 시장에 대한 수요는 각종 규제 및 환경에 대한 사회적 인식, 원자재의 감소에 따른 부품 가격의 상승에 따라 더욱더 커질 것이다. 따라서 재제조 시 중고 부품들을 고객의 요구사항에 맞게 조합하는 방법에 대한 연구도 필요할 것으로 생각한다. 지금까지는 중고컴퓨터 부품을 원재료로 분해 및 환원하는 방법, 재제조 시의 부품의 분해순서, 재제조 시 발생하는 환경적인 악영향을 최소화하는 방법 등이 주로 연구되었다. 하지만 아직은 컴퓨터 재제조 측면에서 중고 부품들을 선별하여 사용 가능한 복수의 컴퓨터들로 조립하는 방법에 관한 연구가 부족했다. 컴퓨터는 부품의 수도 한정적이고 부품들의 수명도 매우 길뿐만 아니라 일정 조건만 만족하면 부품 간의 호환성도 높다. 또한, 컴퓨터가 폐기되는 이유는 부품의 고장보다는 기술의 급격한 발전 때문인 소비자의 변심이 주를 이룬다. 그래서 컴퓨터는 제품의 재사용 측면에서 유리한 대상이다.

### 3. 문제 정의

#### 3.1 개요

재제조 컴퓨터는 주문생산 방식과 일정한 사양을 미리 조합해서 판매하는 방식으로 거래가 이루어진다. 이 중 주문생산 방식에서는 고객이 주문하게 되면 주문을 받은 재제조 회사는 고객이 요구하는 조건(성능 사양 및 신뢰성)에 맞추어 컴퓨터를 조립해 고객에게 판매하게 된다. 이때 회사에서는 고객의 요구 조건을 맞추면서 가능한 한 저렴한 비용으로 컴퓨터를 재조립하기를 원하는데, 각 부품을 어떻게 조합해야만 비용이 최소가 되는지에 대한 최적 의사결정을 내리는 것이 필요하다. 본 연구에서는 이와 관련한 문제를 정의하고 최적 의사결정을 구할 수 있는 해법을 찾고자 한다.



<Figure 1> Sale Processes of Remanufactured Computer

본 연구의 문제를 정의하기 전에 다음과 같은 notation 을 먼저 정의한다.

### 3.2 Notation 정의

#### 3.2.1 파라메타

- $i$  : 부품의 종류,  $i = 1, 2, 3, 4, 5$  (1 : CPU, 2 : 메인보드, 3 : HDD, 4 : 파워공급장치, 5: RAM)
- $j$  : 각 부품 종류별 부품의 정렬순서,  $j = 1, 2, 3, \dots, m$
- $k$  : 재제조 된 컴퓨터의 정렬순서,  $k = 1, 2, 3, \dots, N$
- $C$  :  $N$ 개 중고 컴퓨터 조립 총비용
- $N$  : 고객이 주문한 컴퓨터의 수
- $R$  : 고객이 원하는 컴퓨터의 신뢰도, 보장기간(예, 1년, 2년, 3년)동안 주문한 컴퓨터의 몇 %가 고장이 발생이 안되는가 하는 정도를 확률로 나타낸 값
- $m$  : 컴퓨터 부품의 창고 재고 개수(편의상 부품 타입별로 동일한 개수,  $m$ 개를 재고로 가지고 있다고 가정함)
- $t$  : 컴퓨터 보장기간(단위, 년)
- $c_{ij}$  :  $i$ (부품 종류)의  $j$ 번째 부품의 중고가격,  
 $i = 1, 2, 3, 4, 5, j = 1, 2, 3, \dots, m$
- $d_{ij}$  :  $i$ (부품 종류)의  $j$ 번째 부품의 제품 성능정보,  
 $i = 1, 2, 3, 4, 5, j = 1, 2, 3, \dots, m$
- $D_i$  :  $i$ 번째 부품에 대한 고객의 요구 성능,  
 $i = 1, 2, 3, 4, 5$
- $l_j$  :  $j$ 번째 메인보드의 가능한 RAM Slot 개수,  
 $j = 1, 2, 3, \dots, m$
- $s_{ij}$  :  $i$ (부품 종류)의  $j$ 번째 부품의 호환 정보,  
 $i = 1, 2, 5, j = 1, 2, 3, \dots, m$
- $\lambda_{ij}$  : 부품 타입의  $j$ 번째 부품의 고장율,  
 $i = 1, 2, 3, 4, 5, j = 1, 2, 3, \dots, m$

#### 3.2.2 결정변수

- $X_k$ : 재조립된 컴퓨터 set  $k$ 의 결정변수 행렬
- $x_{ij}^k$ : 재조립된 컴퓨터 set  $k$ 에 부품종류  $i$ 의  $j$ 번째 부품의 선택여부
- $x_{ij}^k = 0$ , 재조립된 컴퓨터 set  $k$ 에 부품종류  $i$ 의  $j$ 번째 부품이 선택 안 됨
- 1 재조립된 컴퓨터 set  $k$ 에 부품종류  $i$ 의  $j$ 번째 부품이 선택 됨

$$X_k = \begin{bmatrix} x_{11}^k & x_{12}^k & \dots & x_{1m}^k \\ x_{21}^k & x_{22}^k & \dots & x_{2m}^k \\ x_{31}^k & x_{32}^k & \dots & x_{3m}^k \\ x_{41}^k & x_{42}^k & \dots & x_{4m}^k \\ x_{51}^k & x_{52}^k & \dots & x_{5m}^k \end{bmatrix}, \quad k = 1, 2, 3, \dots, N$$

### 3.3 파라메타 설명

#### 3.3.1 $d_{ij}$ 의 설명

$d_{ij}$ 는 부품의 성능을 정의한다. 각 부품들(메인보드 제외)은 저마다 성능을 가지고 있다. CPU는 동작속도를, HDD는 디스크용량, 파워공급장치는 정격용량, RAM은 메모리 용량을 가지고 있다. 이것들은 쉽게 정의가 가능하기 때문에 각각의 수치를 제품정보로 사용한다. CPU는 GHz, HDD는 GB(Gigabyte), 파워공급장치는 W, RAM은 MB(Megabyte)를 기준으로 한다. 예를 들어 HDD가 500GB 이면  $d_{3j} = 500$ 이다.

#### 3.3.2 $s_{ij}$ 의 설명

$s_{ij}$ 는 부품의 호환정보이다. 메인보드는 RAM, CPU의 부품 특성과 맞아야 정상적으로 작동된다. HDD도 기술적인 특징(SATA, H-IDE)을 가지고 있지만, 적은 비용으로도 충분히 호환이 가능하기 때문에 본 연구에서는 그 호환정보는 무시한다. 또한, 파워공급장치는 특별한 기술적인 특징이 없고 모든 부품에 적용이 가능하므로, 본 연구에서는 무시한다. 그러므로 3가지의 부품(메인보드와 RAM, CPU)만이 부품의 호환정보를 아래와 같이 추가적으로 갖게 된다.

#### (1) CPU의 호환 정보 정의

CPU의 호환정보는 동작속도와 제품의 사항으로 나눌 수 있다. CPU에 기술적인 특징은 제조회사와 각각 가지고 있는 소켓으로 나타낸다. 각각의 정보에 대한 정의를 나타내면 <Table 2>와 같다.

<Table 2> CPU Compatibility Information Definition

CPU Brand	CPU Socket	$s_{1j}$
INTEL	1567	1
INTEL	1336	2
INTEL	1156	3
INTEL	1155	4
INTEL	988	5
INTEL	775	6
INTEL	771	7
INTEL	603	8
AMD	AM3	9
AMD	AM2+	10
AMD	AM2	11
AMD	940	12
AMD	939	13
AMD	754	14
AMD	F	15
AMD	A	16
AMD	G	17

## (2) RAM의 호환 정보 정의

RAM의 종류는 DDR1, DDR2, DDR3로 나뉜다. RAM의 호환 정보를 정의하기 위해서 DDR1이면  $s_{5j} = 100$ , DDR2이면  $s_{5j} = 600$ , DDR3이면  $s_{5j} = 900$ 로 둔다.

## (3) 메인보드의 호환 정보 정의

메인보드는 컴퓨터의 부품조합 시에 가장 중요한 부품이다. 메인보드에서 필요한 정보는 RAM 슬롯의 개수, CPU 제조회사 및 CPU 소켓, 가능한 RAM의 종류이다. 이를 고려하여 백의 자리는 RAM의 종류, 십 이하 자리는 CPU 회사 및 소켓을 지칭하는 것으로 한다.

3.3.3  $D_i$ 의 설명

$D_i$ 는 제품의 성능을 나타낸다. 예를 들어 고객의 요구사항이 CPU 2.3G, 용량이 160G, 파워공급장치는 450W, RAM은 1GB 이상인 컴퓨터를 원할 때,  $D_1 = 2.3$ ,  $D_3 = 160$ ,  $D_4 = 450$ ,  $D_5 = 1024$ 가 된다.

## 3.4 문제 가정 및 정의

문제를 정의하기 전에 고려하고 있는 가정사항들을 나열하면 다음과 같다.

## 1) 문제의 가정

- i) 고객이 N개의 컴퓨터를 주문했다.
- ii) 보유하고 있는 중고 컴퓨터 부품의 사용 시간을 알고 있다.
- iii) 각 부품 재고 수는 고객의 수요량보다는 많으며, 편의상 모든 부품 타입에 대해 그 재고 수량은 동일하다고 가정한다.
- iv) 중고컴퓨터 조립시 부품 재료비를 제외한 부품의 조립비용은 작다고 생각하여 무시한다.
- v) 조합되는 부품 종류의 숫자는 CPU, RAM, 메인보드, HDD, 파워공급장치들로 총 5가지이다.
- vi) 컴퓨터에 들어가는 부품의 개수는 메인보드, CPU, 파워공급장치는 한 개씩 들어가지만 RAM은 메인보드에서 가능한 슬롯 수만큼, HDD는 최대 4개까지 조합할 수 있다.
- vii) 기존 재고 부품들 중 조립시 마땅한 부품이 존재하지 않으면 새 부품을 추가하여 조립할 수 있다.
- viii) 선택된 HDD중에 신뢰도가 가장 높은 것을 메인 드라이브(운영체제가 깔리는 드라이브)로 한다.
- ix) 제품의 고장시간의 분포는 지수분포를 따른다고 가정한다.

## 2) 문제의 정의

본 연구에서 다루고자 하는 문제는, 복수개의 중고 컴

퓨터의 제조립시 컴퓨터 조립의 기술적인 제약조건과 고객의 요구사항에 관한 조건들(컴퓨터의 사양, 제품의 신뢰도, 필요 대수)을 만족하면서 최소의 비용으로 조립할 수 있는 최적의 부품 조립 안을 찾는 것이다. 이를 수식으로 나타내면 아래와 같다.

Minimize

$$C = \sum_{k=1}^N \sum_i \sum_j x_{ij}^k \cdot c_{ij} \quad (1)$$

Subject to

$$\sum_j x_{ij}^k \cdot d_{ij} \geq D_i \text{ for } \forall k, \forall i \quad (2)$$

$$\sum_j (x_{1j}^k \cdot e^{-\lambda_{1j}^k}) \cdot \sum_j (x_{2j}^k \cdot e^{-\lambda_{2j}^k}).$$

$$MAX_{j=1}^n (x_{3j}^k \cdot e^{-\lambda_{3j}^k}) \cdot \sum_j (x_{4j}^k \cdot e^{-\lambda_{4j}^k}).$$

$$\prod_j (x_{5j}^k \cdot e^{-\lambda_{5j}^k}) \geq R \text{ for } \forall k, x_{5j}^k \cdot e^{-\lambda_{5j}^k} > 0 \quad (3)$$

$$\sum_j x_{2j}^k \cdot s_{2j} = \sum_j x_{1j}^k \cdot s_{1j} + \frac{\sum_j x_{5j}^k \cdot s_{5j}}{\sum_j x_{5j}^k} \text{ for } \forall k \quad (4)$$

$$\sum_j x_{2j}^k \cdot l_{2j} \geq \sum_j x_{5j}^k \text{ for } \forall k \quad (5)$$

$$\sum_j x_{1j}^k = 1 \quad (6)$$

$$\sum_j x_{2j}^k = 1 \quad (7)$$

$$\sum_j x_{3j}^k \leq 4 \quad (8)$$

$$\sum_j x_{4j}^k = 1 \quad (9)$$

$$x_{ij}^k = 0 \text{ or } 1 \quad (10)$$

## 1) 목적 함수

이 문제의 목적함수는 고객요구량에 맞게 재조립 컴퓨터를 만들기 위해 들어가는 부품 가격의 총합(C)을 낮추는 것이 목적이기 때문에 각각의 컴퓨터에 들어가는 부품 비용의 총합인 (1)과 같은 식으로 나타낼 수 있다.

## 2) 제약식

이 문제의 제약식으로는 크게 2가지 종류를 들 수 있는데, 고객의 요구사항에 대한 제약과 제품 조립시에 필요한 기술적인 제약이 있다. 첫 번째로 고객의 요구사항에 대한 제약은 각 부품의 성능에 대한 제약과 부품의 신뢰도에 대한 제약으로 나눌 수 있다. 각 부품의 성능에 대한 제약은 각 부품 별로 고객이 원하는 성능 이상인지를 확

인하는 것으로 (2)번 식으로 나타내었다. 신뢰도에 대한 제약은 컴퓨터의 주요 부품인 CPU, 메인보드, HDD, 파워공급장치, RAM은 한 개의 부품이라도 고장이 나면 컴퓨터가 멈추기 때문에 직렬연결로 간주하고 각 부품의 신뢰도의 곱으로 나타낸다. 여러 개의 부품을 사용할 수 있는 RAM과 HDD의 신뢰도는 따로 계산해야 한다. RAM은 여러 부품이 병렬로 연결되어 있더라도 한 개의 RAM 부품이라도 작동하지 않으면 컴퓨터를 정상적으로 사용할 수 없기 때문에 직렬연결로 간주한다. 하지만 HDD는 OS가 설치된 HDD가 고장이 날 때까지는 정상적으로 컴퓨터를 사용할 수 있기 때문에 HDD의 신뢰도는 OS가 깔린 HDD가 될 것이다. 본 연구에서는 조합가능한 HDD중 제일 신뢰도가 높은 HDD에 OS를 깔다고 가정하였다. 이것을 수식으로 나타내면 (3)과 같은 식이 된다.

두 번째로 제품의 기술적인 결합에 관한 제약은  $s_{ij}$ 에 관한 것으로 메인보드에 CPU와 RAM이 맞는지를 확인하는 제약식이다. 메인 보드의 조합적인 특성은 RAM과 CPU 특성의 합과 같아야 한다. 특히 RAM은 복수의 조합이 가능하여서 모든 부품의 특성들이 같아야 한다. 이것을 수식으로 나타내면 (4)와 같은 식이 된다. 예를 들면 특정 메인보드가 RAM은 DDR2를 사용해야 하고, CPU는 AMD 940 소켓을 사용해야 하면 메인보드의 호환정보는  $s_{2j} = 612$ 가 되어야 하고 RAM을 몇 개를 쓰던 그 호환정보는 맞아야 하며 이를 위해 제약식 (4)가 필요하다.

또한 전체 RAM의 개수는 메인보드의 RAM Slot수보다는 작거나 같아야 된다. 이를 위해 (5)번 제약식이 필요하다. (6)번, (7), (9)번 제약식들은 각각 CPU, 메인보드, 파워공급장치가 하나가 선택되어야 함을 나타내는 수식이고, (8)번 제약식은 조립된 컴퓨터의 사용 하드디스크의 개수가 최대 4개 이하이어야 한다는 가정사항을 만족시키기 위한 것이다. (10)번 제약식은 결정변수가 이진변수임을 나타내는 수식이다.

#### 4. 문제 해법

위에 문제를 풀기 위해서 본 연구에서는 2개의 휴리스틱 서치 알고리즘을 제안하였다. ACNS(Ant Colony and Neighborhood Search)와 DACS(Double Ant Colony Search)이다. 이 2가지 알고리즘 모두 기본적인 전개 방향은 유사하다. 둘 다 문제를 해결하기 위해 총 3단계를 거치게 되는데, 첫 번째 단계에서는 고객이 원하는 성능이상의 부품을 찾아 저가격순으로 정렬하고, 두 번째 단계에서는 부품 조합을 진행할 메인보드를 Ant colony 기법의 페로몬 개념을 활용하여 선별한다. 가장 먼저 메인보드를 선별하는 이유는 메인보드에 따라 사용되어질 수 있

는 CPU와 RAM 종류가 영향을 받기 때문이다. 세 번째 단계에서는 결정된 메인보드에 맞는 부품들을 결정하는 것이다. 즉, 메인보드에 기술적으로 맞는 부품 조합을 최저의 비용으로 선택하는 단계이다. 두 번째와 세 번째 단계가 끝나면 컴퓨터 한대 조립할 부품들이 선택되어지고, 고객이 원하는 수량이 될 때까지 두 번째와 세 번째 단계를 반복하여 문제의 해를 구하고, 이런 과정을 해가 개선되어질 때까지 반복하는 것이 2개 알고리즘의 기본 개념이다. 2개 알고리즘은 세 번째 단계인 메인보드에 맞는 부품을 선택하는 방식에서 차이를 가진다. ACNS는 부품을 선택할 때 이웃해 탐색 기법을 사용하고, DACS는 부품을 선택할 때 Ant Colony 기법을 사용하는 것이 차이점이다.

#### 4.1 ACNS(Ant Colony and Neighborhood Search) 휴리스틱

ACNS 휴리스틱은 크게 복수개의 RAM과 HDD의 조합 결정 단계, 메인보드의 결정 단계, 메인보드에 맞는 나머지 부품을 결정하는 단계로 이루어진다. 첫 번째 단계에서는 고객 사양에 맞는 RAM과 HDD의 가능한 조합들을 골라내고, 두 번째 단계에서는 Ant Colony의 페로몬 기법을 활용해 메인보드를 선택하게 된다. 세 번째 단계는 선택된 메인보드에 나머지 부품조합을 결정하는 단계로 Neighborhood search를 사용하게 된다. 자세한 세부 휴리스틱 절차를 설명하기 전에 이해를 돕기 위해 필요한 Notation들을 아래와 같이 정의한다.

- $M$  아주 큰 수(예, 1000000000)
- $C$   $N$ 개 중고 컴퓨터 조립 총비용
- $C^*$   $N$ 개 중고 컴퓨터 조립 총비용의 최선해
- $C_k$   $k$ 번째 중고 컴퓨터 조립 비용,  $k = 1, 2, 3, \dots, N$
- $C_k^*$   $k$ 번째 중고 컴퓨터 조립 비용의 최선해,  
 $k = 1, 2, 3, \dots, N$
- $X_k$   $k$ 번째 중고 컴퓨터 부품 조합의 현재해,  
 $k = 1, 2, 3, \dots, N$
- $X_k^*$   $k$ 번째 중고 컴퓨터 부품 조합의 개선해,  
 $k = 1, 2, 3, \dots, N$
- $X_k'$   $k$ 번째 중고 컴퓨터 부품 조합의 최선해,  
 $k = 1, 2, 3, \dots, N$
- $X'$   $N$ 개 중고 컴퓨터 부품 조합의 최선해,  
 $X' = \{X_k', k = 1, 2, 3, \dots, N\}$
- $p_{ij}$  부품종류  $i$ 의  $j$ 번째 부품에 부여된 페로몬 값,  
 $i = 1, 2, 3, 4, 5, j = 1, 2, 3, \dots, m$
- $\alpha$  임의의 상수

자세한 approach는 아래와 같다.

Step 1 : 고객의 요구사항에 맞는 부품 선택 및 조합, 그리고 초기화

가지고 있는 제품들 중에서 고객이 원하는 사양에 맞는 부품들을 선별한다. 그리고 해를 찾기 위한 초기화 작업을 한다.  $C^* = C_k^* = M$  for  $k = 1, 2, 3, \dots, N$ , iteration1 = 0, iteration2 = 0,  $k = 1$ 로 놓는다. 특히 부품들 중에서 RAM과 HDD는 복수개의 부품을 사용해서 조합하는 경우가 있기 때문에 복수개의 부품 조합의 경우도 생각해야 된다. RAM과 HDD의 복수개의 부품조합을 결정하는 방법은 아래와 같다.

(1) RAM의 부품 조합 방법

RAM은 부품의 종류가 3개(DDR1, DDR2, DDR3)를 가지고 있고, 메인보드에는 최대 4개의 RAM SLOT를 가지고 있기 때문에 효율적인 부품 조합을 결정하기 위해서 아래의 방법을 사용한다. 참고로 RAM은 한 개의 RAM이라도 고장이 나게 되면 컴퓨터가 정상적으로 작동하지 않기 때문에 신뢰성 분석측면에서 직렬연결로 간주한다.

- 각 RAM 종류들에 대해서,
- 가) 1개의 부품으로 가능한 부품들을 정렬한다.
- 나) 1개의 부품으로 가능한 부품들을 제외하고 2개의 조합으로 가능한 부품들을 정렬한다.
- 다) 2개까지의 부품조합이 포함되어 있는 조합을 제외하고 3개의 조합으로 가능한 부품들을 정렬한다.
- 라) 3개까지의 부품조합이 포함되어 있는 조합을 제외하고 4개의 조합으로 가능한 부품들을 정렬한다.

앞의 순서들 (가) (나) (다) (라)에서, 그전 과정에서 이미 결정된 부품에 부품을 추가해서 조합하게 되면 가격이 증가하게 되고, 신뢰성 측면에서도 떨어질 수밖에 없다. 따라서, 그전 과정에서 고려한 부품들을 제외시키고 가능한 부품들을 정렬한다.

(2) HDD의 부품조합 방법

HDD는 메인 드라이브만 고장나지 않는다면 컴퓨터는 정상적으로 작동할 수 있다. 그 때문에 HDD의 신뢰도는 조합되는 부품 중에서 가장 큰 신뢰도를 가지는 것을 메인 드라이브로 결정하고, 그 부품의 신뢰도를 HDD 조합 전체의 신뢰도로 결정한다. HDD의 조합은 앞에서 서술했던 RAM 조합방식과 같이 조합한다.

Step 2 : k번째 메인보드 선택하고 메인보드에 맞는 부품 조합을 list up

남아 있는 메인보드들에서 k번째 메인보드를 선택하기 위한 방법으로 Ant colony search 휴리스틱 기법의 페로몬 수치를 활용한다. 즉, 메인보드는 자신만의 페로몬 값( $p_{2j} = 1/c_{2j} / \sum_j (1/c_{2j})$ ,  $j = 1, 2, 3, \dots, m$ )을 가지고 있다고 가정하자. 이 값들은 해를 찾는 과정에 따라 변하게 되고, 각각의 페로몬 값의 확률로 메인보드가 선택되어진다. 메인보드 선택 후, 메인보드에 맞는 부품 조합을 나열한다.

Step 3 : 가능한 부품(조합)을 저가격순으로 정렬하고 초기 부품 조합해(P)를 결정

Step 2에서 선택된 메인보드에 맞는 부품(혹은 부품조합)들을 저가격순으로 정렬시킨 후 초기 부품 조합해를 결정한다. 초기 부품 조합해는 부품별로 가장 작은 비용을 갖는 부품들의 조합으로 이루어진다.

Step 4 : 초기 부품 조합해의 가능해 여부 판단 위에서 정한 초기 부품 조합해로 컴퓨터를 조립했을 때, 신뢰도 측면에서 가능해인지를 판단한다. 만약 가능해가 아니라면, Step 5로 이동하고, 가능해라면, Step 6으로 이동한다.

Step 5 : 이웃해탐색 기법을 이용한 가능 부품 조합해 생성 만약 Step 4에서 초기해가 제약조건을 만족하지 않는다면, 다음 부품조합을 결정하기 위해서

- (1) 랜덤으로 4개의 부품(CPU, HDD, 파워공급장치, RAM) 중 1개의 부품을 선택한다.
- (2) 선택된 부품종류에 대해 정렬되어 있는 순서에서 앞이나 뒤로 랜덤하게 이동시킨 후 부품을 선택하는 방식으로 새로운 부품조합해를 만든다. 이때 새로운 부품조합해가 가능해라면 그 부품 조합해를 사용하고, 가능해가 아니라면, 다시 (1)~(2)를 반복하여 가능한 부품 조합해( $X_k$ )를 찾고 결정된 부품 조합해의 조합비용( $C_k$ )을 계산한다.
- (3) 결정된 부품 조합해의 비용( $C_k$ )이 현재 선택된 메인보드의 부품조합해 비용( $C_k^*$ )보다 작은지 비교한다. 만약  $C_k < C_k^*$ 이면,  $X_k^* = X_k$ ,  $C_k^* = C_k$ , iteration1 = 0라 두고 (1)로 이동한 후 (2)를 수행한다. (2)번 수행시에 초기 부품조합해는  $X_k^*$ 로 놓는다.
- (4) 만약  $C_k \geq C_k^*$ 이면, iteration1 = iteration1 + 1로 한다. iteration1이 100을 초과하면 그때까지의  $X_k^*$ 을 Best 부품조합해로 선정하고 다음 Step 6으로 이동한다. iteration1이 100을 초과하지 않으면 (1)로 이동한다.

Step 6 : 고객이 요구하는 수량의 컴퓨터 부품 조합을 찾았는지 체크

- (1) 만약  $k = N$ 이면 Step 7로 이동한다.
- (2) 만약  $k < N$ 이면,  $k = k + 1$ 로 놓고 Step 2로 이동한다.

Step 7 : N개 컴퓨터 조립 총합비용의 계산

고객이 원하는 수량만큼의 컴퓨터의 부품 조합해들 ( $X_k^*$ ,  $k = 1, 2, 3, \dots, N$ )이 결정되어졌으면 총비용( $C$ )을 계산한다.

- (1) 이 비용이 지금까지의 최적해( $C^*$ )보다 좋으면 (즉, 만약  $C < C^*$ 이면), 해를 갱신한다. 즉,  $C^* = C$ ,  $X_k' = X_k^*$  for  $k = 1, 2, 3, \dots, N$ 라 놓고 iteration2 = 0로 놓는다. 이 비용이 지금까지의 최적해( $C^*$ )보다 나쁘면, iteration2 = iteration2 + 1로 둔다.
- (2) 만약 iteration2 > 100이면 Step 9로 간다. 만약 iteration2 ≤ 100이면, Step 8로 간다.

Step 8 : 메인보드 페로몬 재계산 및 부여

- (1) 해의 비교가 끝나면 모든 메인보드의 페로몬은 일정 비율(즉,  $p_j = 0.95 \cdot p_j$ ,  $j = 1, 2, 3, \dots, m$ )로 감소시킨다.
- (2) 그 후에 선택된 메인보드들에 대해  $\alpha/C_k^*$  ( $k = 1, 2, 3, \dots, N$ )을 계산한 후 선택된 메인보드들의 기존 페로몬 값( $p_{2j}$ )에 더한다. 만약 iteration2 = 0이라면 해가 개선되어진 것이므로 기존 페로몬 값( $p_{2j}$ )에  $2 \cdot \alpha/C_k^*$ 을 더한다.
- (3)  $C_k^* = M$  for  $k = 1, 2, 3, \dots, N$ , iteration1 = 0, iteration2 = 0,  $k = 1$ 로 두고 step 2로 간다.

Step 9 : 최종해 결정

지금까지 나온 최적해 ( $C^*$ ,  $X^*$ )를 최종해로 결정한다.

## 4.2 DACS(Double Ant Colony Search) 휴리스틱

DACS휴리스틱은 기본적으로 ACNS와 동일한 방식으로 진행된다. 하지만 세 번째 단계인 메인보드에 맞는 부품들을 결정하는 단계에서도 Ant Colony 기법을 사용한다는 것이 ACNS와 다르다. 이 DACS 휴리스틱의 세부 절차는 아래와 같다.

Step 1 : 고객의 요구사항에 맞는 부품 선택 및 조합, 그리고 초기화

가지고 있는 제품들 중에서 고객이 원하는 사양에 맞는 부품들을 선별한다. 그리고 해를 찾기 위한 초기화 작업을 한다. 즉,  $C_k^* = C_k = M$  for  $k = 1, 2, 3, \dots, N$ ,

iteration1 = 0, iteration2 = 0,  $k = 1$ 로 놓는다.

Step 2 : k번째 메인보드 선택하고 메인보드에 맞는 부품 조합을 list up

남아 있는 메인보드들에서 k번째 메인보드를 선택하기 위한 방법으로 Ant colony search휴리스틱 기법의 페로몬 수치를 활용한다. 즉, 메인보드는 자신만의 페로몬 값( $p_{2j} = 1/c_{2j} / \sum_j (1/c_{2j})$ ,  $j = 1, 2, 3, \dots, m$ )을 가지고 있다고 가정하자. 이 값들은 해를 찾는 과정에 따라 변하게 되고, 각각의 페로몬 값의 확률로 메인보드가 선택되어진다. 메인보드 선택 후, 메인보드에 맞는 부품 조합을 나열한다.

Step 3 : 페로몬 개념을 응용한 초기 부품 조합해를 선택하고 신뢰도측면의 가능해 선별

- (1) 각 부품별로,  $p_{ij} = 1/c_{ij} / \sum_j (1/c_{ij})$ ,  $i = 1, 3, 4, 5, j = 1, 2, 3, \dots, m$ 를 계산한 후, 각 부품별로  $p_{ij}$ 의 확률로 1개의 부품을 뽑아 부품 조합해를 만든다. (가격이 낮을수록 뽑힐 확률이 더 크다).
- (2) 뽑힌 부품 조합해가 신뢰도 관련 제약조건을 만족하는지 체크한다. 만약 제약조건을 만족하면, 즉 가능해이면, 선택된 부품 조합해의 조합비용( $C_k$ )을 계산하고,  $C_k^* = C_k$ ,  $X_k^* = X_k$ 로 놓고 step 4로 간다. 만약 가능하지 않으면, (1)을 가능해가 뽑힐 때까지 반복한다.

Step 4 : 가능한 부품조합( $X_k$ ) 선택하에서 페로몬 갱신

- (1) 각 부품별(메인보드 제외)로 일정비율(0.05) 페로몬을 감소시키고, 선택된 부품에 대해 일정양 페로몬을 증가시킨다. 즉,
  - $p_{ij} = 0.95 \cdot p_{ij}$ ,  $i = 1, 3, 4, 5, j = 1, 2, 3, \dots, m$
  - 각 부품 종류별로 선택된 부품  $j$ 에  $p_{ij} = p_{ij} + \alpha/C_k$  for  $i = 1, 3, 4, 5$

Step 5 : 가능한 부품조합해 탐색

- (1) 각 부품별(메인보드제외)로  $p_{ij}$ 의 확률로 1개의 부품을 뽑아 부품 조합해를 만든다.
- (2) 신뢰도 관련한 제약을 만족하는지 체크한다. 만족하지 않으면 (1)을 반복해 가능한 부품조합해를 찾는다.
- (3) 가능한 부품조합해에 대해 조합비용( $C_k$ )을 계산하고,  $C_k^*$ 와 비교한다.  $C_k^*$ 보다 작으면,  $C_k^* = C_k$ ,  $X_k^* = X_k$ , iteration1 = 0으로 둔다.  $C_k^*$ 보다 크면, iteration1 = iteration1+1로 둔다.



- (3) 만약 iteration1이 100을 초과했으면 step 6으로 이동한다. 초과하지 않았으면 step 4로 간다.

Step 6 : 고객이 요구하는 수량의 컴퓨터 부품 조합을 찾았는지 체크

- (1) 만약  $k = N$ 이면 Step 7로 이동한다.
- (2) 만약  $k < N$ 이면,  $k = k + 1$ 로 놓고 Step 2로 이동한다.

Step 7 : N개 컴퓨터 조립 총합비용의 계산

고객이 원하는 수량만큼의 컴퓨터 부품 조합해들( $X_k^*$ ,  $k = 1, 2, 3, \dots, N$ )이 결정되어졌으면 제품의 총비용( $C$ )을 계산한다.

- (1) 이 비용이 지금까지의 최적해( $C^*$ )보다 좋으면(즉, 만약  $C < C^*$ 이면), 해를 갱신한다. 즉,  $C^* = C$ ,  $X_k' = X_k^*$  for  $k = 1, 2, 3, \dots, N$ 라 놓고 iteration2 = 0로 놓는다. 이 비용이 지금까지의 최적해( $C^*$ )보다 나쁘면, iteration2 = iteration2+1로 둔다.
- (2) 만약 iteration2 > 100이면 Step 9로 간다. 만약 iteration2 ≤ 100이면, Step 8로 간다.

Step 8 : 메인보드 페로몬 재계산 및 부여

- (1) 해의 비교가 끝나면 모든 메인보드의 페로몬은 일정 비율(즉,  $p_j = 0.95 \cdot p_j$ ,  $j = 1, 2, 3, \dots, m$ )로 감소시킨다.
- (2) 그 후에 선택된 메인보드들에 대해  $\alpha / C_k^*$  ( $k = 1, 2, 3, \dots, N$ )을 계산한 후 선택된 메인보드들의 기존 페로몬 값( $p_{2j}$ )에 더한다. 만약 iteration 2 = 0이라면 해가 개선되어진 것이므로 기존 페로몬 값( $p_{2j}$ )에  $2 \cdot \alpha / C_k^*$ 을 더한다.
- (3)  $C_k^* = M$  for  $k = 1, 2, 3, \dots, N$ , iteration1 = 0, iteration2 = 0,  $k = 1$ 로 두고 step 2로 간다.

Step 9 : 최종해 결정

지금까지 나온 최적해( $C^*$ ,  $X'$ )를 최종해로 결정한다.

## 5. 실험 계획 및 결과

### 5.1 실험 Data셋 정의

모든 제품의 Data셋을 정의하기 위해서 프랑스의 한 회사에서 실제로 조사된 6개월~1년간의 각 부품의 고장률[10] 자료를 기초로 해서 각 부품들의 평균 수명 및 고장률 함수를 구하였다(<Table 3> 참조). 그 뒤 제품의 평균 사용 시간을 추정하였다. 평균 수명과 기사용시간으로 남은 잔여 수명 시간을 구하고, 새 제품의 가격에서

남은 잔여수명/평균수명 값을 곱해서 중고 부품의 가격을 결정하였다. 그 뒤 각 부품의 보장기간까지의 신뢰도를 계산하고, 제품별로 필요한 성능 및 조합정보를 결정하였다. 추가적으로 중고부품으로 모든 조합이 생성되지 않는 경우가 있기 때문에 그런 경우에 추가적으로 새로운 제품을 투입해서 제품조합을 결정하였다.

<Table 3> Average Failure Fate of Computer Part (Source : hardare.fr, unit : AFR)

Brand	MB	POWER	RAM	HDD	CPU
Maker 1	0.016	0.01	0.004	0.015	0.0071
Maker 2	0.019	0.013	0.007	0.018	0.0279
Maker 3	0.02	0.02	0.016	0.02	0.0521
Maker 4	0.023	0.016	0.02	0.031	0.0097
Maker 5	-	0.23	0.071	-	-

본 연구에서는 제시된 두개 알고리즘들의 유용성을 평가하기 위해서 알고리즘들을 MATLAB로 구현하였고, 실험 문제 데이터들을 생성하고, Intel core(TM)2 DUO CPU 2.93Hz의 컴퓨터를 사용하여 실험을 실시하였다. 실험 문제 데이터들은 작은 사이즈의 문제들과 큰 사이즈의 문제들로 구분하였다.

- (1) 실험할 문제의 set을 다음과 같이 정의하였다.: ( $m, N, t$ ),  $m$  = 부품의 수,  $N$  = 고객이 주문한 컴퓨터 수,  $t$  = 보장기간. 작은 사이즈 문제에 대해서는  $m$ 는 5~10개 부품,  $N$ 은 1~2개,  $t$ 는 1년~3년의 조합으로 문제 set을 구성하였다. 큰 사이즈 문제에 대해서는  $m$ 는 20, 50, 100, 300개 부품,  $N$ 은 5, 10, 20, 50, 100개,  $t$ 는 1년~3년의 조합으로 문제 set을 구성하였다.
- (2) 각 문제 SET에 대해 각각 10개의 문제를 생성하여 풀어보았다. 각 문제에 대해 10번의 반복실험을 수행하였다.
- (3) 제품의 보장기간은 1년에서 3년 사이로 가정하였고, 각 보장기간동안 하루에 평균 2시간 컴퓨터가 사용된다고 했을 때, 보장기간동안에 신뢰도가 70% 이상이 되는 제품만 판매가 가능하다고 가정하였다.
- (4) 작은 사이즈의 문제들에 대해서는, 제시한 휴리스틱 해들과 실제 full enumeration방법을 이용해서 얻은 최적해와 비교하였다. 이를 위해 PE(Percentage Error) 지표를 사용하였다. PE는  $100 \cdot |H - O| / H$ 로 최적해와 얼마나 유사한가를 나타내는 지표이다. 여기서  $H$ 는 각 문제를 10번씩 돌렸을 때에 나오는 값 중 제일 좋은 값,  $O$ 는 최적해를 나타낸다.
- (5) 큰 사이즈의 문제들에서는 최적해를 구하기 어렵기 때문에, 다른 휴리스틱과의 성능 비교를 통해서 알고리즘의 성능을 비교하였다. 이를 위해 RPE(Relative Percentage

Error) 지표를 사용하였다. RPE는  $100 \cdot |H - H^*| / H^*$ 로 나타내며, 제안한 휴리스틱 기법이 상대적으로 얼마나 좋은 성능을 나타내는지 알려주는 지표이다.  $H$ 는 제안한 휴리스틱 기법의 10번 반복실험 뒤의 제일 좋은 값,  $H^*$ 는 두 개의 휴리스틱 값 중에서 가장 좋은 값을 나타낸다.

- (6) 작은 사이즈의 문제에 대해서는 CPU 소켓의 종류를 2개로 제한하였다. 큰 사이즈의 문제에 대해서는 원래 대로 17 CPU 소켓을 다 고려하였다.
- (7) 고객의 컴퓨터에 대한 요구사항은 최소한 CPU 2.3 GHz, HDD 250G, 파워공급장치 350W, RAM 512M 이상이라고 가정하였다.
- (8) 제시된 두개의 휴리스틱 알고리즘에서 메인보드와 부품조합의 선택 및 해의 개선에 있어서 100번 이상 반복해서 탐색해도 해가 개선되지 않으면 다음단계로 넘어가는 것으로 정하였다.
- (9)  $M = 1000000000$ ,  $\alpha = 1000$ 으로 두었다.

### 5.2 실험 결과

<Table 4>는 작은 문제 size에서의 실험 결과를 보여준다.

ACNS의 경우 작은 문제 사이즈에 대해서는 PE값이 3.17%, DACS의 경우는 3.21% 값이 나왔다. 따라서 작은 문제 사이즈의 경우는 두 휴리스틱 알고리즘간의 큰 성능차이가 존재하지 않는 것을 알 수 있다. 수요가 1, 2 일 경우를 각각 놓고 보더라도 두 휴리스틱 알고리즘간의 성능차이는 없다고 볼 수 있다(<Figure 2>, <Figure 3> 참조). 실험 소요시간은 문제가 커질수록, 고객이 요구하는 제품의 대수가 커질수록 최적 해를 구하는 것과의 계산시간 차이가 점점 커지는 것을 볼 수 있다. 특히 제안된 휴리스틱 시간의 증가보다 최적 해를 찾는 full enumeration방법의 시간 증가가 더 급속도로 증가하는 것을 볼 수 있다. 현실적으로 부품의 수가 큰 재조합 문제에서는 많은 경우의 수가 생기기 때문에 최적 해를 구하는 것은 현실적으로 불가능하다. 그렇기 때문에 제안된 휴리스틱을 사용하는 것이 최적 해와의 차이가 크지 않으면서 빠른 시간 내에 결과값을 도출 할 수 있는 방법이라고 생각한다.

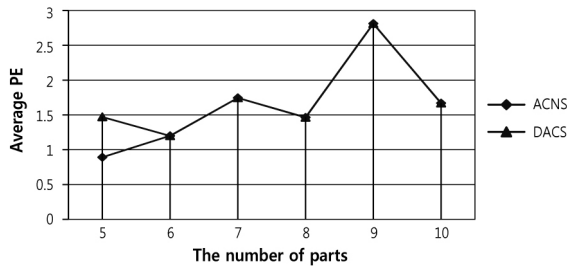
위의 <Table 5>는 큰 문제 사이즈에 대한 실험 결과를 보여준다. 이를 보면 DACS의 RPE는 1.058%, ACNS의 RPE는 3.137%임을 알 수 있다. 즉, DACS 휴리스틱이 ACNS 휴리스틱보다 상대적으로 적은 시간에 더 좋은 해를 나타내었다. 특히 해를 찾아내는 시간에서도 부품의 셋이 50개를 넘어가는 이후부터는 DACS 휴리스틱이 빠르게 해를 구해내는 편이다(<Figure 4> 참조). 각 보장기간에 따른 휴리스틱 해들의 성능을 비교해 보면, 보장

기간이 1년일 때는 오히려 ACNS가 DACS보다 좋은 해를 주지만(<Figure 5> 참조), 보장기간이 2년, 3년인 경우는 DACS가 훨씬 좋은 성능을 보여줌을 알 수 있다(<Figure 6>, <Figure 7> 참조). 이는 보장기간이 길면 갈수록 가능한 부품조합을 구하는 것이 더욱 어려워지기 때문에 DACS가 ACNS보다 보다 복잡한 문제들을 푸는데 좋은 해를 제공한다고 볼 수 있다.

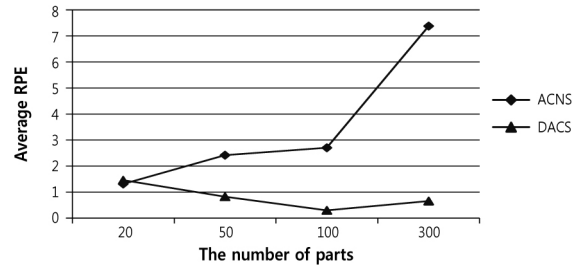
<Table 4> Test Results in Small-sized Problems

Problem Sets	Opt.	ACNS			DACS		
	Time (sec)	Time (sec)	Aver. PE	#	Time (sec)	Aver. PE	#
(5, 1, 1)*	0.01	0.84	1.30	8	0.17	1.30	8
(5, 1, 2)	0.01	1.10	0.22	8	0.23	0.22	8
(5, 1, 3)	0.00	0.12	1.15	9	0.10	2.89	9
(6, 1, 1)	0.02	0.89	0.65	8	0.17	0.65	8
(6, 1, 2)	0.02	1.34	1.69	7	0.23	1.69	7
(6, 1, 3)	0.04	0.26	1.26	9	1.29	1.26	9
(6, 2, 1)	21.92	2.07	2.61	6	0.38	2.61	6
(6, 2, 2)	2.87	2.04	5.38	6	0.41	5.38	6
(6, 2, 3)	2.55	3.07	3.26	7	0.66	3.26	7
(7, 1, 1)	0.02	0.90	2.91	8	0.16	2.91	8
(7, 1, 2)	0.03	1.13	2.32	8	0.21	2.32	8
(7, 1, 3)	0.02	0.27	0	10	1.06	0	10
(7, 2, 1)	157.04	2.14	3.97	4	0.36	3.97	4
(7, 2, 2)	8.24	2.38	4.95	6	0.54	4.95	6
(7, 2, 3)	4.87	3.23	2.93	6	0.78	2.93	6
(8, 1, 1)	0.03	0.94	2.47	9	0.17	2.47	9
(8, 1, 2)	0.06	1.23	0.55	9	0.29	0.56	9
(8, 1, 3)	0.02	0.40	1.37	8	1.84	1.37	8
(8, 2, 1)	46.69	1.93	4.03	5	0.32	4.03	5
(8, 2, 2)	34.97	2.53	6.94	4	0.54	6.94	4
(8, 2, 3)	47.08	3.23	5.46	5	0.91	5.46	5
(9, 1, 1)	0.03	0.94	4.11	6	0.21	4.11	6
(9, 1, 2)	0.07	1.10	1.67	8	0.20	1.67	8
(9, 1, 3)	0.02	0.30	2.65	7	0.99	2.65	7
(9, 2, 1)	157.04	2.14	3.97	4	0.36	3.97	4
(9, 2, 2)	129.91	2.23	3.49	6	0.41	3.50	6
(9, 2, 3)	76.73	7.43	2.35	6	10.52	1.58	6
(10, 1, 1)	0.081	1.00	3.28	7	0.19	3.28	7
(10, 1, 2)	0.11	1.12	0	10	0.19	0	10
(10, 1, 3)	0.09	0.41	1.72	8	1.34	1.72	9
(10, 2, 1)	440.73	2.07	6.63	4	0.36	7.22	4
(10, 2, 2)	601.19	2.55	10.69	3	1.03	10.69	3
(10, 2, 3)	151.30	3.21	8.63	5	1.16	8.63	5
Total average	57.08	1.71	3.17	6.8	0.84	3.22	6.8

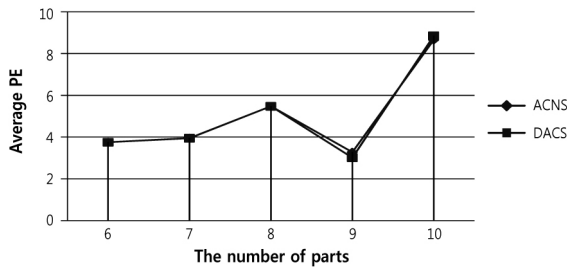
\*( $m, N, t$ )  $m$ =the number of part,  $N$ =the number of ordered computers,  $t$ =warranty period, # the number of optimal solutions



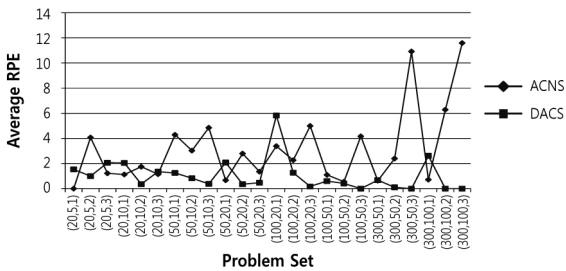
<Figure 2> PE Comparison in Small-sized Problems(N= 1)



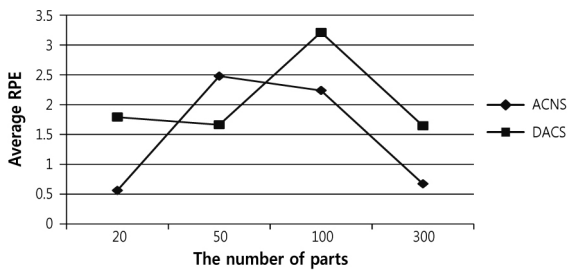
<Figure 7> RPE Comparison According to the Number of Parts (When the Warrant Period is Three Years)



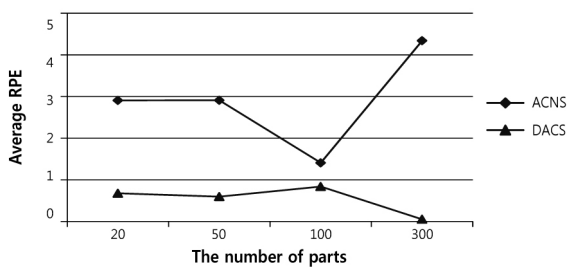
<Figure 3> PE Comparison in Small-sized Problems(N= 2)



<Figure 4> RPE Comparison in Large-sized Problems



<Figure 5> RPE Comparison According to the Number of Parts (When the Warrant Period is One Year)



<Figure 6> RPE Comparison According to the Number of Parts (When the Warrant Period is Two Years)

<Table 5> Test Results in Large-Sized Problems

Problem Sets	ACNS			DACS		
	Time (sec)	Aver. RPE	#	Time (sec)	Aver. RPE	#
(20, 5, 1)*	0.69	0	10	0.19	1.54	8
(20, 5, 2)	2.95	4.07	2	12.91	0.99	8
(20, 5, 3)	4.06	1.22	6	12.67	2.07	4
(20, 10, 1)	2.32	1.13	5	13.51	2.05	5
(20, 10, 2)	3.44	1.74	3	16.41	0.36	6
(20, 10, 3)	8.67	1.15	5	27.02	1.35	5
(50, 10, 1)	2.37	4.29	5	1.16	1.25	5
(50, 10, 2)	9.23	3.03	2	31.56	0.84	8
(50, 10, 3)	18.08	4.85	2	42.25	0.38	8
(50, 20, 1)	42.62	0.67	6	13.46	2.08	4
(50, 20, 2)	15.59	2.79	3	60.23	0.35	7
(50, 20, 3)	42.41	1.35	4	87.61	0.48	6
(100, 20, 1)	7.98	3.38	7	6.90	5.83	3
(100, 20, 2)	60.93	2.27	4	83.34	1.27	6
(100, 20, 3)	92.66	4.99	2	105.11	0.17	8
(100, 50, 1)	101.66	1.09	4	173.23	0.60	6
(100, 50, 2)	116.38	0.55	4	190.72	0.41	6
(100, 50, 3)	268.26	4.16	0	271.42	0	10
(300, 50, 1)	1873.56	0.63	2	920.84	0.68	8
(300, 50, 2)	1788.39	2.40	2	999.29	0.10	8
(300, 50, 3)	2850.07	10.92	0	1765.36	0	10
(300, 100, 1)	107.99	0.72	7	187.11	2.61	3
(300, 100, 2)	1230.07	6.29	0	1220.34	0	10
(300, 100, 3)	2396.42	11.59	0	2677.1	0	10
Total average	460.28	3.14	3.5	371.65	1.058	6.8

\*(m, N, t) m = the number of parts, N = the number of ordered computers, t = warranty period, # = the number of best solutions.

## 6. 결 론

최근까지 가장 활발히 재제조 프로세스가 적용된 분야는 자동차 산업이고, 최근에 점점 다양한 산업으로 발전되고 있다. 그 중에서 컴퓨터는 짧은 제품주기로 인해 버려지는 경우가 많기 때문에 버려진 컴퓨터들의 부품들을 재사용하는 방안에 대한 연구가 필요하다고 할 수 있다. 본 연구에서는 쓰다 버려진 컴퓨터 부품들을 대상으로 고객이 요구하는 제품의 성능 및 제품의 보장기간 등을 고려해 복수개의 컴퓨터를 가장 적은 비용으로 조합하는 문제를 풀기 위한 방법을 연구하였다. 위와 같은 문제를 풀기 위해 본 연구에서는 휴리스틱 기법 중에서 Ant colony Search 휴리스틱과 Neighborhood Search 휴리스틱 기법을 사용하였다. 위와 같은 방법이 효율적으로 해를 찾는지 확인하기 위해서 적절한 예들을 생성한 후 휴리스틱 해의 검증을 해보았다.

본 연구의 추후 연구주제들로 다음과 같은 것을 생각해 볼 수 있다. 이 논문에서는 중고 제품의 사용시간만을 감안하여 재사용 가치를 고려하였다. 하지만 각 제품들의 사용 환경정보들을 얻을 수만 있다면 이를 반영해서 제품의 고장률 및 신뢰도 등을 추정하는 방법에 대한 연구도 필요하다. 사용자의 손을 떠나서 흘러들어온 컴퓨터가 비교적 사용 시간이 긴 산업체에서 사용되었는지, 아니면 상대적으로 적게 사용되는 가정 등에서 사용되었는지에 따라서 제품의 신뢰도 및 잔존수명을 결정해서 최적의 의사결정을 수행하는 연구도 필요하다고 생각한다. 또한, 주어진 재고 조건하에서 일정 사양의 제조립 컴퓨터를 가능한 한 많이 만들어낼 수 있는 방안에 관한 연구도 고려해볼 수 있다. 추가적으로 제품의 범위를 확장시켜서 5개의 핵심 부품이 아니라, 추가적인 부품들(VGA, 모니터 등)을 포함한 최적의 부품조합에 관한 연구가 필요하다고 생각한다.

## Acknowledgement

이 논문은 2011년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(2011C103401). 또한, 2011학년도 홍익대학교 학술연구진흥비에 의하여 지원되었음.

## Reference

- [1] Mok, H.S., Jeon, C.S., Han, C.H., Park, S.J., Sagong, H., and Seliger Günther, Remanufacturing Industry for Automobile Parts of European. *Transactions of KSAE*, 2011, Vol. 19, No. 1, p 38-44.
- [2] Mok, H.S., Jeon, C.S., Han, C.H., Steven J. Skerlos, Kim, H.J., and Lee, K.C., Remanufacturing Industry for Automobile Parts of USA. *Journal of the Korean Society for Precision Engineering*, 2010, Vol. 27, No. 3, p 58-65.
- [3] Yi, H.-C., Kang, H.-Y., Kim, J.H., Shim, K.S., Kim, J.H., and Han, S.C., A Study on the Establishment of the Standards for Recycling Rate of Parts and Materials to Calculate Recyclability Rate of Electrical and Electronic Equipments. *Clean Technology*, 2008, Vol. 14, No. 4, p 232-241.
- [4] Bufardi, A., Sakara, D., Gheorghe, R., Kiritsis, D., and Xirouchakis, P., Multiple criteria decision aid for selecting the best product end of life scenario. *International Journal Computer Integrated Manufacturing*, 2003, Vol. 16, No. 7-8, p 526-534.
- [5] Bufardi, A., Gheorghe, R., Kiritsis, D., and Xirouchakis, P., Multicriteria decision-aid approach for product end-of-life alternative selection. *International Journal of Production Research*, 2004, Vol. 42, p 3139-3157.
- [6] Choi, B., Shin, H., Lee, S., and Hur, T., Life Cycle Assessment of a Personal Computer and its Effective Recycling Rate. *International Journal of Life Cycle Assessment*, 2006, Vol. 11, p 122-128.
- [7] Lee, C., Chang, S., Wang, K., and Wen, L., Management of scrap computer recycling in Taiwan. *Journal of Hazardous Materials*, 2000, Vol. A73, p 209-220.
- [8] Melissen, F. W. and de Ron, A., Decision supporting model for the recovery of computer components. *International Journal Environment and Sustainable Development*, 2002, Vol. 1, No. 2.
- [9] Steinhilper, R., *Remanufacturing*, Korea National Cleaner Production Center, 2005.
- [10] <http://gigglehd.com/zbxe/5517484>.