

맞춤구성을 위한 템플릿과 Option 기반의 추론

이현정*, 이재규**

요 약

전자 카탈로그 상에서의 상품 검색은 카탈로그에 명시되어 있는 상품을 찾는 표준상품검색과 소비자가 원하는 상품을 맞춤 하는 맞춤상품검색으로 분류할 수 있다. 현재의 대부분의 상품 검색은 표준상품 검색에 의존하고 있다. 특히 기업간 구성요소기반(Component-based) 상품의 경우 표준상품검색만으로는 구매자의 다양한 요구에 응하기가 어렵다. 따라서 웹 상의 전자 카탈로그에서의 동적인 맞춤검색에 대한 요구가 증가하고 있다.

본 연구에서는 구성기반 상품에 대해서 표준상품검색만으로는 구매자가 원하는 상품의 검색가능성(Feasibility)과 검색된 대안들이 조정(Adjust) 프로세스 과정을 거쳐 최적해 도달 가능성(Admissibility)이 보장되지 않음을 보이고, 이에 대한 효과적인 방법론으로 검색가능성과 최적해 도달 가능성을 지원하는 Template-based Reasoning 방법론을 제안 한다. Template-based Reasoning은 구매자의 요구사항에 따른 대안탐색 부분과 선택된 대안에 대한 조정과정의 두 단계로 이루어 진다. 구매자의 주요 선호도(MUST Preference)에 근거하여 대안들을 탐색하고, 탐색 된 대안들 간의 우선순위를 결정한다. 조정 단계에서는 옵션(Options)의 확장을 통해 구매자의 맞춤사양에 따른 상품을 제안하고, 제약 및 규칙기반 추론(Constraint and Rule Satisfaction Approach)을 이용하여 옵션(Options)들 간의 제약조건에 따른 호환성(Compatibility)을 조사하고, 적정가격의 상품을 제안한다.

본 방법론은 Template을 사용하여 기본적으로 구매자가 원하는 상품을 검색하기위한 검색노력을 줄이고, 검색된 대안들로부터 구매자와 시스템이 웹상에서 서로 상호작용(interactivity) 하여 해를 찾고, 제약조건과 규칙 들에 의해 적합한 해를 찾아 가는 방법을 제시한다. 본 논문은 구성기반 예로서 컴퓨터 부품 조립을 사용해서 Template-based reasoning 예를 보인다. 본 방법론은 검색노력을 줄이고, 검색에 있어 Feasibility와 Admissibility를 보장한다.

Keywords: Mass Customization, Component-based Product, Configuration, Constraint-and-Rule Satisfaction Problem, Mass Customization, Feasibility, Admissibility, Business-to-Business

1. 서론

웹의 등장이 전자상거래의 괄목할 만한 성장에 중심점이 된 것은 사실이다. 웹에서 진열 가능한 정보의 양은 무한하고, 이에 따라 정보를 검색하기 위한 다양한 방법들이 제안 되어졌다. 웹상에서의 상품 검색은 카탈로그에 명시되어 있는 상품을 검색하는 표준 상품 검색과 구매자가 원하는 상품을 맞춤 하는 맞춤상품 검색의 방법으로 분류된다. 구성기반 상품(Component-based Product)의 경우 단순 표준 상품 검색만으로 구매자의 요구사항에 맞는 상품을 검색하기란 쉽지 않다.

현재 대표적으로 유용한 웹 기반 온라인 쇼핑으로는 테이블 형식을 빌어 상품의 세부사양을 단계별로 비교하는 비교쇼핑(Comparison Matrix)과 구매자의 성향에 기반해 상품을 제안(Recommendation Agent)하는 쇼핑 시스템이 있다. 비교쇼핑은 데이터베이스를 이용한 단순 탐색에 주로 의존한다. 구매자의 요구조건에 있어 항목마다의 구매자의 가중치 등은 크게 고려되어지지 않는다. 따라서 구매자의 요구사항에 부합하는 동적인 상품의 제안에는 한계가 있다. 구매자성향 기반 상품 제안 시스템의 경우 구매자의 성향에 기반 한 상품을 검색 제안하는

기능을 제공한다. 한계점은 구매자가 요구하는 상품이 데이터베이스에 검색 되어 있지 않으면 상품을 제안하지 못한다는 점이다. 그렇다고 모든 가능한 상품의 조합을 데이터베이스에 저장하는 일도 효율적이진 않다. 선택된 상품이 구매자의 요구에 부합하지 않으면 대안들의 조정(Adjust)이 허락되나, 조정된 대안이 양질의 저가상품이라는 해를 보장하지는 않는다.

본 연구에서는 상품의 특성에 따라 표준상품검색과 맞춤상품검색이 유용한 경우를 살피고, 구매자가 원하는 상품을 검색할 수 있는 검색가능성(Feasibility)과 검색된 대안의 조정(Adjust)과정에서 최선의 해에 도달 가능한 지(Admissibility)에 대한 보장성에 대해 비교 분석한다. 표준검색의 경우 단순 검색 만으로 구매자의 요구사항을 만족하는 해를 찾기는 어렵다. 선택된 대안이 조정과정을 거치더라도 항상 최적의 해를 보장하지는 않는다.

따라서 검색가능성과 최적해 도달 가능성을 보장할 수 있는 Template-based Reasoning 방법론을 제안한다.

Template-based Reasoning은 대안 검색과정과 대안 조정과정의 두 단계로 이루어진다. 대안 검색과정은 구매자가 주요하게 여기는 세부사양 (Factor)의 선호도에 기반해서 검색한다. 이 과정에서는 검

* 한국과학기술원 테크노경영대학원 경영공학과, E-mail: leehj@kgsms.kaist.ac.kr

** 한국과학기술원 테크노경영대학원, E-mail: jklee@kgsms.kaist.ac.kr

색 사이즈를 줄일 수 있는 장점이 있다. 대안 조정 과정에서는 선택된 대안들의 Options의 확장을 통해 구매자의 요구를 만족하는 최선의 해를 찾는다. 이 과정에서는 Options 확장에 따른 제약조건을 조사하고, 수정된 Option들간의 충돌해결 (Conflict Resolution)을 위해 제약 규칙 기반 추론 (Constraint and Rule Satisfaction Reasoning)을 이용한다. 구성기반의 상품의 예로 컴퓨터 부품을 이용한 맞춤 구성을 보인다.

본 논문의 구성은 인공지능기법 및 구성기반(Configuration) 관련 문헌들을 조사하고, 인터넷에서의 구매 의사결정 지원시스템에 대해 살펴본다. 제안하는 Template-based Reasoning 시스템은 대안검색과 대안조정 두 단계로 이루어 진다. 예제로 사용된 컴퓨터 상품검색을 위한 상품 지식베이스와 상품들간의 유사성(Similarity)을 정의하는 시소러스(Thesaurus)를 구성한다. 마지막으로 Template-based Reasoning 방법의 Feasibility와 Admissibility에 대한 성과 분석을 한다.

본 연구는 기업간 전자상거래에서 부품 구성기반 상품에 대한 맞춤형 제작 요구 증가에 따른 효과적인 맞춤 구성을 위한 방법론을 제안한다.

2. 문헌조사

2.1 웹 환경에서의 구매시스템

웹 환경에서 상품 검색 시스템은 일반검색(www.google.com)에서부터 비교검색, 에이전트 기반 검색 등에 이르기까지 다양하다.

구매자는 단순 상품정보를 수집하기 보다, 요구 사항에 맞는 상품정보의 동적인 제공에 관심이 있다. 구매자의 구매의사결정을 보조하는 시스템을 온라인 구매 의사 결정 지원 시스템이라 한다. 대표적인 두 가지 방법론이 있다.

Recommendation Agent(Häubl and Trifts, 2000)는 웹 기반 환경에서 구매자의 세부사항 선호도에 기반해서 구매자 중심의 맞춤 상품리스트를 작성한다. 한계점은 상품 품목 리스트가 카탈로그상의 모델들로 구성되어진다는 것과, 대안을 조정하였을 때 조정된 대안이 최선의 해를 반드시 보장하지 않는다.

Comparison Matrix(Häubl and Trifts, 2000)는 상품 데이터 베이스로부터 단순검색에 의해 대안을 선정한다. 대안은 부품별로 상품비교가 가능하다. 한계점은 단순검색으로 선정된 대안은 부품 각각에 따른 구매자 선호도를 고려하지 않는다. 또한 조정된 대안이 최적 해를 보장하지 않는다.

구성기반 상품의 경우 부품들의 단순조합으로는 부품들간의 호환성(compatibility)을 보장하기 힘들다. 맞춤구성을 위해서 구성들간의 종합성(Synthesis)과 통합성(Integrity)이 요구되어 진다.

표준검색에 유용한 대표적 상품으로는 단일 사양의 책, 비누 등을 들 수 있다. 다중사양이지만 화학제품의 경우, Options의 수가 적고, Options의 작은 변화에도 제품의 성격이 완전히 바뀔 수 있어 구성기반상품으로는 적합하지 않다. 맞춤검색의 대상은 Options의 수가 많아 다양한 상품의 조합이 발생할 수 있는 상품이 적합하다. Options의 수가 적

은 상품의 경우 비교쇼핑 등이 이용된다.

Options의 수가 많은 상품은 그 수의 증가에 따라 상품들간의 제약조건과 조합의 종류가 증가하므로, 이에 따른 구매의사결정 지원시스템이 요구되어 진다.

2.2 기업간 전자상거래와 맞춤시장의 성장

웹의 등장은 시장의 유연성을 증가시켰다. 웹 이전의 시장은 대량생산을 기본으로 하고, 구매자의 요구사항을 즉각적으로 파악하거나 시장에 반영하기 어려운 환경이었다. 따라서 맞춤 생산보다 표준상품 생산에 중점을 두었으며, 따라서 제품의 생산에 있어 유연성이 적었다. 대부분의 상품이 완제품 형태로 공장에서 출하되며, 구매자가 결정되면 완제품으로 생산을 완료한다. 즉 상품의 다양성에 대한 지원이 가능해 졌다.

웹 환경에서는 구매자 맞춤형에 기반을 둔다. 즉 구매자의 요구사항을 그 즉시 반영하는 것이 가능하다. 표준상품보다는 구성기반 상품에 대한 조합이 유용하며, 끊임없이 변하는 구매자의 요구사항에 따라 부품들의 구성이 가능하다.

현재 맞춤구성이 지원할 수 있는 전자상거래 서비스로는 Integrated Make-to-Stock, Build-to-Order 및 Configure-to-Order 등이 있다. 맞춤구성은 제품 정보에 대한 다양한 정보 수집과 종합적 구성이 가능하다. 구성기반 중심의 구매자 맞춤형 시스템은 기업간 전자상거래상의 SCM과 하이테크(High tech) 산업에서 응용되어지고 있다.

2.3 맞춤 구성 시스템

구성기반(Configuration)에 관한 연구는 인공지능 기법과 결부되진 사례가 많이 있다. 1982년 McDermott의 RI/XCON시스템은 DEC의 VAX workstation을 대상으로 한 전방향 추론(forward Chaining)을 중심으로 한 규칙기반 시스템이다. 후방향추론(Backward Chaining)과 휴리스틱 조합의 대표적인 예로는 1984년 Shortliffe와 Clancey의 MYCINE이 있다. 이외에 Well-defined Approach로 1988년의 MI/MICON와 엘리베이터 시스템 제작을 위한 VT가 있다. 1989년의 COSSACK은 기능을 주요 Component로 하고, 너비우선탐색(Best-first search)과 구매자 선호도를 지원한다.

현재 Trilogy사의 SalesPLUS는 제약조건기반 맞춤구성(Configuration) 시스템이다. 제약조건(CSP)에 기반한 시스템들은 NP-Hard 문제로 빠질 수 있는 가능성이 높다. 이 경우는 시간의 범위를 두고, 제한된 시간 안에서 최적 해를 찾는 방법을 제공한다. ILOG사는 객체지향방식의 제약조건을 이용한 맞춤구성시스템을 제공한다. 이외에 Dell과 CISCO도 웹상의 맞춤구성(Configuration)을 위한 구매시스템을 지원하고 있다.

2.4 맞춤구성(Configuration)과 인공지능 방법론

맞춤구성(Configuration)을 지원하는 인공지능 방

법론들은 다음과 같다.

사례기반추론(Case-based Reasoning)은 추론을 위한 지식으로 사례(Case)를 사용한다. 사례는 대상범위 내에서 구성들간의 제약이 강하지 않으나, 구성이 종합화 된 그 자체가 독립적 의미를 가질 때 유용하다. 한계점으로는 사례를 구성하는 요소들간의 제약관계가 명확하지 않아, 다른 형태로의 변형이 용이하지 않다. 모델기반추론(Model-based Reasoning)은 구성 요소들간의 전체적인 인과관계를 설명한다. 한계점은 모델이 가능한 모든 구체적인 경우를 설명하지 못할 수 있다는 점이다. 규칙기반추론(Rule-based Reasoning)은 요소들간의 인과관계를 규칙을 이용해 표현한다. 대상영역의 범위가 넓고, 지식의 변화가 잦을 경우, 생성되는 규칙의 수가 기하 급수적으로 증가하고, 따라서 규칙의 관리가 어려운 한계점이 있다. 제약조건(Constraint-Satisfaction Problem) 추론은 대상 영역의 전문지식을 이용해 요소들간의 인과 관계와 제약관계를 표현한다. 제약조건 추론의 한계점은 Feasible한 해를 보장하지 못한다는 것과 NP-Hard 문제라는 점이다. 제약규칙기반(Constraint and Rule Satisfaction Problem) 추론은 규칙과 제약에 기반을 한 추론 방식이며 동시진행추론을 가능하게 한다. 제약조건들간의 충돌해결(Conflict Resolution)의 한 방법으로 이용된다.

2.5 사례 및 제약조건 기반 맞춤구성(Configuration) 시스템

사례 및 제약조건 기반의 복합 연구가 진행되어 왔다. 제약조건 기반해서 상품을 제조할 경우, NP-Hard 문제에 직면하게 되는 한계가 있다. 이러한 문제점을 해결하기 위해 사례기반추론 방식을 이용 검색시간을 줄이고, 제약조건기반 방식을 이용하여 최적 해에 접근하는 방식이 연구되어 지고 있다. 대표적인 예로, 자동차 부품 구성을 위한 1995년의 COMPOSER시스템이 있다. 사례기반추론과 제약조건추론의 복합방식을 이용한다. 본 연구의 Template-based Reasoning방식은 템플릿 및 제약규칙기반의 복합 추론 방식이란 점에서 사례 및 제약조건 기반 추론과 유사하다. 차이점은 템플릿 및 제약규칙 기반 추론 방식은 대상이 특정한 영역에 응용 가능하다. 예를 들어 컴퓨터와 같은 구성기반 상품 등의 적용에 적합하다. 요소들간의 인과관계 표현이 명확하고 다양한 Options과 구성 가능한 Options 수가 다수 일 때 적합하다.

2.6 사례 조정(Adjustment) 방법론

대안 조정방법으로는 대체(Substitution), 변형(Transformation), 유도(Derivation), 합성(Composition) 등의 방법이 있다 (Wilke, Smyth, and Cunningham, 1999). 대체방식은 단순한 값의 대체를 허용한다. 변형은 일정 제약조건 하에서 값의 변화 및 요소의 접속이 가능하도록 하는 것을 의미한다. 유도는 확장 변경이 이루어 지는 것을 일컫는다. 합성방식은 다중 사례들로부터 새롭게 조정된 해를 얻는 경우를 일컫는다.

템플릿 기반 제약 규칙 기반 추론은 대체와 변

형 방식을 이용한다.

3. Template-based Reasoning과 맞춤구성(Configuration Problem)

웹 이전에는 대량생산과 이에 따른 표준상품 판매를 중심으로 한 전략이 현실적 이었다. 반면 웹의 등장은 구성기반 상품의 맞춤판매를 가능한 현실로 끌어 왔다. 현재 기업소비자간 뿐 아니라, 기업간 전자상거래에서 유연적 생산(Flexible manufacturing)에 대한 요구가 강하게 증가하고 있다. 본 연구에서 맞춤상품이란, 조합가능한 다양한 사양들이 존재하고, 각각의 항목에 따른 사양의 다중표준이 제공되는 경우를 일컫는다. 구매자가 구성 가능한 항목(options)의 수가 많고 구매자의 선호도가 반영될 수 있으며, 구성항목의 값의 변화에 따라 상품의 성격이 완전히 변하지 않을 때 맞춤상품이라 한다.

3.1 Template-based Reasoning

웹이 제공하는 기본 특징들에는 제품 정보의 동적인 수집을 가능하게 하는 상호작용성(Interactivity), 구매자의 다양한 요구사항에 능동적으로 대처할 수 있는 제품 생산의 유연성(Flexibility), 웹상에 흩어져 있는 제품 정보의 무한성(Infinite shelf-space) 등이 있다. 이러한 웹의 특성에 기반 해 구매자의 요구사항에 동적으로 대처하고 최적의 상품을 제공할 수 있는 Options에 기반 한 Template-based Reasoning 접근 방법을 제안하고자 한다.

Template은 두 종류의 Factor를 가지며, 각각 Factor들은 구체적인 값을 가진다.

Template={ {MUST}, {Options} }

MUST: 구매자가 절대적으로 원하는 구성품목 (가중치=1)

OPTIONS: 구매자의 요구사항 기반 변화가 가능한 구성품목($0 \leq \text{가중치} < 1$)

Template-based Reasoning은 두 단계로 이루어진다. 대안검색을 위한 템플릿 기반추론방식과 대안조정을 위한 제약 규칙 기반 추론 방식이다. 템플릿 기반 추론 방식은 MUST factors 기반으로 대안을 탐색한다. 이 단계는 탐색할 대상의 양을 줄여 준다. 제약 규칙 기반 추론은 Options을 이용한 대안 조정과정을 지원한다. 이 단계는 대안 조정과정을 거쳐 최선의 해에 도달할 가능성을 증가시켜 준다. 제약 규칙 기반 추론 방식은 충돌해결(Conflict Resolution)방식을 지원하며, 제약조건방식과 규칙기반 추론의 하이브리드 방식이다. 본 추론의 장점은 웹상에서 구매자와 시스템 간의 상호작용성(Interactively)을 지원한다는 점이다. 항목의 부분적인 변경도 지원한다.

본 연구에서는 대안 항목들의 조정방법으로 대체와 변형 두 가지 방법이 이용된다. 구매자가 원하는 사양과 선택된 대안의 사양이 일치하지 않고 구매자의 선호 사양이 보다 우수한 경우 구매자가 선호하는 사양으로 먼저 대체한다. 대체된 사양과

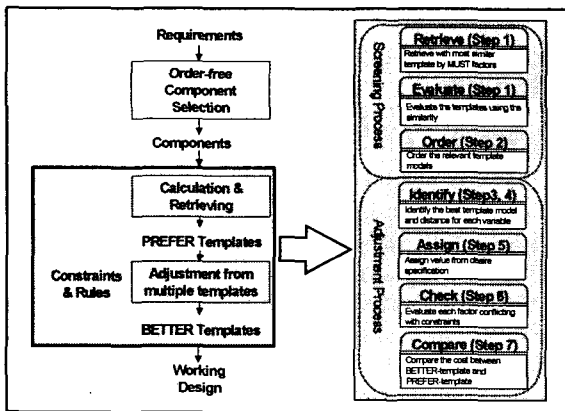
원래의 사양간의 충돌(Conflict)을 검사하고 이를 해결(Resolution)한다. 제약 규칙 기반 추론 방식의 특징은 다음과 같다. (1)구매자의 요구사항과 시스템간에 상호작용적 (Interactive)인 변형이 가능하다 (High Interactivity). (2) 부분적인 항목(Sub-part)의 변형이 가능하다 (Locally decomposition). (3) 부분적으로 대체된 사양 항목과 전체 항목들간의 호환성 (Compatibility) 검증을 지원한다. (4)위와 같은 특징들에 기반 해 추론의 속도를 빠르게 할 수 있다.

맞춤구성을 지원하는 Template-based Reasoning의 시스템 구조는 다음과 같다.

3.2 맞춤구성 시스템 구조(Configuration Problem Architecture)

다음 [그림. 1]은 Template-based Reasoning에 의한 맞춤상품 구매 의사결정 지원 시스템의 구조 (Architecture) 이다.

[그림. 1] 맞춤구성 시스템 구조



검색과정에서의 상태는 다음과 같이 나타낸다.

- OPEN: 선정된 대안들
- CURRENCY: 현재 최고의 대안
- ACHIEVED: 현재까지의 최고의 대안
- CLOSED: 도태된 대안들

Template-based Reasoning 프로세스는 대안검색과 대안조정 의 두 단계로 이루어진다.

대안검색 과정(Screening Process)은 다음과 같다.

- STEP1: 대안 검색 및 선택된 대안의 유사성 검증 (Retrieve & Evaluate)
- STEP2: MUST factors에 의해 선정된 대안들 정렬 (Order)

대안선택과정에서 선정된 여러 개의 대안들로부터 대안조정 과정(Adjustment Process)은 다음과 같다.

- STEP3: 현재 최고의 해를 CURRENCY로 정한다. (Identify CURRENCY)
- STEP4: 구매자가 요구하는 각 Factor의 사양과 선정된 템플릿의 사양간의 차이를 정의한다. (Identify Distance)

STEP5: 구매자의 요구하는 각 Factor의 사양에 맞도록 템플릿의 사양을 조정한다.(Assign & Check)

STEP6: STEP5에서 조정된 템플릿이 구매자의 요구사항과 일치하면 검색을 마친다. 그렇지 않으면, 차선의 해를 CURRENCY로 하고 STEP 3으로 Backtracking한다.

STEP7: 최선의 대안과 차선의 대안의 조정 값을 비교하여 비교우위에 있는 대안을 ACHIEVED로 한다. 선택되지 못한 대안은 CLOSED로 한다. (Compare)

본 논문에서 사용되는 표기법은 다음과 같다.

D : 구매자가 선호하는 Factor들의 값, $D=[d_j], \forall j$

d_j : j 번째 Factor 값

$T=[T(i)], \forall i$: 템플릿 집합

$T(i)$: Template i

$T(i)=[t(i)_j], \forall j$

$t(i)_j$: Template i 의 j 번째 사양의 값

$A(i)$: $T(i)$ 의 조정 Template

$O(i)$: $T(i)$ 의 조정된 Options 사양, $O=[o_j], \forall j$

$o(i)_j$: Template $O(i)$ 의 j 번째 사양

$R(i)=T(i)-D$: Template $T(i)$ 와 D 사이의 차이

$R(i)=[r(i)_j], \forall j$

$CT(i)$: Template $T(i)$ 의 가격, $CT(i)=\sum_j = \text{Cost of } t(i)_j$

$CO(i)$: Options의 조정에 따른 변동 가격

$CA(i)$: 조정된 Template의 가격, $CA(i)=CT(i)+CO(i)$

구매자의 제품 선호도는 다음과 같이 표현된다.

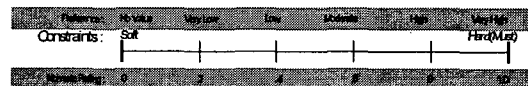
$$D=[d_1, d_2, \dots, d_j, \dots, d_n], 0 \leq d_j \leq 1$$

각 선호도의 가중치는 다음과 같다.

$$W=[w_1, w_2, \dots, w_j, \dots, w_n],$$

$$w_j=\{w_j | w_j=0.2x, x \in \{0, 1, 2, 3, 4, 5\}, 0 \leq w_j \leq 1\}$$

[그림. 2] 구매자 선호도 (Preference)



구매자 선호·제품과 대안들간의 유사성 (Similarity)은 Nearest-Neighbor Ranking 알고리즘을 사용한다. Nearest-Neighbor Ranking 알고리즘은 가장 근접한 해를 찾는데, 비교대상의 값과 그 각각의 가중치를 고려하여 유사성을 측정한다.

<표 1> Nearest Neighbor Algorithm에 의한 유사성

Numeric Evaluation Function	: $sim(d_j, t_j(i))$ for all i, j
	sim : similarity function for primitives
	d_j : the j^{th} slot value for feature D in the input requirement
	$t_j(i)$: the j^{th} slot value for feature T in the retrieved templates
Aggregate Match Score	: $\sum (w_j \times sim(d_j, t_j(i))) / \sum w_j$
	w_j : the preference of dimension (slot), $0 \leq w_j \leq 1$

대안조정은 대안 선정에서 도출된 값에 근거하여 이루어진다. 구매자의 각 구성항목(Factor)에 대한 요구사항과 선정된 대안간에 값의 차이 (Distance)는 다음과 같이 표현된다.

$r(i)_j = t(i)_j - d_j, -1 \leq r(i)_j \leq 1, \forall d_j, \forall t(i)_j$
 값의 차이는 방향성에 의해 다음과 같이 정의된다.

$t(i)_j - d_j > 0$ if and only if Over Qualify
 $t(i)_j - d_j < 0$ if and only if Under Qualify

Over Qualify는 Template에서 선정된 Factor의 사양이 구매자의 요구사항보다 우수한 경우로, Template의 사양을 그대로 유지한다. Under Qualify는 구매자의 요구사항에 비해 Template에서 선정된 Factor의 사양이 열등한 경우로 선정된 Template의 해당 Factor의 사양을 구매자의 요구사항에 맞도록 변경한다.

본 연구에서 Search Beam Width는 1로 한다. 즉 현재의 최고의 Template으로부터 최선의 해를 찾기 위한 탐색을 한다.

맞춤상품검색 알고리즘은 다음과 같다.

<표 2> 맞춤검색 알고리즘

Step1. Template based Reasoning	
1.1 Set the cut-off distance	Use the users for the identification of criteria Apply the pre-emptive screening criteria by MUST factors
1.2 Evaluate the templates using the similarity	
Step2. Order the relevant template models	
Number the templates as $T(1)$ according to the order of satisfaction. Assume the Beam width = $1, T_1(T(1)), T_2(T(2)), \dots, T_n(T(n))$ is the template set, $C(T)$ - Cost of T	
Step3. Identify the currency i to most preferred template model	
Initially, set iteration $\leftarrow 1$ If $D = T(1)$, Go to Step 7; The desired model is found.	
Step4. Identify the difference between the desire and template by value distance for each variable $D - T(i)$; Identify the non-zero variables.	
Step5. Seek the desired specification by adding the required optional specification, $A(i) = D - T(i)$ Search scheme for adjustment.	
5.1 Variable Ordering: Non-zero factor of the unsatisfied variables is ordered by importance criteria	
5.2 Adjust toward the optional specification enhancement $C(i)$ by the importance criteria, and find extra cost, $CC(i)$	
5.2.1 Consider the constraints for the optional specifications (CRSP), $A(i) = T(i) + C(i)$.	
5.2.2 Compute the total cost, $CA(i) = C(T(i)) + CC(i)$.	
5.3 If the adjustment cannot meet the desired specification ($D > A(i)$), keep the currently reachable specification with the $A(i)$ and its cost, $CA(i)$.	
Step6. If iteration=1, Repeat step 3-5 with next similar.	
Go to step 7	
Step7. Compare $CA(i)$ with $CA(j)$ If $CA(i) < CA(j)$, Go to step 4 If $CA(i) > CA(j)$, reset the currency to next similar, $A(j)$ and repeat step 3-5	

4. 컴퓨터 구매를 위한 맞춤상품검색

컴퓨터시스템을 이용한 구성기반 맞춤 상품 검색 예는 다음과 같다. 컴퓨터 시스템은 다양한 Option들로 이루어져 조합 가능하다. 따라서 단순 검색만으로는 구매자의 요구조건에 맞는 차별화된 상품을 검색하기란 쉽지 않다. 또한 표준화된 상품만으로는 구매자의 다양한 요구사항에 능동적으로 대처하기 어렵다.

본 연구에서는 가능한 최소한의 노력으로 구매자의 요구사항에 부합하는 맞춤상품 검색을 목표로 한다. 구매자가 정한 MUST 사양이 대안 선정의 제약이 된다. 즉 MUST 사양을 만족하는 Template들을 대안으로 선정하고, 이들 중 최선의 해를 찾는 것을 목적으로 한다.

본 연구에서 사용된 컴퓨터에 대한 구매자의 사양에 대한 세부적 요구사항은 다음과 같다.

<표 3> 컴퓨터 사양 및 구매자번호

Factor	Spec	Desired	Weight
CPU	Processor	Pentium III	1.0
	Speed (MHz)	650	1.0
	Speed (MHz)	300	1.0
Hard Drive	RAM	32MB	1.0
	Form	3.5"	1.0
	File	ATA	1.0
Monitor	Size (inches)	15"	1.0
	Resolution	1024x768	1.0
	Type	Monitor	1.0
Video Card	Size (MB)	32	0.4
	AGP Speed	4x	0.4
CD/DVD	Type	COM-ATI Radeon	0.4
	Speed	52x	0.4
Sound Card	Type	Sound Blaster 16V PCI	0.4
	Type	NEW Harmon Kardon HK-505 Surround Sound	0.4
Speakers	Type	No Modem Requested	0.0
Modem	Type	SCODE (V.90 Fax) Ethernet PC Card	0.8
Network Card	Type	MicroSOFT Network Keyboard Pro	0.2
Keyboard	Type	M.B. IntelliMouse	0.2
Mouse	Type	New Microsoft TM Works Suite 2001 with Money 2001	0.4
Bundled Software	Type	1 yr. Next Business Day On-Site PART# 44399, Yrs. 2-3 Parts	0.4
Limited warranty and support	Type		0.4

4.1 대안탐색

대안탐색은 STEP1과 STEP2로 이루어진다.

STEP1: Template-based Reasoning

요구사항과 Template의 사양간에 유사성을 정의한다. MUST(Weight=1) 사양을 우선적으로 고려하여, 대안을 선정한다. 유사성의 정도는 Nearest-Neighbor 알고리즘에 의해 계산된다.

OPEN={T(8), T(9), T(10)}

CURRENCY=[]

ACHIEVED=[]

CLOSED=[]

STEP2: MUST에 의해 선정된 대안의 정렬 만족도가 높은 순으로 선정된 대안을 정렬한다.

OPEN={T(9), T(8), T(10)}

CURRENCY=[]

ACHIEVED=[]

CLOSED=[]

Search beam width는 1로 한다. 즉 현재 만족도 값이 가장 큰 템플릿으로부터 최적의 대안을 찾는다. 본 실험에서는 T[9]이 PREFER-Template이다

<표 4> 대안으로 선정된 템플릿(T(8), T(9), T(10))

Factor	Spec	T(8)	T(9)	T(10)
CPU	Processor	1.0	1.0	1.0
	Hz (M)	1.0	1.0	1.0
Memory	Size (MB)	1.0	1.0	1.0
	Speed (MHz)	1.0	1.0	1.0
Hard Drive	Size (GB)	1.0	1.0	1.0
	rpm	1.0	1.0	1.0
Monitor	size (")	1.0	1.0	1.0
	Refresh rate (KHz)	1.0	1.0	1.0
	Resolution	1.0	1.0	1.0
Norm Match score for MUST		1.0	1.0	1.0
	Factors	Spec	T(8)	T(9)
Video Card	Size (MB)	0.0	1.0	1.0
	AGP Speed	0.0	1.0	1.0
CD/DVD	Type	0.2	0.5	0.5
	Speed	0.2	0.2	0.2
Sound Card	Type	0.8	0.8	0.8
	Type	1.0	0.2	0.2
Modem	Type	0.0	0.0	0.0
	Type	0.0	0.0	0.0
Network Card	Type	0.8	0.6	0.6
	Type	0.8	0.6	0.6
Mouse	Type	0.8	1.0	1.0
	Type	1.0	1.0	0.2
Bundled Software	Type	1.0	1.0	0.2
	Type	1.0	1.0	0.2
Norm Match Score for all		0.700	0.700	0.700
Cost		\$1,259.00	\$1,239.00	\$1,599.00

4.2 대안조정

대안조정은 STEP3에서 STEP7로 이루어진다.

STEP3: 현재 최고의 해를 CURRENCY로 정한다. T(9)이 CURRENCY가 된다.

OPEN=[T(8), T(10)]
 CURRENCY=[T(9)]
 ACHIEVED=[]
 CLOSED=[]

T(9)이 구매자의 요구사항을 모두 만족하면, 검색을 완료한다. 그렇지 않으면, Options을 조정하면서 최선의 해를 찾는다.

STEP 4: 구매자의 요구사항의 각 Factor의 사양에 대응하는 T(9)의 사양을 찾고, 각각의 차이를 정의한다. Over Qualify ($t(i)_j - d_j > 0$) 와 Under Qualify ($t(i)_j - d_j < 0$) 값을 가지는 사양들을 모두 정의한다.

<표 5> 대안과 구매자 요구사항 간의 차이

Factors	Distance of T(9)
VideoCard Size	0.0
VideoCard AGP Speed	0.0
VideoCard Type	-0.4
CD/DVD Speed	0.4
CD/DVD Type	-0.8
Sound Card	0.2
Speakers	-0.8
Modem	1.0
Network Card	-1.0
Keyboard	-0.4
Mouse	0.0
Bundled Software	0.0
Limited warranty and support	0.0

STEP 5: 구매자의 요구사항의 각 Factor 사양에 맞도록 T(9)의 사양을 조정한다.

STEP 5.1: Over Qualify와 Under Qualify된 변수들을 각각 내림차순으로 정렬한다.

<표 6> Over Qualify와 Under Qualify

Factors	Distance of T(9)	Preference
CD/DVD Speed	0.4	0.4
Sound Card	0.2	0.4
Modem	1.0	0.8
Network Card	-1.0	0.8
Speakers	-0.8	0.4
CD/DVD Type	-0.8	0.4
VideoCard Type	-0.4	0.4
Keyboard	-0.4	0.2

$t(i)_j - d_j > 0$: Over qualify

$t(i)_j - d_j < 0$: Under qualify

STEP 5.2: Options조정은 단순대체(Assign)와 변수들간의 제약조건을 검사하여 호환성을 조사(Check)로 이루어진다.

STEP 5.2.1: 단순대체(Assign)에 의한 Options조정 Over Qualify된 Factor는 Template의 값을 이용하고, 대체 연산자(Operator)에 의해 Under Qualify된 사양에 대해서만, 구매자의 요구사항으로 각 Factor의 값을 대체한다. 각 항목은 다시 세부항목으로 이루어질 수 있는데, 이때 세부 항목 중 하나라도 Under Qualify되면 Under Quality항목으로 한다. 조정된 사양에 근거하여 변화된 가격을 계산한다. T(9)의 가격인 CT(9)이 \$1,299.00이고, Options 조정된 가격의 총합 CO(9)은 \$348.00이다. 이 둘을 합한 CA(9)의 가격의 총합은 \$1,647.00이다.

<표 7> Under Qualify 예제

Factors		Distance of T(9)
CD/DVD	Speed	0.4
	Type	-0.8

Over qualify
Under qualify

<표 8> T(9)의 보정 된 O(9)

Options	Preference	Price	Distance	O(9)
Network Card	3COM 1615R Fast Ethernet PC Card	No network card	-1.0	\$49.00
Speakers	NEW Herman Jordan HC-855 Surround Sound	Herman Jordan Speakers	-0.8	\$79.00
CD/DVD	8X/4X/2X CD-RW Drive	8X Max Variable CD-ROM Drive	-0.2	\$59.00
Video Card	New 32MB DDR ATI Radeon Card	32MB NVIDIA GeForce2 MX 4X AGP Graphics Card	-1.1	\$59.00
Keyboard	Microsoft Natural Keyboard Pro	QuickKey Keyboard	-0.4	\$49.00
Price				\$348.00

STEP 5.2.2: Options조정에 의한 변수들간의 제약 조건을 제약 규칙 기반 추론과정에 의해 호환성을 검사한다. 각각의 항목들간의 제약관계는 관계테이블 <표 9>에서 정의된다.

Positive Relationship: 1 (변화된 사양과 영향을 받는 사양간의 변화의 방향이 같은 경우)

Negative Relationship: -1 (변화된 사양과 영향을 받는 사양간의 변화의 방향이 반대인 경우)

Independent Relationship: 0 (두 사양간의 관계가 없는 경우)

T(9)의 항목들간의 제약관계 테이블은 다음과 같이 정의된다.

<표 9> 제약관계 테이블

Independent factors	Dependent factors															
	CPU	Memory	Hard Drive	Video Card	AGP	Sound Card	Speakers	Modem	Network Card	Keyboard	Mouse	Bundled Software	Limited warranty and support	CD/DVD	Video Card	
CPU	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Memory	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Hard Drive	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Video Card	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
AGP	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Sound Card	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Speakers	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Modem	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Network Card	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Keyboard	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Mouse	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bundled Software	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Limited warranty and support	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CD/DVD	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Video Card	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

충돌 해결을 위한 제약 규칙기반 추론(CRSP)의 프로시저(이재규, 권순범, 1993)는 다음과 같다.

<표 10> CRSP Procedure

<ol style="list-style-type: none"> 1. Goal setting 2. Variable ordering according to the preference value 3. Seed variables selection 4. For each seed, <ul style="list-style-type: none"> Repeat from the current variable, Rule Inference or Constraint propagation If inconsistency or dead-end happen, goto step5 else Move to the next variable. Until all variables are exhaustively propagated. Remove currency from the Constraint and Rule graph with a seed variable If propagation of all the other consistent Constraint and Rule graph with a seed variable are completed then end with a solution else goto the first of step4 5. Perform backtracking <ul style="list-style-type: none"> If backtracking fail to resolve the inconsistency or dead-end then goto step6 else move to the next variable go back to step 4. 6. Perform user answer adjustment and trade-off negotiation process. <ul style="list-style-type: none"> If the inconsistency is still not resolved then end with no solution else goto Step4.
--

대안 조정을 위한 CRSP알고리즘은 다음과 같다.

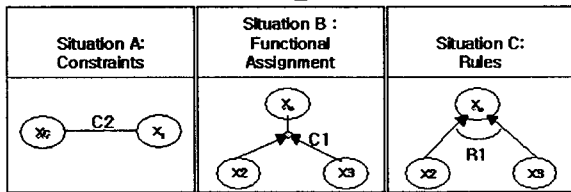
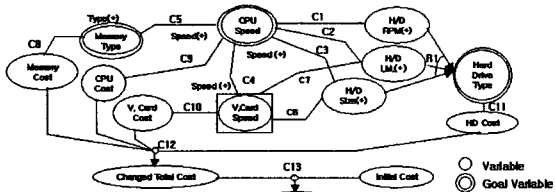
- Goal Setting: Set MUST factors as goal variables and values
CPU: 933 Mhz

Hard Drive: 40GB, 7200RPM, ATA
Memory Type: SDRAM

- Variable Ordering: 선호도 값에 따른 변수정렬
- Seed variable: 조정이 요구되어지는 변수의 값을 Seed variable로 한다.
Video Card Speed
- 동시 추론(Concurrent Reasoning): 다중 Seed Variable을 이용한 동시 추론
- Backtracking: 충돌 해결(Conflict Resolution)

제약 조건들 간의 호환성 그래프(Compatibility Constraint Graph)는 다음과 같다.

[그림. 3] CRSP 호환성 그래프



제약 조건과 규칙 (Compatibility Constraints and Rules)들은 다음 표와 같다.

<표 11> 제약조건 및 규칙의 호환성

Variables	Functional Assignment (C1-C13)	Rules (R1)
Hard Drive Selection Method CPU Speed Hard Drive Type Hard Drive Size Hard Drive RPM Hard Drive L.M. Hard Drive Selection Method Video Speed Memory Type Changed Cost of CPU Changed Cost of Video Type Changed Cost of Memory Changed Cost of Monitor Changed Total Cost Total Price	C1. Compatibility of CPU speed & HD Size C2. Compatibility of CPU speed & HD RPM C3. Compatibility of CPU speed & HD L.M. C4. Compatibility of CPU speed & Video Speed C5. Compatibility of CPU speed & Memory Type C6. Compatibility of CPU speed & Monitor C7. Compatibility of CPU speed & Video Card Type C8. Compatibility of CPU speed & Video Card Speed C9. Compatibility of CPU speed & Video Card Price C10. Compatibility of CPU speed & Video Card Cost C11. Compatibility of CPU speed & Video Card Price C12. Compatibility of CPU speed & Video Card Cost C13. Compatibility of CPU speed & Video Card Price	R1. Rule 1

충돌 해결(Conflict Resolution)을 위한 연산자(Operators)는 3가지로 종류로 분류한다.
Operator1 (Desire): Over Qualified 값은 구매자의 선호 값으로 바꾼다.
Operator2 (Criteria): Operator1에 의해 해를 구할 수 없으면, 제약조건이 약한 순서로 사양의 값을 변경한다.
Operator3 (ASK): Operator1, 2에 의해 해를 구할 수 없으면, 구매자가 해를 선택하도록 한다. 이 경우는 권장사항은 아니다.

<표 12> 충돌해결을 위한 동시추론

Conflict	Resolution
Conflict 1: CPU Speed & HD Size	Conflict Resolution Needed
Conflict 2: CPU Speed & HD RPM	Weak constraints value(options) changed for conflict resolution

STEP 5.2.3: Options조정후의 가격계산
T(9)의 대안의 대체조정 이후 (STEP5.2.1)의 가격은 \$1,647.00이고, CRSP과정에서 \$90.00가 보정되어 보정된 최종가격인 CA(9)은 \$1,557.00이다.

- CPU Speed: 933Mhz
- HD RPM: 7200
- HD L.M.: Ultra ATA
- HD Size: 40G
- Video Card Type: 32MB NVIDIA GeForce2 MX 4X AGP Graphics Card
- Memory Type: SDRAM
- Changed Cost of CPU: \$0
- Changed Cost of HD: \$0
- Changed Cost of Video Type: -\$90
- Changed Cost of Monitor: \$0
- Changed Cost of Memory: \$0
- Changed Total Cost: Current Value = -\$90
- Price: \$1,647
- Total price: \$1,557

<표 13> 가격의 총합

Category	Item	Quantity	Unit Price	Total Price
CPU	Processor	1	\$933	\$933
	Cache	1	\$100	\$100
	Chipset	1	\$100	\$100
	Motherboard	1	\$100	\$100
	RAM	1	\$100	\$100
	Power Supply	1	\$100	\$100
	Cooling Fan	1	\$100	\$100
	Case	1	\$100	\$100
	Keyboard	1	\$100	\$100
	Mouse	1	\$100	\$100
HD	Hard Drive	1	\$100	\$100
	Controller	1	\$100	\$100
	Adapter	1	\$100	\$100
	Power Supply	1	\$100	\$100
	Cooling Fan	1	\$100	\$100
	Case	1	\$100	\$100
	Keyboard	1	\$100	\$100
	Mouse	1	\$100	\$100
	Monitor	1	\$100	\$100
	System Unit	1	\$100	\$100

STEP 5.3: 현재 도달 가능한 최적해
현재 도달된 최적 해가 구매자의 요구사항을 만족하지 않으면, 그 해(A(9))와 가격(CA(9))을 유지하면서, STEP6으로 진행한다. 현재 조정된 최종 가격(CA(9))은 \$1,557.00이다.

STEP 6: STEP3으로 Backtracking
현재 최적인 CURRENCY가 T(9)이고, 비교 대상이 없으면, STEP2로부터 차선의 해 T(8)을 가지고 STEP 3-7을 반복한다.
OPEN=[T(8), T(10)]
CURRENCY=[T(9)]
ACHIEVED=[]
CLOSED=[]
T(9)과 비교대상이 있으면 STEP7로 이동한다.

STEP7: T(9)과 T(8)의 조정된 해를 비교한다.
보정된 가격인 CA(8)이 CA(9)보다 작으면, T(10)을 CURRENCY로 한다. STEP 3으로 Backtracking한다

OPEN=[]
 CURRENCY=[T(10)]
 ACHIEVED=[T(8)]
 CLOSED=[T(9)]

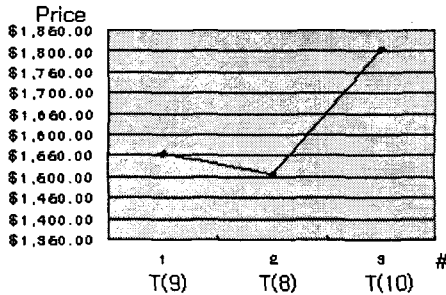
4.3 The Successive Stage

다음 표는 T(8)과 T(10)의 Backtracking과정을 보여준다. 본 예제에서는 T(8)이 최선 해로 선정되었다.

<표 14> 대안조정 예

	Reset the currency to T(9) and repeat step 3-5	Reset the currency to T(8) and repeat step 3-7	Reset the currency to T(10) and repeat step 3-7
List of states	OPEN={T(9), T(10)} CURRENCY={T(9)} ACHIEVED={} CLOSED={}	OPEN={T(10)} CURRENCY={T(8)} ACHIEVED={T(9)} CLOSED={}	OPEN={} CURRENCY={T(10)} ACHIEVED={T(8)} CLOSED={T(9)}
Cost of the currency	CA(9)=\$1,557	CA(8)=\$1,507	CA(10)=\$1,807
Solution	A(9)	A(8)	A(8)
Stopping Condition	If iterations=1 Repeat step 3-5 with next similar	If CA(9)>CA(8), reset the currency to next similar, A(8) and repeat step 3-5	If CA(8)>CA(10), goto stop

[그림. 4] 가격에 따른 대안의 선정 T(8)

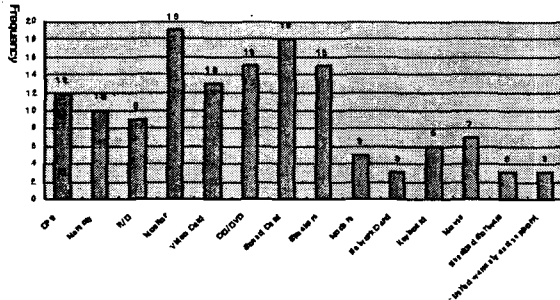


5. 성능평가

본 연구에서는 Options의 수가 증가할수록 구매자의 요구사항에 맞는 해를 찾기 위한 노력이 증가됨을 보이고, 이러한 노력을 줄이기 위한 효율적인 알고리즘을 제안했다. Options의 수에 따른 알고리즘의 효율성을 평가한다.

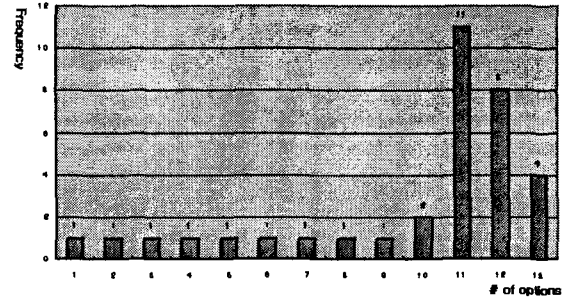
실험에 사용된 사례의 개수는 34개이며, 각 실험에서 고려되어진 사안의 수는 각각 14개이다. 사용된 평균 Options의 평균 수는 11개이고, MUST factor의 평균은 3개이다. MUST factor로 사용된 factor들의 빈도수는 다음과 같다.

[그림. 5] 사용된 MUST factors 분포도



사례에서 선택된 Option의 개수의 빈도수는 다음과 같다.

[그림. 6] 가격에 따른 대안의 선정 T(8)



5.1 맞춤검색 알고리즘의 효과성

구성기반 상품의 경우 단순검색에 의해 선정된 대안들 중, 최고의 해를 선택하여 조정과정을 통해 최선의 해를 구할 수 있는지에 대해 평가한다. 평가결과는 다음과 같다.

<표 15> 최적해 와 Backtracking 횟수

# of options	1st iteration	2nd iteration	3rd iteration	Current Optima
10	\$1,557.00	\$1,467.00	\$1,807.00	\$1,467.00
11	\$1,794.00	\$1,959.00		\$1,794.00
11	\$1,818.00	\$2,207.00	\$2,008.00	\$1,818.00
11	\$2,088.00	\$2,298.00		\$2,088.00
12	\$2,208.00	\$2,108.00	\$2,318.00	\$2,108.00
11	\$2,178.00	\$2,228.00	\$2,058.00	\$2,058.00
11	\$1,828.00	\$2,028.00		\$1,828.00
11	\$1,558.00	\$1,708.00	\$1,498.00	\$1,498.00
11	\$1,098.00	\$1,178.00	\$1,078.00	\$1,098.00
12	\$1,668.00	\$1,228.00	\$1,808.00	\$1,228.00
12	\$1,828.00	\$1,498.00	\$2,318.00	\$1,498.00
11	\$1,818.00	\$1,818.00	\$1,808.00	\$1,818.00
11	\$1,427.00	\$1,367.00	\$1,387.00	\$1,367.00
8	\$1,128.00	\$1,158.00	\$1,128.00	\$1,128.00
11	\$1,428.00	\$1,678.00	\$1,478.00	\$1,428.00
13	\$1,888.00	\$1,738.00	\$1,318.00	\$1,318.00
12	\$1,338.00	\$1,448.00	\$1,128.00	\$1,128.00
12	\$1,818.00	\$1,848.00	\$1,808.00	\$1,818.00
11	\$1,808.00	\$1,468.00	\$2,298.00	\$1,468.00
10	\$2,688.00	\$2,108.00	\$2,788.00	\$2,108.00
13	\$2,458.00	\$2,258.00	\$2,088.00	\$2,088.00
6	\$1,808.00	\$1,758.00	\$2,158.00	\$1,808.00
12	\$1,254.00	\$1,204.00	\$1,888.00	\$1,204.00
12	\$1,228.00	\$1,208.00		\$1,208.00
12	\$1,418.00	\$1,408.00	\$1,198.00	\$1,198.00
13	\$2,728.00	\$2,628.00	\$2,608.00	\$2,628.00
1	\$1,818.00	\$1,818.00		\$1,818.00
8	\$1,388.00	\$1,798.00		\$1,388.00
13	\$2,378.00	\$2,228.00	\$2,827.00	\$2,228.00
7	\$2,258.00	\$2,448.00		\$2,258.00
4	\$1,278.00	\$1,818.00		\$1,278.00
3	\$1,738.00	\$1,738.00		\$1,738.00
2	\$1,884.00	\$1,818.00		\$1,884.00
5	\$1,308.00	\$1,253.00		\$1,253.00
Error	7.18%	3.21%	0.00%	0.00%

실험결과 34개의 사례 모두 최적 해에 도달하였다. 19개의 사례는 2-3번의 Backtracking과정을 통해 최적의 해에 도달하였다. 구성기반상품으로 컴퓨터를 대상으로 하였으며, Backtracking 3번안에 모두 최

적 해에 도달하였다. 따라서 단순 검색에 의해 선정된 최적 해가 보정 이후에 최적 해를 보장하지 않음이 확인됐다. 상품에 따라 Options 수에 따라 Backtracking의 횟수는 변할 수 있다.

단순검색에 의해 선택된 최선의 해와 보정된 이후의 최적해 사이에 값의 차이는 P값 ($T=42.712 > 3.609 = t_{(33, 0.0005)}$) 0.0005의 범위에서 유의하다. 두 해 사이의 평균오류는 7.18%이다.

<표 16> 단순검색과 맞춤검색 최적해 비교

# of options	1st Iteration	Current Optimum	Offset %	Error
10	\$1,557.00	\$1,467.00	\$ 90.00	6.15%
11	\$1,744.00	\$1,744.00	\$ 0.00	0.00%
11	\$1,918.00	\$1,918.00	\$ 0.00	0.00%
11	\$2,088.00	\$2,088.00	\$ 0.00	0.00%
12	\$2,208.00	\$2,108.00	\$100.00	4.74%
11	\$2,178.00	\$2,058.00	\$120.00	5.85%
11	\$1,928.00	\$1,928.00	\$ 0.00	0.00%
11	\$1,558.00	\$1,498.00	\$ 60.00	4.01%
11	\$1,068.00	\$1,068.00	\$ 0.00	0.00%
12	\$1,668.00	\$1,228.00	\$440.00	35.85%
12	\$1,928.00	\$1,488.00	\$440.00	28.70%
11	\$1,918.00	\$1,818.00	\$100.00	6.60%
11	\$1,427.00	\$1,367.00	\$ 60.00	4.39%
8	\$1,129.00	\$1,129.00	\$ 0.00	0.00%
11	\$1,429.00	\$1,429.00	\$ 0.00	0.00%
13	\$1,889.00	\$1,819.00	\$670.00	48.21%
12	\$1,389.00	\$1,129.00	\$210.00	18.80%
12	\$1,819.00	\$1,819.00	\$ 0.00	0.00%
11	\$1,509.00	\$1,489.00	\$ 20.00	2.72%
10	\$2,589.00	\$2,109.00	\$480.00	22.78%
13	\$2,489.00	\$2,089.00	\$370.00	17.71%
6	\$1,609.00	\$1,609.00	\$ 0.00	0.00%
12	\$1,284.00	\$1,204.00	\$ 80.00	4.16%
12	\$1,229.00	\$1,209.00	\$ 20.00	1.65%
12	\$1,419.00	\$1,199.00	\$229.00	19.24%
13	\$2,729.00	\$2,629.00	\$ 0.00	7.71%
1	\$1,549.00	\$1,549.00	\$ 0.00	0.00%
9	\$1,599.00	\$1,599.00	\$ 0.00	0.00%
13	\$2,378.00	\$2,228.00	\$150.00	6.73%
7	\$2,268.00	\$2,268.00	\$ 0.00	0.00%
4	\$1,279.00	\$1,279.00	\$ 0.00	0.00%
3	\$1,738.00	\$1,738.00	\$ 0.00	0.00%
2	\$1,694.00	\$1,694.00	\$ 0.00	0.00%
6	\$1,509.00	\$1,268.00	\$ 241.00	4.47%
		\$1,051.6		7.18%

본 연구의 해 검색가능성(Feasibility)과 최적해 도달 가능성(Admissibility)에 관한 효과성 측정결과는 다음과 같다.

<표 17> 최적해 검색을 위한 반복 수

STEP	반복 수	Feasibility	Error Ratio (%)	Admissibility	Error Ratio (%)
단순 검색	1	No Guarantee	100.00 (88.24%)	No Guarantee	100.00 (11.20%)
	1	No Guarantee	20.00 (17.655%)	No Guarantee	69.01 (7.73%)
호환성 보정	1	Guarantee	0.00 (0.00%)	No Guarantee	64.13 (7.18%)
	2	Guarantee	0.00 (0.00%)	No Guarantee	28.64 (3.21%)
	3	Guarantee	0.00 (0.00%)	Guarantee	0.00 (0.00%)

단순검색으로 찾아진 해가 최적을 보장되지 않

는다. 단순검색으로 찾아진 해는 보정 과정과 호환성 조사 및 Backtracking에 의해 Feasibility와 Admissibility를 보장 받는다.

5.2 Options 효과

최적 해에 도달하기 위해 Options 수의 증가와 Backtracking의 횟수 간의 관계를 조사한다. 실험결과는 다음과 같다.

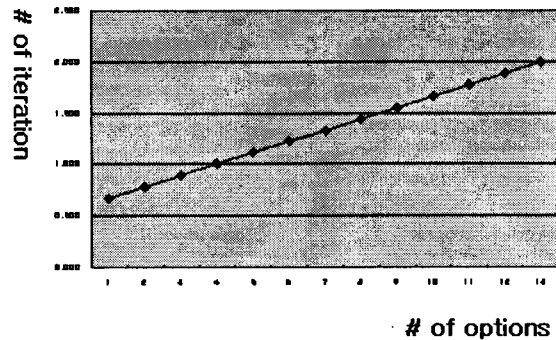
<표 18> OPTIONS과 Backtracking의 관계

# of Iteration	1	2	3	4	5	6	7	8	9	10	11	12	13	# of Cases
1	1	1	1	1	0	1	1	1	1	0	6	1	0	15
2	0	0	0	0	1	0	0	0	0	2	3	5	2	15
3	0	0	0	0	0	0	0	0	0	0	2	1	3	9

Options의 수가 증가됨에 따라 최적 해에 도달하기 위한 Backtracking의 횟수가 증가됨을 보였다. 실험결과에 따른 선형식은 다음과 같다.

$$Y = 0.567 + 0.110X, Y: \# \text{ of Iteration}, X: \# \text{ of Options}$$

[그림. 7] Options 효과



Options효과 실험에 따른 기울기에 대한 T-테스트는 결과는 다음과 같다. $H_0: \beta \leq 0, H_a: \beta > 0, n=34, p\text{값}(T=3.262 > 2.455 = t_{(32, 0.01)})$ 0.01에서 H_0 는 기각된다.

Options효과 실험에 따른 ANOVA분석 결과는 다음과 같다.

<표 19> Options과 Backtracking의 선형관계

Source	SS	df	MS=SS/df	F
Regression	4.65	1.00	4.65	10.64
Error	13.97	32.00	0.44	
Total	18.62	33.00		

$H_0: \beta = 0, H_a: \beta \neq 0, F\text{값}(10.64 > 5.31 = F_{(2, 32, 0.01)})$ 은 0.01에서 H_0 를 기각한다. 맞춤구성 Options의 수가 증가할수록 Backtracking의 횟수도 증가한다.

Options 효과에서의 Backtracking 그룹간 차이에 대한 ANOVA 결과는 다음과 같다. $H_0: \tau_1 = \tau_2 = \dots = \tau_n, F\text{값}(6.83 > 5.38 = F_{(2, 31, 0.01)})$ 은 0.01에서 유의하다. 즉 Backtracking 그룹간에는 유의한 차이가 있다.

<표 20> Backtracking 그룹간의 ANOVA 분석

Between sample	110.24	2.00	55.12	6.83
Within Sample	250.10	31.00	8.07	
Total	360.33	33.00		

두 표본간의 T-테스트 결과는 다음과 같다.
 $H_0: \mu_m - \mu_s = 0, H_a: \mu_m - \mu_s > 0, \mu_m = 19, \mu_s = 15, n = 34$
 μ_s : Backtracking이 없는 표본의 평균 Options 수
 μ_m : Backtracking이 2번 이상인 표본의 평균 Options 수
 p 값($T=1.893 > 1.695 = t_{(32, 0.05)}$)이 0.05에서 두 표본간의 차이가 유의함을 보인다.

<표 21> Backtracking 그룹

Number of iterations	Number of iteration	Number of iteration
11	10	11
11	12	11
11	12	13
11	12	12
11	11	13
8	11	12
11	11	
11	10	
6	12	
1	12	
9	13	
7	13	
4	5	
3		
2		

6. 결론 및 향후 연구방향

본 연구에서는 기존의 단순검색이 구성기반 맞춤검색에 적합하지 않음을 보이고, 맞춤 상품 검색을 위한 알고리즘을 제시하였다.

맞춤 구성의 수가 증가할수록 단순검색으로는 구매자의 요구사항에 부합하는 최적 해를 찾을 수 없으며, 사양의 조정에 의해 구매자의 요구에 부합하는 사양을 찾더라도, 가격에 있어 최선의 해가 아닌 경우가 발생한다. 이러한 문제점을 해결하기 위해 제약 규칙 기반 추론 알고리즘에 기반 한 맞춤상품 검색 알고리즘을 제안했다.

본 연구는 기업간 전자상거래에서 경쟁력 있는 차별화 된 상품을 구매자에게 제공하는 데 이용될 수 있으며, 단일 판매자(Vendor)뿐 아니라, 다중 판매자(Vendor)간의 구성부품조립에서 사용되어 질 수 있다. 다중 판매자 구성 조합이 이루어 지기 위해서는 에이전트 기반 검색기술, XML/XRML기반 Data Processing 및 사양의 단일화 등이 요구되어진다.

[참고문헌]

[1] Alba, Joseph, John Lynch, Barton Weitz, Chris Janiszewski, Richard Lutz, Alan Sawye, Stacey Wood.

1997. Interactive home shopping: consumer, retailer, and manufacturer incentives to participate in electronic marketplaces. *J. Marketing* 61 (July) 38-53.

[2] Barker, E. Virginia, and O'conner, Dennis E. 1989. Expert systems for configuration at digital: XCON and Beyond. *CACM*(March) 32 (3) 298-318.

[3] Borning, A., Maher, M., Martindale, A., Wilson, M., 1989. Constraint Hierarchies and Logic Programming, In *Proceedings of the Sixth International Conference on Logic Programming*. Lisbon, June. 149-164.

[4] Chang, Chao-Fan, and Shao-Kung Chang. 1998. A Layer-based Layout Approach for Semiconductor Fabrication Facilities. *IEEE/SEMI Advanced Semiconductor Manufacturing Conferences*. 385-390.

[5] Feinberg, Fred M., Joel Huber. 1996. A theory of cutoff formation under imperfect information, *Management Sci.* 42 (1) 65-84.

[6]Freuder, Eugene C. 1998. The Role of configuration knowledge in the business process. *IEEE Intelligent Systems* (July/August) 29-33.

[7] Häubl, Gerald, and Balerie Trifts. 2000. Consumer decision making in online shopping environments: the effects of interactive decision aids. *Marketing Sci.* 19 (1) 4-21.

[8] J. K. Lee, Sunh B. Kwon. 1995. ES*: An Expert Systems Development Planner Using a Constraint and Rule-based Approach. *Expert Systems with Applications* 9 (2).

[9] Klein, Rudiger.1996. A logic-based description of configuration: the constructive problem solving approach. *AAAI 96 Fall Symposium Workshop on Configuration*, MIT (Nov) 1-10.

[10] Kolodner, Janet. 1993. *Case-Based Reasoning*. Morgan Kaufman Publishers.

[11] Liebowiz, Jay.1990. Expert Configuration Systems: A Survey and Lessons Learned. *Expert Systems with Applications* 1 183-187.

[12] Moorthy, Sridhar, Brian T. Ratchford, and Debabrata Talukdar. 1997. Consumer information search revisited: theory and empirical analysis. *J. Consumer Res.*, 23 (March) 263-277.

[13] Payne, John W., James R. Bettman, Eric J. Jhonson. 1993. *The Adaptive Decision Maker*. Cambridge University Press, Cambridge, UK.

[14] Ratchford, Brian T., and Narasimhan Srinivasan. 1993. An empirical investigation of returns to search. *Marketing Sci.* 12 (winter) 73-87.

[15] Sabin, Daniel, and Rainer Weigel. 1998. Product Configuration Frameworks-A Survey. *IEEE Intelligent Systems* (July/August) 42-49.

[16] Sathi, Neena, Mark S. Fox, Rajay Coyal, and Alexander S. Kott.1992. Resource Configuration and Allocation: A Case Study of Constrained Heuristic Search. *IEEE*.

[17] Stefik, Mark. 1995. *Introduction to Knowledge systems*. Morgan Kaufman Publishers.

[18] Yu, Bei, and Hans Jargen Skovgaard. 1998. A Configuration Tool to Increase Product Competitiveness. *IEEE Intelligent Systems*, (July/August) 34-41.

[19] Wilke, Smyth, Cunningham. 1999. Using configuration techniques for adaptation. *Case based reasoning technology* (M. Lenz, B. Bartsch-Spörl, H.-D. Burkhard, S. Wess (Eds.)). 139-168. Springer-Verlag New York Heidelberg Berlin.

[20] Wright, Peter. 1975. Consumer choice strategies: simplifying vs. optimizing. *J. Marketing Res.* 12 (February) 60-67.