

조합 경매에서의 비드 선택 휴리스틱

송진우^o 양성봉
연세대학교 컴퓨터과학과
(fantaros, yang)@cs.yonsei.ac.kr

A Bid Selection Heuristic for Combinatorial Auction

Jin-Woo Song^o Sung-Bong Yang
Dept. of Computer Science, Yonsei University

요 약

조합 경매는 구매자들이 원하는 상품들을 조합해서 입찰할 수 있는 경매다. 그러나 판매자의 이익을 최대로 하는 승자를 결정하는 문제는 \mathcal{NP} -complete이다. 분기 한정법(branch and bound)을 사용해서 최적의 승자를 찾을 때, 어떤 비드(bid)를 선택해서 분기할 것인가를 결정해야 한다. 이 때, 비드를 선택하는 휴리스틱이 분기 한정법의 성능을 결정하는 중요한 역할을 한다. 본 논문에서는 조합경매의 승자를 결정하기 위해서 분기 한정법과 Linear Programming을 사용할 때, 분기할 비드와 충돌하는 비드들을 동시에 고려하는 비드 선택 휴리스틱을 제안한다. 실험을 통해서 제시하는 휴리스틱의 성능이 우수함을 보였다.

1. 서론

경매(Auction)에서는 다양한 종류의 상품들이 거래되고 있고, 상품들 간에 상보성(complementarity)이 존재하는 경우가 있다. 미국 FCC(Federal Communication Community) 스펙트럼 경매와 airport time slot 경매[1]같은 예를 들 수 있다. 조합경매(Combinatorial auction)에서는 이와 같이 상품들의 상보성을 고려할 수 있다.[2][3][4][5] 그러나 조합 경매에서 최적의 구매자를 선택하는 문제는 \mathcal{NP} -complete이다.[2]

이 문제를 보다 빠르게 풀기 위해서 여러 가지 방법이 제시되었었고, 그 중 Mixed Integer Programming을 푸는 상용 프로그램인 ILOG의 CPLEX[6]와 분기 한정법을 이용한 CABOB[7]이 좋은 결과를 보였다. 본 논문에서는 분기 한정법에서 분기를 할 비드를 선택하는 휴리스틱을 제안하고, 제시하는 휴리스틱과 성능향상을 위한 기법들이 적용된 분기 한정법의 성능을 기존의 논문들과 비교한다.

2장에서는 관련연구에 대해 알아보고, 3장에서는 조합 경매의 정의에 대해 알아본다. 4장에서는 본 논문에서 제시하는 휴리스틱과 그 외의 알고리즘에 대해 기술한다. 5장에서는 실험 결과에 대해 알아보고 6장에서는 결론과 앞으로 연구할 방향 대해 알아본다

2. 관련연구

2.1 MIP(Mixed Integer Programming)

조합경매의 승자를 결정하는 문제는 MIP로 수식화될 수 있다.[6] Andersson은 MIP를 풀어주는 상용 프로그램인 CPLEX 6.5를 사용해서 여러 데이터 분포에 대해서 기존의 실험 결과들과 비교를 했다. 그 결과 CPLEX는 좋은 성능을 보였고, 또 특정한 데이터 분포뿐만 아니라 일반적인 데이터에 대해서도 좋은 성능을 나타냈다.

2.2 CABOB(Combinatorial Auction Branch On Bids)

CABOB[7]은 Sandholm이 제시한 알고리즘으로 기본적으로 깊이 우선 탐색 분기 한정법(Depth First Branch and Bound)으로 조합경매의 승자 결정을 한다. CABOB에서도 좋은 최대 한계치(upper bound)를 빠른 시간에 구하기 위해서 CPLEX의 Linear Programming(LP)를 사용했다. 그리고 분기할 bid를 선택하는 휴리스틱으로 OB(One Bids)와 NSS(Normalized Shadow Surplus)를 혼합한 방식을 적용했다. 그 밖에도 성능을 향상시키기 위한 다양한 휴리스틱을 제시했다. CABOB의 실험에서는 여러 데이터 분포에 대한 실험 결과를 CPLEX 7.0의 결과와 비교를 했고, CABOB이 일부 데이터 분포에서 더 좋은 결과를 보였다.

3. 조합 경매(Combinatorial Auction)

3.1 정의

조합 경매는 입찰자들이 다수의 상품들을 하나의 꾸러미(package)로 묶어서 입찰할 수 있는 경매다. 따라서 상품들간의 상보성을 표현할 수 있다. 판매자가 광고자 하는 상품들에 대해서 조합 경매를 실시하면 입찰자들은 각자가 원하는 꾸러미와 꾸러미에 대한 입찰 가격을 제시한다. 판매자는 자신의 이익을 최대로 하는 입찰자들을 선택하고, 선택된 입찰자들이 승자가 된다. 판매자의 이익은 승자들이 제시한 입찰 가격들의 합이 된다.

n 명의 입찰자가 m 개의 상품에 대해 입찰을 할 때,

- 입찰자 집합: $B = \{b_1, b_2, \dots, b_n\}$
- 상품 집합: $S = \{1, 2, \dots, m\}$
- 입찰자 b_i 의 입찰: $b_i = (s_i, p_i)$ (단, s_i 는 공집합을 제외한 S 의 부분집합. p_i 는 s_i 에 대한 입찰가격)

3.2 승자 결정(Winner Determination)

판매자의 이익을 최대화 하는 승자들은 다음과 같다.

$$Max \sum_{i=1}^n x_i p_i \quad s.t. \quad \sum_{i|j \in S_i} x_i \leq 1, \forall j \in S \quad (1)$$

$$x_i \in \{0,1\}, \text{ for each } i = 1,2,\dots,n.$$

(x_i 가 1이면 승자, 0이면 패자)

3.3 LP

상품이 승자들에게 분할될 수 있다면 LP식은 식(1)에서 $x_i \in \{0,1\}$ 이 $x_i \in \mathfrak{R}^+$ 로 바뀐 식과 같다.

분기 한정법에서는 한계치를 이용해서 탐색 공간을 가지치기(prune)하면서 최적의 해를 찾아나간다. 이때 LP를 사용하면 좋은 최대 한계치를 구할 수 있고, 이는 최적의 해를 찾는 시간을 줄이는데 영향을 준다.[4]

4. 휴리스틱과 알고리즘

4.1 비드 선택 휴리스틱(Bid Selection Heuristic)

분기할 때 최적의 해에 포함되는 비드라고 생각되는 비드를 선택하기 위해서, 제안하는 휴리스틱 방법들은 어떤 비드와 충돌하는(상품이 중복되는) 비드들의 x 값도 같이 고려한다.

• Conflict Bids Sum(CBS) : 비드 i 의 x 값이 0.5이상이면 비드 i 와 충돌하는 비드들의 x 값의 합을 계산해서 CBS_i 를 구한다. 비드 i 의 x 값이 0.5미만이면 비드 i 의 x 값이 CBS_i 가 된다. 그리고 가장 높은 CBS_i 값을 가지는 비드 i 가 선택된다.

$$CBS_i = \begin{cases} \sum_{j=1}^i x_j & , (x_i \geq 0.5) \\ x_i & , (x_i < 0.5) \end{cases} \quad (2)$$

이때, x_j 는 비드 j 와 충돌하는 b_j 의 x 값, $b_j \in B$

$$CBS = \max(CBS_i) \quad (3)$$

단, $i = 1,2,\dots,n$

• Sum of Adjacent Sum(SAS) : 비드 i 와 충돌하는 모든 비드들의 x 값을 더하고, 충돌하는 비드의 수만큼 비드 i 의 x 값을 더한다.

$$SAS_i = \sum_{j=1}^i x_j + x_i \quad (4)$$

이때, x_j 는 비드 j 와 충돌하는 비드 i 의 x 값, x_i 는 구하려는 비드 i 의 x 값.

$$SAS = \max(SAS_i) \quad (5)$$

단, $i = 1,2,\dots,n$

4.2 최대 한계치 재사용(Upper Bound Reuse)

남아있는 비드들의 최대 한계치를 구하기 위해서는 LP를 계산한다. 만일 식(3)또는 (5)에서 선택된 비드의 x 값이 1인 경우, 선택한 비드의 방향으로 분기할 때의 최대 한계치는 분기하기 이전의 최대 한계치에서 선택한 비드의 가격을 뺀 값과 동일하므로 이를 재사용하면 최대한계치를 구하는 시간을 줄일 수 있다.

4.3 BDS(Best-Depth-Search)

BFS(Best-First-Search)는 최대 한계치가 큰 방향으로 탐색을 하지만 필요한 메모리 공간이 지수적으로 증가하므로 문제가 커질 경우 메모리 부족 문제가 발생한다. 그에 반해 DFS(Depth-First-Search)는 선형 공간(linear space)을 사용하는 이점이 있다.[7]

하지만 DFS는 한계치에 의해 가지치기되지 않는 한 끝까지 탐색을 하기 때문에 BFS보다 더 많은 탐색을 할 수 있다. 그래서 본 논문에서는 BFS와 DFS를 혼합한 형태의 탐색 방법인 BDS(Best-Depth-Search)를 사용했다.

BDS는 처음에는 BFS를 사용해서 탐색을 하다가 남은 비드들의 수가 일정수준 이하가 되면 DFS를 사용해서 남은 비드들을 탐색을 한다. DFS를 통해 끝까지 탐색을 해도 남은 비드의 수가 적기 때문에 빠른 시간에 찾은 해가 최적인지 판단할 수 있다. 문제의 크기가 크더라도 탐색의 후반부에는 DFS를 사용하기 때문에 BFS만을 사용하는 것 보다는 메모리 문제에 있어 안정적인 탐색이 가능하다.

4.4 Integer Case 와 Complete Case

• Integer Case : 남은 비드들에 대해 LP를 통해 최대 한계치를 구했을 때, 모든 비드들의 x 값이 0이나 1인 경우 Integer Programming의 해가 된다. 따라서 x 가 1인 비드들이 최적의 해가 되므로 더 이상의 탐색이 필요 없다.[7]

• Complete Case : 남은 비드들이 서로 충돌하는 경우 승자는 한 명밖에 될 수 없으므로, 가장 큰 가격을 제시한 입찰자가 선택된다.[7] 이러한 경우는 발생할 가능성이 적으므로 탐색을 시작하기 전에만 체크를 한다.

4.5 전처리(Preprocessing)

중복된 비드나 절대로 선택될 수 없는 비드들은 입력할 때 제거한다. 온라인 상으로 경매가 진행된다고 볼 때, 미리 입력하는 비드들을 전처리하는 시간은 승자를 결정하는데 있어서 제외될 수 있다.

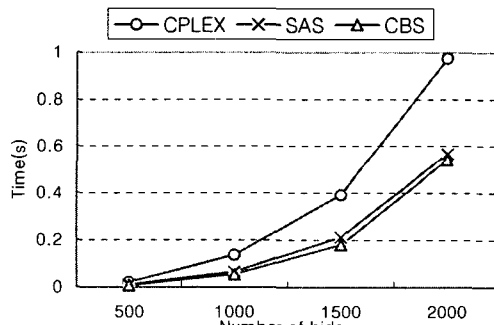
5. 실험 및 결과

5.1 실험 환경 및 데이터

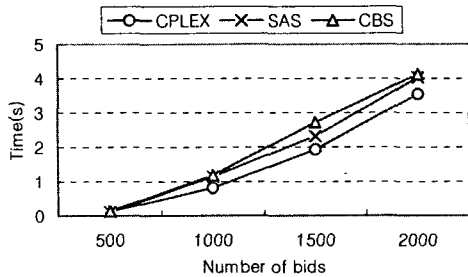
실험은 펜티엄IV-1.0G, 메모리 512M, 윈도우 2000 환경에서 이루어졌고 CPLEX는 7.0버전을 사용했다. 실험 데이터는 Sandholm이 제시한 5가지 데이터 분포(Random, Weighted Random, Decay, Bounded, Uniform)[7]를 사용했다. Uniform 분포를 제외한 각 데이터들의 상품의 수는 비드 수의 1/10이다. 각각의 데이터 분포마다 100개 개체의 평균 시간을 구했다.

5.2 결과

CABOB은 전처리를 하지 않은 입력 데이터를 CPLEX에 넣었기 때문에 전처리 시간이 CPLEX 시간에 포함되었었다. 본 논문의 실험에서는 전처리를 동일하게 거친 데이터들이 사용되었기 때문에 CPLEX의 시간이 CABOB의 결과보다 빨랐다.

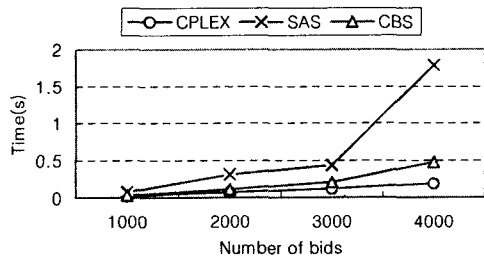


[그림 1] Random 분포



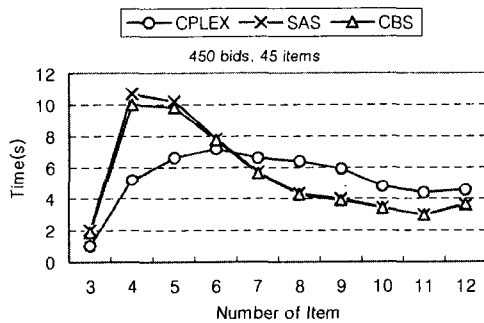
[그림 2] Weighted Random 분포

Random 분포(그림 1)에서는 SAS와 CBS가 CPLEX보다 더 빠른 결과를 얻었다. Weighted Random 분포(그림2)에서는 SAS와 CBS가 CPLEX보다 조금 느리긴 하지만 비슷한 증가 추세를 보였다.



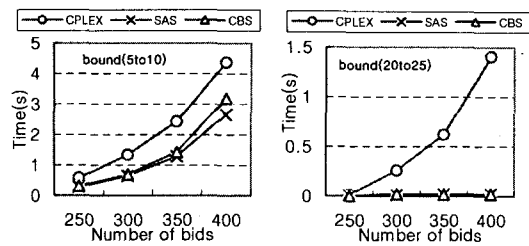
[그림 3] Decay 분포($\alpha = 0.45$)

Decay 분포(그림3)에서는 SAS가 CPLEX, CBS보다 느린 결과가 나왔다. Decay분포는 대부분이 Integer Case에 해당된다. 그렇지 않은 데이터들에 대해서 SAS는 비드의 x 값이 0.5로 동일한 비드들이 있을 때, 잘못된 방향으로 탐색을 해서 시간이 더 걸린다. 그러나 CBS는 이러한 경우 올바른 해를 찾는 방향으로 탐색을 한다.



[그림 4] Uniform 분포

Uniform 분포(그림 4)의 경우, SAS와 CBS는 비슷한 성능을 보였다. 상품의 수가 적은 경우($m = 4, 5$) CPLEX보다 느렸지만 그 외의 경우는 비슷하거나 더 빠른 성능을 보였다.



[그림 5] Bounded 분포

Bounded 분포(그림 5)에서는 바운드(bound)가 5에서 10인 경우 SAS가 가장 빠르고, CBS가 이와 비슷한 결과를 보였다. 바운드가 20에서 25인 경우는 모든 데이터가 Complete Case였기 때문에 시간이 거의 걸리지 않는다.

CBS와 SAS는 실험한 데이터 분포에 있어서 CPLEX와 비슷하거나 보다 나은 성능을 보였고 CABOB보다는 더 빠른 결과가 나왔다. CBS는 Decay 분포에서의 x 값이 같은(tie break) 상황을 SAS보다 잘 해결한다.

6. 결론 및 향후 연구과제

본 논문에서는 분기를 할 때 비드를 선택하는 휴리스틱들을 제안했다. 실험 결과 충돌하는 비드들의 x 값을 고려하는 SAS와 CBS를 사용한 분기 한정법이 기존의 방법들보다 빠른 시간에 최적의 해를 찾아냈다. 그리고 최대 한계치를 재사용함으로써 최대 한계치를 구하는 시간을 최소화 시켰다.

향후 과제로서는 실제 조항 경매에 사용되는 데이터 분포의 특성에 대해서 연구를 하고 이에 맞는 휴리스틱을 개발하는 것이 필요하다.

참고 문헌

- [1] S. J. Rassenti, V. L. Smith, and R. L. Bulfin, "A combinatorial auction mechanism for airport time slot allocation", *Bell Journal of Economics*, 13, pp. 402-417, 1982.
- [2] Michael H. Rothkopf, Aleksandar Pekec, and Ronald M Harstad, "Computationally manageable combinatorial auctions", *Management Science*, 44(8):1131-1147, 1995.
- [3] Yuzo Fujishima, Kevin Leyton-Brown, and Yoav Shoham, "Taming the computational complexity of combinatorial auctions: Optimal and approximate approaches", In *IJCAI*, pp. 548-553, 1999.
- [4] Noam Nisan, "Bidding and allocation in combinatorial auctions", In *ACM Conference on Electronic Commerce*, pp. 1-12, 2000.
- [5] Sven de Vries, Rakesh Vohra, "Combinatorial auctions: A Survey", Draft, 2000.
- [6] Arne Andersson, Mattias Tenhunen, and Fredrik Ygge, "Integer programming for combinatorial auction winner determination", In *Proceedings of the Fourth International Conference on MultiAgent Systems (ICMAS)*, pp. 39-46, 2000.
- [7] Tuomas Sandholm, Subhash Suri, Andrew Gilpin, and David Levine, "CABOB: A fast optimal algorithm for combinatorial auctions", In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 1102-1108, 2001.