Combinatorial Auctions, Knapsack Problems, and Hill-climbing Search

Robert C. Holte

School of Information Technology and Engineering,
University of Ottawa, Ottawa, Ontario, Canada, K1N 6N5
holte@site.uottawa.ca

Abstract. This paper examines the performance of hill-climbing algorithms on standard test problems for combinatorial auctions (CAs). On single-unit CAs, deterministic hill-climbers are found to perform well, and their performance can be improved significantly by randomizing them and restarting them several times, or by using them collectively. For some problems this good performance is shown to be no better than chancel; on others it is due to a well-chosen scoring function. The paper draws attention to the fact that multi-unit CAs have been studied widely under a different name: multidimensional knapsack problems (MDKP). On standard test problems for MDKP, one of the deterministic hill-climbers generates solutions that are on average 99% of the best known solutions.

1 Introduction

Suppose there are three items for auction, X, Y, and Z, and three bidders, B1, B2, and B3. B1 wants any one of the items and will pay \$5, B2 wants two items – X and one of Y or Z – and will pay \$9, and B3 wants all three items and will pay \$12. In a normal auction items are sold one at a time. This suits buyers like B1, but not B2 and B3: they cannot outbid B1 on every individual item they require and stay within their budget. If X is auctioned first it will likely be won by B1, and the seller's revenue (\$5) will be much worse than optimal (\$14). In a combinatorial auction (CA) each bid offers a price for a set of items (goods). Thus, bidders can state their precise requirements and the seller can choose the winners to optimize total revenue (sum of the selected bids' prices).

Combinatorial auctions have been studied since at least 1982[32], when they were proposed as a mechanism for selling time-slots at airports in order to permit airlines to bid simultaneously for takeoff and landing time-slots for a given flight. Fuelled by the FCC's interest [13] and potential e-commerce [22] and other applications [21, 25], interest in combinatorial auctions has increased rapidly in recent years. Among the many research issues raised, of main interest to AI is the fact that "winner determination" - selecting the set of winning bids to optimize revenue - is a challenging search problem. A recent survey of CA research is given in [10]. Previous research has mainly focused on single-unit CAs, in which there is exactly one copy of each item. Multi-unit CAs, in which there can be any number of identical copies of each item, were introduced to the AI community in [27], which claimed the problem was new. One contribution of the present paper is to point out that multi-unit CAs have been studied extensively in the Operations Research literature, where they are called multidimensional knapsack problems (MDKP).

This paper examines the performance of hill-climbing algorithms on standard test problems for CAs. Theoretical analysis shows that greedy algorithms cannot guarantee finding near-optimal solutions for winner determination [1, 7, 9, 12, 15, 17, 18, 29, 33, 34]. But these are mostly worst-case results, and in some cases apply only to specific types of greedy algorithm and not to the type of hill-climbing algorithm considered here. The main finding of this paper is that on the standard CA and MDKP test sets, hill-climbers perform very well.

2 The Hill-climbing Algorithms

The hill-climbing algorithms compared in this paper are identical except for the criterion used to select which successor to move to. Search begins with the empty set of bids and adds one bid at a time. Because bids have positive prices and search can only add bids to the current bid-set, the total revenue for the current bid-set must increase as search proceeds. Search terminates when a bid-set is reached that has no successors. The solution reported is the bid-set seen during search that maximizes revenue, which may be different than the local maximum at which the search terminated.

Each different way of adding one bid to the current bid-set creates a potential successor. A potential successor is eliminated if it is infeasible or if it can be shown that no extension of it can possibly yield a greater revenue than the best solution seen so far. For this purpose Sandholm & Suri's "admissible heuristic" (p.96, [35]) is used. Of the remaining successors the one that adds the bid with the maximum "score" is selected to be the current bid-set and and the process is repeated. Three different ways of computing a bid's score are considered:
Price: the bid's price
N2norm: the bid's price divided by its "size", where the size of bid j is the 2-norm (square root of the sum of squares) of the $f_{i,j}$, the fraction of the remaining quantity of item i that bid j requires.
KO: the bid's price divided by its "knockout cost", where a bid's knockout cost is the sum of the prices of the available bids that are eliminated if this bid is chosen. KO is the only novel scoring function; the others, and many variations of them, have been studied previously [5, 17, 23, 28, 39, 40].

Also included in the experiments is a form of randomized hill-climbing in which, after pruning, a successor is chosen randomly: the probability of choosing each successor is proportional to its score. Such hill-climbers can produce different solutions each time they are run. In the experiments each is restarted from the empty bid-set 20 times and the best solution on any of the runs is recorded. In the tables these are identified by appending $\times 20$ to the scoring function. For example, **Price**$\times 20$ is the randomized hill-climber that makes its probabilistic selection based on Price.

heuristic	arb	match	path	r75P	r90P	r90N	sched
1. Price	3	39	4	0	10	8	14
2. N2norm	4	39	11	0	11	9	14
3. KO	4	25	14	1	10	8	15
best of 1-3	5	54	28	1	11	9	27
4. Price×20	12	39	4	1	19	15	22
5. N2norm×20	9	39	11	0	15	13	19
6. KO×20	13	25	14	1	18	16	22
best of 4-6	19	54	28	2	26	22	37

Table 1. Percentage of problems on which the heuristic solution is optimal

3 The Test Problems

Test problems were generated using the CATS suite of problem generators version 1.0 [26]. Each problem generator in CATS models a particular realistic scenario in which combinatorial auctions might arise. For example, matching.c models the sale of airport time-slots. The experiments use each of CATS's five generators for single-unit CAs, one with 3 different parameter settings, for a total of seven different types of test problem. The abbreviations used to identify the type of test problem in the tables of results and the corresponding CATS program and parameter settings are as follows (default parameter settings were used except as noted): **arb** (arbitrary.c), **match** (matching.c), **path** (paths.c with NUMBIDS=150), **r90P** (regions.c), **r90N** (regions.c with ADDITIVITY= -0.2), **r75P** (regions.c with ADDITIONAL LOCATION= 0.75), **sched** (scheduling.c with NUMGOODS=20 and NUMBIDS=200).

100 instances of each problem type are generated. In addition to the hill-climbers, a systematic search algorithm is run in order to determine the optimal solution. This is a relatively unsophisticated branch-and-bound search. There were a handful of instances that it could not solve within a 1 million node limit; these are excluded from the results.

4 Heuristic Hill-climbing Experimental Results

Tables 1-3 have a column for each type of test problem and a row for each of the hill-climbers. There are also two "best of" rows. "Best of 1-3" refers to the best solution found by the deterministic hill-climbers on each individual test problem. "Best of 4-6" is the same but for the randomized hill-climbers. Because hill-climbing is so fast, these represent realistic systems which run a set of hill-climbers on a given problem and report the best of their solutions.

Table 1 shows the percentage of test problems of a given type that are solved optimally by a given hill-climber. On the **r75P** problems the hill-climbers almost never find the optimal solution. On **match**, **path**, and **sched** problems, the deterministic hill-climbers collectively (best of 1-3) find the optimal solution on

heuristic	**arb**	**match**	**path**	**r75P**	**r90P**	**r90N**	**sched**
1. Price	60-69 (1)	80-89 (8)	70-79 (2)	50-59 (3)	70-79 (7)	60-69 (1)	60-69 (1)
2. N2norm	70-79 (22)	80-89 (8)	80-89 (4)	60-69 (7)	70-79 (8)	60-69 (1)	70-79 (8)
3. KO	60-69 (1)	80-89 (3)	80-89 (3)	60-69 (16)	70-79 (7)	60-69 (1)	70-79 (2)
best of 1-3	70-79 (14)	90-99 (46)	80-89 (1)	60-69 (2)	70-79 (4)	70-79 (5)	80-89 (9)
4. Price×20	80-89 (23)	80-89 (7)	80-89 (28)	70-79 (2)	70-79 (1)	80-89 (18)	80-89 (11)
5. N2norm×20	70-79 (1)	80-89 (7)	80-89 (3)	70-79 (1)	80-89 (11)	80-89 (21)	80-89 (13)
6. KO×20	80-89 (29)	80-89 (1)	80-89 (3)	70-79 (2)	80-89 (13)	80-89 (16)	70-79 (1)
best of 4-6	80-89 (10)	90-99 (46)	90-99 (72)	80-89 (29)	80-89 (3)	80-89 (4)	80-89 (1)

Table 2. Suboptimality decile of the worst heuristic solutions

over a quarter of the problems, and on all types of problem except **r75P** the randomized hill-climbers collectively find the optimal solution on between 19% and 54% of the problems.

Table 2 summarizes the worst solutions found by each heuristic on each type of problem. The heuristic's solution, as a percentage of the optimal solution, is put into a 10-point bin, or decile (e.g. 63% falls in the 60-69% decile). The worst non-empty decile is reported in the table; in brackets beside the decile is the percentage of test problems that fell into that decile. For example, the $60-69(1)$ entry in the upper left indicates that on 1% of the **arb** problems, the solutions found by the Price hill-climber were 60-69% of optimal, and on none of the **arb** problems were this hill-climber's solutions worse than 60% of optimal. On all the problems all the hill-climbers find solutions that are 50% of optimal or better, and only very rarely do any of them find solutions worse than 70% of optimal. The solutions found by the randomized hill-climbers are very rarely worse than 80% of optimal.

Table 3 gives the average percentage of optimal of the solutions found by each heuristic on each type of problem. The first row is for the "blind" hill-climber discussed in the next section and will be ignored until then. **r75P** is clearly the most difficult type of problem for the hill-climbers. **arb**, **r90P** and **r90N** are moderately difficult for the deterministic hill-climbers. On all problem types other than **r75P** the randomized hill-climbers find solutions that are more than 90% of optimal on average (95% if taken collectively).

The differences between the solutions found by the different hill-climbers are not large in most cases, but paired t-tests indicate that some of the differences are significant ($p < 0.05$). On all types of problem except **match**, where the difference was not significant, a randomized hill-climber is significantly better than the deterministic version with the same scoring function and the randomized hill-climbers collectively ("best of 4-6") are significantly better than the deterministic hill-climbers collectively. The Price scoring function is never superior to others. For deterministic hill-climbing KO is significantly better than N2norm on **sched** problems, but the opposite is true on **arb**, **path** and **r75P** problems. For randomized hill-climbing N2norm is significantly better than KO on **path**

heuristic	arb	match	path	r75P	r90P	r90N	sched
blind	84	63	52	73	90	88	65
1. Price	85	97	91	75	90	89	92
2. N2norm	87	97	97	81	90	89	92
3. KO	86	97	96	79	90	89	94
best of 1-3	87	99	98	83	90	89	96
4. Price×20	94	97	92	88	95	94	95
5. N2norm×20	93	97	97	89	94	94	95
6. KO×20	93	97	96	90	95	94	96
best of 4-6	95	99	98	92	96	96	98

Table 3. Average solution value as a percentage of optimal

problems; on all other types of problem either the difference is not significant or KO is better.

The overall conclusion of this experiment is that hill-climbing always finds acceptable solutions, usually very good ones, for the problem types studied. **r75P** is the most challenging problem type. On it the hill-climbers rarely find an optimal solution, but the randomized hill-climbers collectively find a solution that is at least 90% of optimal more than 60% of the time. Thus, very good solutions are found most of the time even on the most challenging type of problem. Problem types **match**, **path**, and **sched** are the easiest. For them very good solutions can almost always be found even by the deterministic hill-climbers (collectively).

5 Blind Hill-climbing

The experiment in this section was suggested by the unexpectedly strong performance of Monte Carlo search on some of the standard test problems for the multidimensional knapsack problem [4]. The previous section has shown that the scoring mechanisms used by the hill-climbers, especially KO and N2norm, lead to good solutions. But perhaps a blind hill-climber, which, after pruning, selects among successors randomly with uniform probability, would do equally well. To examine this, 100 instances of each problem type were generated, as above, and solved by the deterministic hill-climbers. In addition, each instance was solved 200 times by the blind hill-climber.

Table 4 gives the percentage of the blind hill-climber's solutions that are strictly worse than the solution found by a particular deterministic hill-climber on each problem type. On **match**, **path** and **sched** problems the deterministic hill-climbers virtually always outperformed the blind hill-climber. On these types of problem a well-chosen scoring function is essential for good performance. On the other types of problem the scoring functions were no better than chance. This may also be seen by comparing the blind hill-climber's average solution quality – the first row in Table 3 – with the averages for the deterministic hill-climbers.

heuristic	arb	match	path	r75P	r90P	r90N	sched
1. Price	20	100	100	53	7	6	98
2. N2norm	40	100	100	76	16	23	99
3. KO	20	100	100	63	7	6	99

Table 4. Percentage of blind solutions worse than heuristic solutions

% of optimal	arb	match	path	r75P	r90P	r90N	sched
10 – 19			0.01				
20 – 29		0.005	2				
30 – 39		1	13				0.05
40 – 49		8	29	0.12			6
50 – 59		29	32*	4			30
60 – 69	2	38*	19	30		0.56	33*
70 – 79	21	20	5	48*	9	13	20
80 – 89	63*	4	0.7	16	46*	45*	8
90 – 99	13	0.1	0.03	2	34	33	1.5
100	1				11	9	0.075

Table 5. Percentage of blind solutions in each suboptimality decile

Each column in Table 5 is a histogram. Each of the blind hill-climber's solutions is expressed as a percentage of the optimal solution and put into the appropriate decile. The table shows what percentage of the solutions fall into each decile for each type of problem. For example, the 0.01 at the top of the **path** column means that on problems of type **path** 0.01% of the blind hill-climber's solutions were 10-19% of optimal (i.e. extremely poor). A blank entry represents 0. In each column an asterisk indicates the median decile. On **match**, **path** and **sched** problems blind hill-climbing sometimes produces very poor solutions and has a poor median. The opposite is true of **arb**, **r90P** and **r90N**. On these types of problems no blind hill-climbing solution is worse than 60% of optimal and the median decile is 80-89%. **r75P** is of medium difficulty. The bottom row gives the percentage of blind hill-climbing runs which find the optimal solution. Comparing this to the deterministic hill-climbing rows in Table 1, it is apparent that on **arb**, **r90P** and **r90N** problems the ability of the scoring functions to guide the hill-climber to optimal solutions is no better than chance, whereas on **match**, **path** and **sched** problems they are far better than chance.

Two overall conclusions follow from the experiments in this and the preceding section. In problems of type **match**, **path** and **sched** good solutions are relatively rare, but the scoring functions are very effective for these problems and deterministic hill-climbing performs very well on them. If suboptimal solutions are acceptable, problems of these types (with the parameter settings used in this study) are not especially promising as testbeds for comparing search strategies. By contrast, for problems of type **arb**, **r90P**, **r90N** and **r75P** the guidance

hill-climber	mknap1	mknap2	mknapcb1	mknapcb2	mknapcb3	mknapcb7
1. Price	90	94	89	89	89	93
2. N2norm	98.99	99.00	98.94	99.03	99.21	98.35
3. KO	83	79	85	85	85	85
blind	84	58	82	83	83	81

Table 6. Average solution value as a percentage of optimal

of the scoring functions is no better than chance. These types of problems are therefore good choices for evaluating search strategies, but in using them it is crucial to take into account the high baseline performance of blind hill-climbing.

6 Multidimensional Knapsack Experimental Results

A multi-unit combinatorial auction is precisely a multidimensional knapsack problem (MDKP): each item in the auction is a "dimension" and including a bid in the solution bid-set corresponds to putting the bid into the knapsack. MDKP has been the subject of several theoretical analyses [6, 9, 12, 15, 29, 38] and experimental investigations involving all manner of search methods, including genetic algorithms[8, 20, 24], TABU search [2, 19], local search [5, 11, 30] and classical complete algorithms such as branch-and-bound [16, 37] and dynamic programming [41]. A good review of previous work is given in [8].

A standard set of test problems for the MDKP is available through J. Beasley's ORLIB[3]. Files mknap1 [31] and mknap2 [14, 36, 37, 41] contain real-world test problems widely used in the literature. The others were generated with the aim of creating more difficult problems[8]. Each problem has an associated best known solution, which in some cases is known to be optimal, and in all cases is extremely close to optimal.

The hill-climbing algorithms were run on the six test sets indicated in the column headings of Table 6. N2norm performs extremely well. Its average solution is 99% of the best known solution on all the test sets except mknapcb7, where its average is 98.35%. Only on two problems in mknap2 is its solution worse than 90% of the best known (those solutions are in the 80-89% range). On more than 25% of the problems in mknap1 and mknap2 its solution is equal to the best known (this virtually never happens for the other test sets). N2norm is competitive with all previously reported systems on these datasets, and superior to previous "greedy" approaches. The blind hill-climber's median decile is 50-59% for mknap2, but it is 80-89% for the other test sets, indicating that very good solutions are abundant. KO performs poorly in the multi-unit setting.

7 Conclusions

The primary aim of this paper has been to examine the performance of hill-climbing algorithms on standard test problems for combinatorial auctions (CAs).

On the CATS suite of test problems for single-unit CAs deterministic hill-climbers perform well, and their performance can be improved significantly by randomizing them and restarting them several times, or by using them collectively. For some types of problem their performance, although good, is no better than chance: these types of problems therefore have an abundance of high-quality solutions. Providing the chance performance baseline is taken into account these problems are good testbeds for comparative studies. On the other types of CATS problems the good performance is due to the scoring function that guides the hill-climbing. Unless parameter settings can be found which result in poor performance by the hill-climbers, these problems are not especially good choices for testbeds in experiments where suboptimal solutions are permitted. On the standard test problems for multi-unit CAs (also known as multidimensional knapsack problems) deterministic hill-climbing using N2norm as a scoring function generates solutions that are on average 99% of the best known solutions; it is therefore competitive with all previously reported systems on these problems.

7.1 Acknowledgements

I am grateful to the maintainers of the compendium of NP optimization problems [9], and to J. Beasley and the authors of CATS for creating and distributing their tests suites. ORLIB is available at http://mscmga.ms.ic.ac.uk/info.html. CATS is available at http://robotics.stanford.edu/CATS/. This research was supported in part by an operating grant from the Natural Sciences and Engineering Research Council of Canada.

References

[1] E. M. Arkin and R. Hassin. On local search for weighted packing problems. *Mathematics of Operations Research*, 23:640–648, 1998.

[2] R. Battiti and G. Tecchiolli. Local search with memory: Benchmarking RTS. *Operations Research Spectrum*, 17(2/3):67–86, 1995.

[3] J. E. Beasley. OR-library: distributing test problems by electronic mail. *Journal of the Operational Research Society*, 41(11):1069–1072, 1990.

[4] M. Bertocchi, I. Brandolini, L. Slominski, and J. Sobczynska. A monte-carlo approach for 0-1 programming problems. *Computing*, 48:259–274, 1992.

[5] M. Bertocchi, A. Butti, L. Slominski, and J. Sobczynska. Probabilistic and deterministic local search for solving the binary multiknapsack problem. *Optimization*, 33:155–166, 1995.

[6] A. Caprara, H. Kellerer, U. Pferschy, and D. Pisinger. Approximation algorithms for knapsack problems with cardinality constraints. *European Journal of Operational Research*, 123:333–345, 2000.

[7] Barun Chandra and Magnus M. Halldorsson. Greedy local improvement and weighted set packing approximation. *Proceedings of the tenth annual ACM-SIAM Symposium on Discrete Algorithms*, pages 169–176, 1999.

[8] P.C. Chu and J.E. Beasley. A genetic algorithm for the multidimensional knapsack problem. *Journal of Heuristics*, 4:63–86, 1998.

[9] Pierluigi Crescenzi and Viggo Kann. A compendium of NP optimization problems. http://www.nada.kth.se/~viggo/wwwcompendium.

[10] Sven de Vries and R. Vohra. Combinatorial auctions: A survey. Technical Report (discussion paper) no. 1296, The Center for Mathematical Studies in Economics and Management Science, Northwestern University, 2000.

[11] A. Drexl. A simulated annealing approach to the multiconstraint zero-one knapsack problem. *Computing*, 40:1–8, 1988.

[12] M.E. Dyer and A. M. Frieze. Probabilistic analysis of the multidimensional knapsack problem. *Maths. of Operations Research*, 14(1):162–176, 1989.

[13] FCC. Public notice DA00-1075: Auction of licenses in the 747-762 and 777-792 MHZ bands scheduled for September 6, 2000: Comment sought on modifying the simultaneous multiple round auction design to allow combinatorial (package) bidding.

[14] A. Freville and G. Plateau. Hard 0-1 multiknapsack test problems for size reduction methods. *Investigation Operativa*, 1:251–270, 1990.

[15] A. M. Frieze and M. R. B. Clarke. Approximation algorithms for the m-dimensional 0-1 knapsack problem: worst-case and probabilistic analyses. *European Journal of Operational Research*, 15:100–109, 1984.

[16] B. Gavish and H. Pirkul. Efficient algorithms for solving multiconstraint zero-one knapsack problems to optimality. *Mathematical Programming*, 31:78–105, 1985.

[17] Rica Gonen and Daniel Lehmann. Optimal solutions for multi-unit combinatorial auctions: Branch and bound heuristics. *Proceedings of the Second ACM Conference on Electronic Commerce (EC-00)*, pages 13–20, 2000.

[18] M. M. Halldorsson. Approximations of weighted independent set and hereditary subset problems. *J. Graph Algorithms and Applications*, 4(1):1–16, 2000.

[19] S. Hanafi and A. Freville. An efficient tabu search approach for the 0–1 multidimensional knapsack problem. *European Journal of Operational Research*, 106(2-3):663–697, 1998.

[20] Christian Haul and Stefan Voss. Using surrogate constraints in genetic algorithms for solving multidimensional knapsack problems. In David L. Woodruff, editor, *Advances in Computational and Stochastic Optimization, Logic Programming, and Heuristic Search: Interfaces in Computer Science and Operations Research*, chapter 9, pages 235–251. 1998.

[21] Luke Hunsberger and Barbara J. Grosz. A combinatorial auction for collaborative planning. *Proceedings of the Fourth International Conference on Multi-Agent Systems (ICMAS-2000)*, pages 151–158, 2000.

[22] Joni L. Jones. Incompletely specified combinatorial auction: An alternative allocation mechanism for business-to-business negotiations (Ph.D. thesis).

[23] A. H. G. Rinnooy Kan, L. Stougie, and C. Vercellis. A class of generalized greedy algorithms for the multi-knapsack problem. *Discrete Applied Mathematics*, 42:279–290, 1993.

[24] S. Khuri, T. Back, and J. Heitkotter. The zero/one multiple knapsack problem and genetic algorithms. *Proceedings of the ACM Symposium of Applied Computation*, pages 188–193, 1993.

[25] Erhan Kutanoglu and S. David Wu. On combinatorial auction and Lagrangean relaxation for distributed resource scheduling. Technical report, Lehigh University, April, 1998.

[26] K. Leyton-Brown, M. Pearson, and Y. Shoham. Towards a universal test suite for combinatorial auctions algorithms. *Proceedings of the Second ACM Conference on Electronic Commerce (EC-00)*, pages 66–76, 2000.

[27] K. Leyton-Brown, Y. Shoham, and M. Tennenholtz. An algorithm for multi-unit combinatorial auctions. *Proceedings of the Seventeenth National Conference on Artificial Intelligence (AAAI-2000)*, pages 56–61, 2000.

[28] R. Loulou and E. Michaelides. New greedy-like heuristics for the multidimensional 0-1 knapsack problem. *Operations Research*, 27:1101–1114, 1979.

[29] Michael J. Magazine and Maw-Sheng Chern. A note on approximation schemes for multidimensional knapsack problems. *Mathematics of Operations Research*, 9(2):244–247, 1984.

[30] M. Ohlsson, C. Peterson, and B. Soderberg. Neural networks for optimization problems with inequality constraints - the knapsack problem. Technical Report LU TP 92-11, Dept. of Theoretical Physics, Univ. of Lund, 1992.

[31] C. C. Petersen. Computational experience with variants of the Balas algorithm applied to the selection of R&D projects. *Management Science*, 13(9):736–750, 1967.

[32] S. J. Rassenti, V. L. Smith, and R. L. Bulfin. A combinatorial auction mechanism for airport time slot allocation. *Bell Journal of Economics*, 13:402–417, 1982.

[33] M. H. Rothkopf, A. Pekec, and R. M. Harstad. Computationally manageable combinatorial auctions. *Management Science*, 44(8):1131–1147, 1998.

[34] Shuichi Sakai, Mitsunori Togasaki, and Koichi Yamazaki. A note on greedy algorithms for maximum weighted independent set problem. http://minnie.comp.cs.gunma-u.ac.jp/ koichi/TecRep/gamis.ps.

[35] Tuomas Sandholm and Subhash Suri. Improved algorithms for optimal winner determination in combinatorial auctions and generalizations. *Proceedings of the Seventeenth National Conference on Artificial Intelligence (AAAI-2000)*, pages 90–97, 2000.

[36] S. Senju and Y. Toyoda. An approach to linear programming with 0-1 variables. *Management Science*, 11:B196–B207, 1967.

[37] Wei Shih. A branch and bound method for the multiconstraint zero one knapsack problem. *J. Operational Research Society*, 30(4):369–378, 1979.

[38] Krzysztof Szkatula. On the growth of multi-constraint random knapsacks with various right-hand sides of the constraints. *European Journal of Operational Research*, 73:199–204, 1994.

[39] J. Thiel and S. Voss. Some experiences on solving multiconstraint zero-one knapsack problems with genetic algorithms. *INFOR*, 32(4):226–242, 1994.

[40] Y. Toyoda. A simplified algorithm for obtaining approximate solution to zero-one programming problems. *Management Science*, 21:1417–1427, 1975.

[41] H. M. Weingartner and D. N. Ness. Methods for the solution of the multidimensional 0/1 knapsack problem. *Operations Research*, 15:83–103, 1967.