An Efficient Approximation Algorithm for Combinatorial Auctions

G. Anandalingam, R.Kwon, L.H. Ungar

Robert H. Smith School of Business, University of Maryland, College Park, MD 20742

Department of Systems Engineering, University of Pennsylvania, Philadelphia, PA 19104

Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA 19104

Abstract

We present a mathematical programming approximation approach to the winner determination problem for multi-round combinatorial auctions. The winner determination problem is a set packing problem, and hence NP-complete. Most methods developed recently rely on exhaustive search based methods which are exponential in worst case time complexity. We develop a two phase method where the first phase is a primal-dual type approximation algorithm that rapidly computes an initial feasible solution and the second phase is a refinement procedure that uses the dual of the LP relaxation of the winner determination problem and a greedy heuristic to improve upon the solution from the first phase. We present novel performance guarantees for the two phase method. In addition, single item prices are maintained that represent approximate marginal values for items. The method was tested against CPLEX MIP Solver 7.0 on several representative distributions from the literature and was found to achieve high solution quality in addition to scaling well in the number of items and bids with up to three orders of magnitude faster than CPLEX MIP Solver 7.0.

- **Keywords:** (combinatorial auctions; integer programming; approximation algorithms; set packing)

1 Introduction

In this paper, we develop a two-phase mathematical programming based procedure to approximate the winner determination problem for iterative combinatorial auctions. Combinatorial auctions are auctions where single bids are allowed on multiple distinct items. Single item auctions have been studied for many of years and particular mechanisms that result in incentive compatible efficient auctions have been discovered Vickrey (1961). In recent years, combinatorial auctions have become the focus of intense research. For a recent survey, DeVries and Vohra (2001). It stands at this time that there are no practical combinatorial auction formats that are guaranteed to be incentive compatible. The only incentive compatible combinatorial auction is a generalized version of the Vickrey auction but it is computationally intractable DeVries and Vohra (2001). Hobbs et al (2000) give examples of other reasons why the generalized Vickry auction may not be effective. Recent auctions held by the FCC for allocating spectrum licenses employed a simulateous ascending multiple round format whereby bidders could place separate simultaneous bids on spectrum licenses over multiple rounds, where bids on items are non-decreasing over rounds. The multi-round format allows the posting of tentative auction results so that bidders can use the information to bid more efficiently, this is analogous to the ascending English auction for single items. This feature helps combat the winners curse that may occur in the attempt to attain multiple items. However, the price that a bidder offers for one item may depend in a complicated way on what other items he wins. This is generally the case when a set of items has synergistic values. For example, in the FCC auctions many firms would want a set of licenses that are adjacent geographically so that the firm can have economies of scale in radiowave services. One major drawback in the FCC simultaneous auction format is that bidders may face substantial risk in obtaining an incomplete set of licenses see Cramton et al (1998). Inefficiency may result as bidders may be too conserative in their bids for items in fear of not obtaining a complete set. Indeed, in a number of settings it may be desirable to simply allow bids on combinations of objects where the bidders reveal the set of items that they truly desire. Thus, one natural extension to the multiple round simultaneous ascending format is to allow for combinatorial bidding.

Iterative combinatorial auctions have been considered for wireless spectrum allocation by the Federal Communications Agency (FCC) by Milgrom (2000), and for general resource allocation by DeMartini et al (1999). In the latter work, an iterative combinatorial auction method is developed which is called RAD. The RAD design is essentially the multiple round simultaneous ascending auction but for combinatorial bidding. Thus, it inherits the activity, bidding, and stopping rules of this method. These mechanisms go towards the minimization of gaming and thus tries to induce bidders to truthfully bid for packages Milgrom (2000). No theoretical incentive compatible or efficiency properties exist for multi-round combinatorial bidding under general conditions. A recent design by Parkes (2000) obtains efficiency of allocation under the assumption of a particular myopic bidding strategy and incentive compatibility in certain cases. Parkes et al (2001). We assume only that bidders have private valuation functions for packages.

One major hurdle in the design of combinatorial auctions is in the allocation of packages to bidders. This problem, known as the winner determination problem (WD), is computationally intensive. Winner determination is equivalent to the weighted set packing problem which belongs to the class of integer programs which are NP-hard (For a detailed exposition of this issue see Rothkopf et al (1998). Algorithms for combinatorial auctions include provably optimal and polynomial time allocation algorithms for a restricted class of bids, and provably optimal allocations that are obtained by enumerative search over the allocation space Nisan (2000). Neither of these works explicitly considers iterative auction design in that the main focus is on attaining allocations that are feasible and close to optimal. Park et al 2001 discuss an alternative mechanism whereby bidders endogenously determine bids submit an ordered list of bids to the auctioneer.

In general, approximate dual information may be useful to bidders in multi-round settings. The resource allocation design (RAD) mechanism adapts the iterative ascending auction mechanism to auctions with package bidding. During each round, bidders submit bids on packages and then the auctioneer determines a provisional winning allocation of bundles to bidders. Winner determination (WD) is performed by solving a set packing problem, and the new prices to be charged for single items in the next round are obtained from an approximation to the dual of this problem. One weakness with the RAD mechanism is the assumption that the WD can be solved exactly which can not guaranteed for large scale instances of the problem in a reasonable time frame. Rothkopf et al 1998 reviews classes of bids that result in tractability for the WD problem and presents specialized polynomial time algorithms for these classes of bids. Nisan (2000) considers the strategy of using linear programming (LP) relaxations for the WD problem, and highlights the conditions under which the LP relaxation actually provides integer solutions to the WD problem. In most cases, however, the LP relaxation, as expected, provides fractional solutions. In order to obtain feasible, i.e. integer solutions, a ranking heuristic based on the dual information obtained from the LP relaxation is employed to obtain solutions quickly. As noted in the paper, it is unclear that this method will always give "quality" solutions, i.e. those that are close to optimality. In bad cases, the use of branch-and-bound using the rounded fractional solutions as a starting point is suggested. The mixed integer solver of Cplex Version 7.0 employs such strategies similar to those suggested by Nisan (2000).

Another class of methods developed for the WD problem involves search based methods similar to branch and bound but employ various heuristics to effectively prune search trees Fujishima et al (1999). Sandholm (2000) also considers such a tree based search method with the additional use of linear programming relaxations. However, these methods are still exponential in the worst case and only effective in solving small problem instances. In general, most methods that are exact or approximate for solving the WD problem focus on obtaining feasible solutions that hopefully are close to optimal. For single round auctions this may be a very reasonable strategy. However, it may be beneficial to construct approximation algorithms that also allow construction of approximate dual information that may be useful to bidders for multiple round auctions. Approximate dual information for individual item prices can be used as approximate mariginal values that enable bidders to bid more efficiently in subsequent rounds (Demartini et al, 2000).

In this paper, we consider the winner determination problem in a multiple round setting, and similar to RAD we use allocation (primal) and pricing (dual) mechanisms iteratively. Our main contribution is in the design of a two phase procedure to approximate the WD problem specifically for a multiple round setting. For the first phase we develop a polynomial time primal-dual algorithm that rapidly computes an intial feasible solution to the WD problem. The second phase consists of two components. The first component is a refinement procedure based on the optimal solution of the dual of the WD relaxation that seeks to improve the solution from the first phase. The second component is a greedy heuristic that improves on the solution from the first component by searching among unallocated packages with higher rewards than some packages in the tentative allocation but with more items in the package.

In addition, the method that we consider constructs and maintains prices for individual items as well as for packages. The pricing of items is such that a winning bid is equal to the sum of the prices of the individual items in the package that was allocated, and that sum of the prices in unallocated package is greater than or equal to the losing bid offer. Such single item prices enable bidders to bid more efficiently in multi-round auctions as bidders can guage how much more (if at all) to bid for a package in the event of losing a bid during a round. However, Rothkopf et al (1998) show that the computation of prices to make unsucessful bids competitive again is NP-hard as well.Thus, prices for items that we compute will approximate the marginal value of items. These single item prices will usually have to be approximated the since integer programming lacks a strong duality (pricing) theory Nemhauser et al, 1999. In addition, it is not always possible to construct a linear set of prices, thus at times prices for single items are non-linear and non-anonymous. The price construction we propose will be based directly on the approximating procedures for solving the WD.

Single item price constuction has been considered by Marron (1996) in the construction of an auction mechanism for a smart market for tradable pollution permits. The auction is a single round auction that consists of an allocation phase (a mixed integer program) and a pricing phase (a linear program). The pricing procedure seeks non-discriminatory prices for different pollution licenses. Rassenti et al (1982) have considered single item pricing in the context of combinatorial auctions for airport take-off and landing slots. An integer programming phase for allocation and a linear programming phase for pricing is developed as well. Experiments using the RAD method demonstrated that the presence of such prices in the context of an iterative combinatorial auction format can enable more efficient allocations, lower bidder losses and faster completion times than ones obtained from the other major combinatorial auctions to date such as AUSM Kelly et al (2000). The method that we devise will produce more accurate although not necessarily anonynmous prices that support the allocation as well as enable large scale instances to be computed.

The paper is organized as follows. In section 2 we discuss the general iterative combinatorial auction framework that motivates the two phase winner determination approximation algorithm. In section 3, we give the formulation of the winner determination problem (WD) and give the relation between the linear programming relaxation of the winner determination and single item pricing. Section 4 discusses the two phase approach for the winner determination problem that includes a method for approximating single item prices. Performance guarantees are also presented along with empirical results that were obtained by running the algorithm on a set of standard combinatorial auction test distributions.

2 General Iterative Combinatorial Auction Structure

We consider a multi-round ascending combinatorial auction. Multi-round auctions enable bidders to see and react to information in the form of current prices and winning allocations, and thus can reduce maladies such as the winners curse that occur in single round auctions Cramton et al, 1998. The major elements of a round consist of (1) a bidder evaluation/submission phase, during which other rules e.g. activity/eligibility rules may be employed, (2) a winner determination phase, and (3) a pricing phase. There are many ways to decide on the particulars of each phase. Rules are usually selected to best prevent strategic gaming, induce efficient time closure of the auction and/or achieve economic efficiency. For example, the FCC spectrum auctions employed eligibility rules that forced bidders to be active on packages ("use it or lose it") during consecutive rounds so as to prevent strategic bidding in later rounds and to speed up the auction Milgrom (2000). We assume the following rules for the auction, other rules can be incorporated without loss of generality.

(1) **Bidder Evalution/Submission**

Bidders submit bids on packages based on information provided by the auctioneer on the results of the previous round, which include the prices of packages and single items. A tentative winning bid is automatically resubmitted until it loses. Any new bid must be greater than or equal to the sum of current prices of items in the package plus a bid increment.

(2) **Winner Determination**

Tentative winning bids are determined by maximizing auctioneer's revenue subject to feasibility, i.e., no single item is allocated to two or more bidders. Current winners must pay what they bid.

(3) **Single item price determination**

Single item prices are constructed to reflect the winning bid prices.

(4) **Stopping rules**

Various stopping rules can be employed. One possible rule is to stop when there are no new bids in a round as in the FCC design. Other rules involve stopping after a certain time duration or at the discretion of the auctioneer.

The effectiveness of the above basic multi-round format in terms of achieving efficiency, bidder loss minimization, and time to closure will critically depend on the quality of the allocation (winner determination) provided by the autioneer as well as the single item pricing determination during a round. The resulting single item prices will play an important role in the drive for auction efficiency, as it provides the link between rounds of the auction. Thus, the ability to formulate new bids based on single item price information, which in turn is based on the tentative winning allocation, assumes that single item prices at each round are at the very least reflective of the tentative winning allocation.

It should be noted from a revenue stand point that it is reasonable for the auctioneer to maximize revenue when determining winners at a round since packages should go to bidders that value them the most at current prices. Also, the maximization of revenue will go towards minimizing resale activity on the part of bidders after the auction in the case where certain bidders win packages below their valuation and in turn can sell it to other bidders that value them higher but did not win during the auction Ausubel (1999).

3 The Winner Determination (WD) Problem

3.1 The Problem

Next we formulate the winner determination problem.

Let

S	set of items
V	set of all bidders
a_{ij}	1 if item i is in bidder j's bundle else 0
x_j	1 if bundle for bidder j is selected else 0
p_j	bid price for bidder j's bundle

Then the winner determination problem is

(WD)

$$\text{maximum } \sum_{j \in V} p_j x_j \tag{1}$$

$$s.t. \quad \sum_{j \in V} a_{ij} x_j \leq 1 \qquad \forall \quad i \in S \tag{2}$$

$$x_j = 0, 1 \qquad \forall \quad j \in V \tag{3}$$

The objective (1) is to select those bids j that maximize revenue with the constraint (2) that any item k can belong to at most one allocated bundle. This problem is also known as the weighted set packing problem (SP) and is NP-complete. Thus, a polynomial time algorithm for this problem is not likely to be found.

There are many approximation techniques and heuristics in the literature for the SP problem Hoffman (1999). Noted that it is not just a matter of applying well known techniques to strive for an integer feasible solution that is close to optimality. Valuable dual (single item price) information can be attained from a well formed approximation to WD and it is this connection with the dual information corresponding to the primal problem that will drive our design choice for approximation.

3.2 Linear Programming and the WD Problem

The LP relaxation of the WD is given by

(P1)

$$\max \ \sum_{j \in V} p_j x_j \tag{4}$$

$$s.t. \ \sum_{j \in V} a_{ij} x_j \leq 1 \qquad \forall \ i \in S \tag{5}$$

$$x_j \geq 0 \qquad \forall \ \ j \in V \tag{6}$$

and the dual of the relaxation is

(D1)

$$\min \ \sum_{i \in S} \pi_i \tag{7}$$

$$s.t. \ \sum_{i \in S} a_{ij} \pi_i \geq p_j \qquad \forall \ \ j \in V \tag{8}$$

$$\pi_i \geq 0 \qquad \forall \, i \in S \tag{9}$$

where π_i is the dual variable associated with the ith constraint of type (5) in P1.

Primal Complementary Slackness and Single Item Prices We aim to have single item prices such that a winning bid price is the sum of the prices of the items and a losing bid price is less than or equal to the sum of prices of the items in the packages. If a particular allocation of packages to bidders given by x^* through the solution of the WD is such that the primal complementary slackness conditions are satisfied then this pricing property is met. That is, if bundle j is allocated (i.e., $x_j^* = 1$) then it must the case that $(a_j^t \pi^* - p_j) = 0$ or $a_j^t \pi^* = p_j$which is condition that the sum of the prices of bundle j must be equal to the bid price p_j. However, this condition can not be guaranteed to be met for integer programs such as the WD problem.

One technique to deal with solving integer programs is to first solve the linear programming relaxation of the IP.

Theorem 1 *If the LP relaxation of the WD problem admits an integer solution then this solution is optimal for the WD.*

We give the proof of the following basic fact to illustrate the relationship between optimality the LP relaxation of WD and single item pricing.

Theorem 2 *If the linear programming relaxation of the WD admits an integer solution, then the corresponding optimal solution to the dual of the LP relaxation is a set of prices for single items that supports the optimal allocation.*

proof:

Let $x^* = (x_1^*, .., x_V^*)$ be the optimal solution to the LP relaxation of WD and π_i be the dual variable associated with primal row i and let $\pi^* = (\pi_1^*, ..., \pi_M^*)$ be the optimal solution to the dual of the LP relaxation, then by complementary slackness we have

(1) $x_j^* = 1$ implies that $\sum_{i \in M} a_{ij} \pi_i^* = p_j$

(2) $x_j^* = 0$ implies that $\sum_{i \in M} a_{ij} \pi_i^* > p_j$

Thus, the dual prices π_i^* act as prices for item i and thus condition (1) states that a winning bid for bundle j is such that the sum of the single item prices is equal to the winning bid price p_j and (2) states that an unsuccessful bid is such that the bid price p_j is less than the sum of the prices of the items in the bundle j. ■

Thus, in the ideal case of Theorem 2 primal (as well as dual) complementary slackness is satisfied, and so single linear item prices will exist that support the integer allocation. These single item prices will be like unique market clearing prices for the round. But, in general the solution to the LP relaxation may often contain fractional solutions and thus prices that support the (optimal) integer allocation of WD may not exist or be unique. In this case, a strategy for approximating the single tiem prices would be needed.

Single Item Price Approximation

Single item price approximation has been attempted for combinatorial auctions. In the single item approximation techniques considered by Rassenti et al (1982), Marron (1996), and Demartini et al (1999), the strategy has been to first produce an allocation of items or packages and then use these results to form a "pseudo-dual" problem from which to compute single item prices. The first of these problems is always some form of integer program and the second is a linear program.

The essential characteristic of the approaches is that an approximate dual problem is formed from the allocation provided by the winner determination. During a round, the bids are collected by the auctioneer and then the winner determination problem is supposed to be solved *exactly*. Based on the allocation provided by the WD an optimization problem is constructed that computes single item prices.

Given the bid prices $p_j's$ single item prices are constructed that try to mimic the prices that one would obtain through the primal complementary slackness conditions under the ideal conditions of Theorem 2. The set of constraints try to ensure that prices for items in a winning bundle sum up to the price of the winning bid and try to enforce that losing bid prices are less than sum of the prices of the single items.

Two important limitations in the allocation and pricing mechanisms of the "psuedo-dual" approach described above is that the integer programming problem is assumed to be able to be solved exactly which clearly is not a realistic assumption for large scale instances. Another potential problem is that the "pseudo dual" pricing program may produce prices that are not reflective of marginal values since minimization of dual prices is not addressed. It is shown in Anandalingam et al (2001) that these types of methods can produce erroneous prices and potentially mislead bidders.

Thus, one way to improve on the limitations of these methods is to (1) obtain feasible integer solutions to WD in a reasonable amount of time with revenue surplus (objective) as high as possible and (2) construct single item prices that are reasonably accurate by maintaing primal complementary slackness conditions.

4 Two Phase Approximation Algorithm for WD

We propose a two phase approximation technique to solve the winner determination problem for combinatorial auctions that maintains primal complementary slackness to approximate dual prices. The Phase I procedure will consist of a primal-dual approximation algorithm that attempts to produce an allocation with revenue as large as possible and that will satisfy primal complementary slackness. Phase II is an iterative mechanism based on the dual of the LP relaxation of WD and a greedy heuristic that attempts to improve the solution provided by Phase I. A novel feature of the method is that the use of a primal-dual strategy allows for optimality based performance guarantees for the entire two phase method. These bounds can offer very strong performance guarantees. In addition, single item price approximations are constructed by solving a restricted version of the dual of the relaxation of WD. or a variant.

4.1 Phase I

The goals of revenue maximization for the auctioneer (i.e. IP maximization) and accurate single item pricing (i.e. maintaining of primal complementary slackness) suggest the use of a non-traditional primal-dual optimization strategy based on the LP relaxation of the WD problem. Traditional primal-dual strategies are usually based on iterative methods that maintain primal and dual complementary slackness and dual feasibility and strives to attain primal feasibility Papadimitriou (1982). Instead, we strive for primal and dual feasibility and primal complementary slackness but relax dual complementary slackness Goemans et al (1997).

Note that the dual of the LP relaxation will usually be a strict upperbound to the WD problem for any feasible integer solution. The hope is that by maintaining the three optimality conditions, a solution that is near-optimal with a set of supporting prices will result as it has for many other combinatorial optimization problems Williamson (1998).

4.2 The Primal-Dual Approximation Algorithm (PD1)

We will now present with the mathematical form of the phase I algorithm.

Let	M	largest average reward per item for any bundle.
	X	be set of primal variable indices set to 1
	Y	set of primal variable indices set to 0
	$Active\_set$	set of indices of dual constraints that will be activate (set to equality)
	$Inactive\_set$	set of indices of dual constraints that will be inactive (set to inequality)
	π_i	be the dual (price) value associated with item i

q	maximum dual directional decrease index
δ_q	maximum dual directional decrease

Phase 1

$X := \emptyset$; (i.e. set $x_j = 0 \ \ \forall\ j \in V$)

$\pi_i := M; \quad \forall\ \ i \in S$

While $\exists$ an item k such that $k \notin \cup_{j \in X} S_j$ do

  $q = \arg\min_{j:k \in S_j}\{\sum_{i:i \in S_j} \pi_i - p_j\}$;

  $\delta_q = \sum_{i:i \in S_q} \pi_i - p_q$;

  $Active\_set = Active\_set \cup \{q\}$;

    if S_q is a dummy bid then

      $Inactive\_set = Inactive\_set \cup \{q\}$;

    else

      for $i \in S_q$ do

        $\pi_i = \pi_i - \frac{\partial_q}{|S_q|}$; (i.e. $M - \frac{\partial_q}{|S_q|}$)

      end

    $x_q = 1$;

    $X := X \cup \{x_q\}$;

 for S_l such that $S_l \cap S_q \neq \emptyset$ do

    $x_l = 0$;

    $Inactive\_set = Inactive\_set \cup \{l\}$;

    $Y = Y \cup \{x_l\}$;

(19) end

(20) end

We assume that we have dummy subsets that consists of a single item i for each each $i \in S$ with bid price equal to 0. When such subsets are allocated we have the corresponding dual constraints set to inactive which has the effect of enforcing non-negativity of price for unallocated items.

The idea behind Phase 1 is to construct an integer feasible solution to WD through considering the LP relaxation of WD and its dual. For linear programming problems, a solution is optimal when a solution for the primal problem is feasible and a corresponding dual feasible solution exists and dual and primal complementary slackness conditions are satisfied. The strategy of Phase 1 is to construct a primal solution that is feasible for WD implicitly during the process of creating a dual solution. In particular, the dual problem is initialized and then we search for a descending direction by iteratively reducing the set of dual variables such that at any time primal complementary slackness is attained for a constraint in the dual we set the corresponding primal variable to 1. We wish we descend from an initial dual solution rather than ascend because the dual is an upper bound to the WD problem and we wish to tighten the gap between the primal and dual solutions. In this manner, we guarantee primal feasibility and primal complementary slackness, and often times dual feasibilty as well. In the cases, when dual complementary slackness is achieved as the solutions are optimal.

Lines (1)-(2) initialize the dual variables to M and primal variables to 0 and then PD1 iterates over items line (3) A bundle containing the current item is sought such that the corresponding constraint in the dual at prices M allows the greatest decreasing search direction line (4). The dual variables in this bundle are uniformly reduced by the average directional magnitude by lines (10)-(12). Then, the corresponding bundle j is allocated i.e. $x_j = 1$ by line (13). All bundles that have elements in common are then removed line by lines (15)-(18). The PD1 terminates when all items have been iterated over.

4.3 Complexity of PD1

It is not hard to see that PD1 is a polynomial time algorithm. In particular, we have the following worst case time complexity bound. In essence, PD1 provides a rapid feasible solution to the winner determination problem.

Theorem 3 *The PD1 algorithm can be run in time complexity* $O(|S|^3|V|)$ *assuming that* $|V| \geq |S|$.

proof: (see Appendix) ■

4.4 Primal and Dual Feasibility and Single Item Pricing

We now show that PD1 alwas produces a primal feasible solution, and give the conditions when PD1 constructs a feasible solution to the dual of the relaxation of the WD problem.

Theorem 4 *The primal-dual algorithm (PD1) produces a feasible solution for the WD problem.*

proof: (see Appendix) ■

Existence and construction of feasible dual (single item) prices

Theorem 5 *PD1 produces a feasible solution to the dual of the linear programming relaxation of the WD problem if* $|S_j|M - \sum_{l=1}^{N_j}(|S_l|M - p_l) \geq p_j \quad \forall j \in Inactive$ *where* N^j *is the number of different active subsets that have elements in* S_j.

proof: (see Appendix) ■

It should be noted that only constraints that correspond to bids that were not successful (not allocated) can be infeasible as the pricing rule of PD1 is feasible for all contraints corresponding to bids that are winners for a round by the proof of above theorem.

4.5 Linear and non-linear pricing of packages

When PD1 produces a feasible solution to the dual of the relaxation of the WD problem the set of dual (item) prices will satisfy primal complementary slackness by construction. However, by Theorem 5 PD1 will not always produce a feasible set of dual prices. In such an instance, one can still construct a set of dual prices for items.

Augmented Pricing (AP) Algorithm

For all dual constraints j in D1 (7)-(9) that are violated by the pricing produced under PD1 for the WD problem perform steps (1) through (3).

(1) Let $v_j = p_j - |S_j|M - \sum_{l=1}^{N_j}(|S_l|M - p_l) \geq 0$ and $Q_j = \lceil \frac{v_j}{M} \rceil$.

(2) Now add Q_j variables $\pi_l^{v_j} \geq 0$ for $l = 1$ *to* Q_j that are to be unique to dual constraint j. These variables will correspond to additional ground items denoted by $i_l^{v_j}$.

(3) Assign $\pi_l^{v_j} = M$ for $l = 1$ *to* Q_j and let all other dual values be equal to values produced by PD1.

Let WD^0 be the original instance of the WD problem and DWD^0 the dual of WD^0 but with constraints $j \in Active$ set to equality (where the *Active* set is determined by PD1). Let WD^Q be the same instance as WD^0 , but with the new variables $\pi_l^{v_j}$ added to all constraints j of the dual of WD^0 that are infeasible under the dual prices produced by PD1. Define DWD^Q to be the dual of WD^Q with constraints j set to equality for all $j \in Active$.

Call this pricing scheme defined by steps (1) thru (3) WD^0-augmented pricing. Clearly, the WD$^0-$augmented pricing restores feasibility for constraints in D1 infeasible under PD1 since at least the violated amount v_j is augmented to each infeasible constraint j.

Then, we have shown the following.

Theorem 6 *The WD^0-augmented pricing results in a feasible solution to the dual of the LP relaxation of WD^Q.*

In addition, we have the following.

Theorem 7 *The PD1 algorithm will produce the same primal allocation for WD^0 and WD^Q and will induce WD^0- augmented pricing for WD^Q.*

proof: (see Appendix) ■

Single item price construction **Case (1)** If the dual prices defined by PD1 are feasible then the restricted dual defined by setting constraints that are indexed by *Active_set* to equality will also be feasible In this case, we can solve the restricted dual which we call Rdual to optimality. (In general Rdual problems are always defined with respect to some *Active* and *Inactive* pair of sets.) The optimal dual prices will define a set of prices that enable linear non-discriminatory pricing for packages. All bidders will be subject to the same set of prices for all packages.

Case (2) If dual prices defiend by PD1 are not feasible then solve DWD^Q to obtain the dual prices π^*. For any dual contraint j that has at least one $\pi_l^{v_j} > 0$ compute $d_j = \sum_{l=1}^{Q_j} \pi_l^{v_j}$. This quantity represents the amount by which constraint j is infeasible. In this case, theses extra single item prices will be used to form non-linear prices for packages associated with bidders whose corresponding dual constraint is violated under PD1.

4.6 Package pricing/bidding rule

Non-discrimatory pricing (*case (1)* above): For any package S_j that was bid on in the previous round and is in the tentative winning allocation, the price at the start of the next round has to be at least $\sum_{i \in S_j} \pi_i^*$ for the package S_j. If S_j was not in the tentative winning allocation, then the bidder must bid at least $\sum_{i \in S_j} \pi_i^* + \epsilon$ (where ϵ is the minimum bid increment) if package S_j is still desired.

Discriminatory pricing (*case(2)* above): For any bidder whose package S_j at bid p_j was not allocated and whose corresponding dual constraint was infeasible under PD1 must bid at least $\sum_{i \in S_j} \pi_i^* + d_j + \epsilon$ for the same package for the next round (if so desired by the bidder). The pricing mechanism in this case corresponds to a form of non-linear pricing where some bidders must bid an amount greater than the sum of the dual prices of the items in the package desired.

The use of non-linear pricing in the above procedure is similar to the use of non-linear pricing of packages in a combinatorial auction formulation of Bikchandani et al (1997). The allowance of non-anonymous nonlinear prices for packages enables a charaterization of efficiency and incentive compatibility for single round combinatorial auctions. The main point is that there must be at least some allowance for non-linear and non-anonymous pricing of packages in order to have the properties of efficiency and incentive compatibility for combinatorial auctions.

By construction of the above package pricing mechanism, price discrimination occurs for a bidder when the sum of the item prices of a package is less than the bid for that package (here we take linear prices to mean the value of optimal dual prices that correspond to real items under DWD^Q as opposed to new items that were added in the construction of DWD^Q). This is equivalent to infeasibility of the dual of the WD problem for that round at the constraint implied by the bidders package. Basically, this bidder's bid prevented the existence of a set of linear prices that would support the corresponding tentative allocation of packages as given by the approximating of the WD primal problem. This occurs although the bidder bid more than the sum of linear prices of the items in the package she did not bid enough to make anonymous linear single item pricing possible for all bidders. Thus, we require that such a bidder pay d_j in addition to her bid which is exactly the amount that, if added to her original bid, would enable linear pricing (given that the same is done for all such bidders whose constraint in the dual is infeasible under their respective original bids). Another choice is to add d_j to the bid of all bidders j that have been tentatively allocated their package, however this may discourage bidders from bidding aggressively in fear of overbidding since it will not be known apriori what the quantity d_j will be and thus possibly incur some form of winner's curse.

4.7 Phase I Optimality Conditions and Performance Bounds

We will now present some optimality conditions and optimality based performance bounds for Phase I of the method. The bounds will be very useful in the case that PD1 does not compute an optimal solution. In addition, these bounds will also provide the foundation for the performance bounds for the entire Phase I-II method.

Sufficient Optimality Condition

Theorem 8 *If a primal solution for the WD produced by PD1 covers all of the items in set S through non-dummy subsets of S, and the dual prices are feasible for the dual of the WD relaxation, then the primal solution is optimal.*

proof: (see Appendix) ■

These bounds will allow a provable assessment of closeness to the theoretical optimal value of an instance of the WD without the need to know the theoretical optimum.Such information would be valuable information for the auctioneeer as she may have a very good measure of the quality of a particular allocation. Also, for very difficult or large scale instances of the WD problem it may be impractical to wait for the optimal solution. First we need the following definition and a lemma.

Definition 1 *An algorithm is an α-approximation for a constrained minimization (maximization) problem if it returns a solution in polynomial time within a factor of $\alpha \geq 1$ ($\alpha < 1$) of the optimal value.*

Theorem 9 *The PD1 algorithm is no worse than an $\frac{1}{|S|}-$ approximation algorithm.*

proof: (see Appendix) ■

Next, we strive for a better bound .

Let U set of elements not covered by the PD1 algorithm

X set of indices of bundles allocated by PD1

X^* set of indices of bundles in the optimal allocation

OPT optimal objective value

$\varepsilon_u = \frac{|U|*M}{|X|}$

$p_j^* = p_j + \varepsilon_u \quad \forall j \in X$

k_j is such that $p_j k_j = p_j^* \quad \forall j \in X$

First we consider a performance bound for PD1 that is possible *after* the algorithm is run.

Theorem 10 *(A posteriori performance bound) The PD1 algorithm is also no worse than a* $\frac{1}{k}-$ *approximation algorithm where* $k = \max_j\{k_j\}$.

proof: (see Appendix) ■

Observation: As $|U|$ as decreases and X increases ε_u will decrease which leads uniformly to smaller $k_j's$ for $j \in X$. This is in accord with the optimality condition of PD1 which implies that the more elements covered, the better the solution. Also, all that is required to check the quality of the bound after the computational run is the number of elements covered after the run as well as M and the number of allocated subsets.

Now let $\varepsilon_u = (|S| - |S_n|) * M$,

$p_n^* = p_n + \varepsilon_u$

k_n is such that $p_n k_n = p_n^*$.

where p_n is the reward associated with S_n

Corollary 11 *(A priori performance bound) The PD1 algorithm is no worse than a* $\frac{1}{k}-$ *approximation algorithm where* $k = k_n$.

proof: (see Appendix) ■

Corollary 12 *(a priori version) The PD1 algorithm is a* $\max\{\frac{1}{|S|}, \frac{1}{k_n}\} -$ *approximation algorithm.*

proof: Follows immediately from Theorem 10 and Corollary 12

■

Corollary 13 *(a posteriori version)The PD1 algorithm is a* $\max\{\frac{1}{|S|}, \frac{1}{k}\} -$ *approximation algorithm.*

proof:*Follows immediately from Theorems 10 and 11* **QED**

If PD1 does not produce a feasible dual solution to the relaxation of the WD problem then one can construct DWD^Q and with WD$^Q-$augmented prices we can get similar performance bounds.

Let $S^* = S \cup \{i_l^{v_j}\}$ $\quad\forall$ violated constraints j in SP^0 under PD1 and $l = 1, .., Q_j$

Corollary 14 *(a priori version) The PD1 algorithm is a* $\max\{\frac{1}{|S^*|}, \frac{1}{k_n}\} -$ *approximation algorithm for WD*^{Q}*.*

proof:

Follows from Theorems 10 and Corollary 13 by letting the ground set $S = S^*$and letting $\varepsilon_u = \ (|S^*| - |S_n|) * M$. ■

Corollary 15 *(a posteriori version)The PD1 algorithm is a* $\max\{\frac{1}{|S^*|}, \frac{1}{k}\} -$ *approximation algorithm for WD*^{Q} .

proof:

Follows immediately from Theorems 10 and 11, but with U =set of elements (non-dummy) that are not covered by PD1 including the set of new elements $\{i_l^{v_j}\}$ $\forall$ violated constraints j in WD^0 under PD1 and $l = 1, .., Q_j$ ■

Currently, the best approximation guarantees for the weighted set packing algorithm has a performance ratio of $2(|S|+1)/3$ (Chandra et al 2001) where $|S|$ is the number of items in the auction. These guarantees are generally weak and computed solutions often have much better performance. The problem here is that the optimality ratios are functions of the number of items in the problem. For large scale problems with thousands or just hundreds of items these kinds of performance bounds provide almost no useful information. Other measures for gauging closeness such as the duality gap can provide too weak information as well. In general, most evaluations of optimality rely on extensive empirical testing.

The quality of the worst case performance bounds above will depend on (1) the largest average item reward and (2) the number of ground elements covered by PD1 and the number of subsets allocated. k becomes small as the number of elements covered is large and as ε_u is not too large in comparison to the smallest reward p_j among the allocated subsets.This bound can be seen to be better in many instances than the $\frac{1}{|S|} - approximation$ bound. For example, suppose that $M = 200$,$|U| = 5$, $|X| = |30|$, the smallest p_j among all allocated subsets is 700, and $n = 100$ then $k = \frac{340}{300} = 1.13$ and so $\frac{1}{k} = .954$ where as $\frac{1}{|S|} = \frac{1}{100} = .01$ and $\frac{1}{2(|S|+1)/3} = .01485$.

Observation: The level of infeasibility of dual values produced by PD1 for the dual of the SP relaxation is related to the quality of the performance bound for WD^Q as the more infeasible the dual is the more ground elements will be added to form WD^Q which results in a higher ground set cardinality S and U set cardinality.

5 Duality Gap and the Phase II Algorithm

The resulting primal solution x^* and dual solution π^* from PD1 may not be guaranteed to satisfy dual complementary slackness conditions for the LP relaxation i.e. $\pi_i^*(a_i x_i^* - b_i) \neq 0$ is possible for many instances of the WD problem. This implies that there will be a difference between the value of the WD solution $p^t x^*$ and value of $b\pi$. By the weak duality theorem of linear programming then $p^t x^* < b\pi$.

We develop a phase II algorithm that will iteratively refine the primal-dual solution pair (x^*, π^*) produced by PD1 so as to reduce the gap $b\pi^* - p^t x^*$ and to try to maintain the three optimality conditions that PD1 tries to uphold i.e. primal-dual feasibility and primal complementary slackness.

5.1 Phase II Algorithm (Active Set Strategy)

Phase II consists of two components. The *first component* of phase II will solve the dual of the LP relaxation to get a set of dual prices. The sum of these prices will be the the smallest upper bound possible on the revenue that the auctioneer can collect. We then examine the constraints that are active (binding) and identify those active constraints that were not active under the solution provided by PD1. The associated packages of such constraints are candidates to select for improvement. Only those packages that improve the objective function will be selected.The first component of phase II can be seen to be an active set method where the dual of the WD relaxation is solved to find potential entering columns among newly discovered active constraints in the dual with respect to the current PD1 solution. The restriction to active constraints ensures that only columns (bids) that satisfy primal complementary slackness will be entered. Once the most profitable set of columns has been found we update the *Active* and *Inactive* sets.

The *second component* of phase II consists of a search among unallocated bids which contain some of the allocated items produced by the first component, and has a larger number of items than the total number of items in all the bids that intersect with it. This second phase is motivated by the scenario where PD1 and the the first component of phase II is sub-optimal due to the fact that larger bids are not favored in general due to the fact that shorter bids will have more likelihood of having a larger reward per item ratio.

x	feasible primal solution produced from the PD1 algorithm
π_{PD1}	dual solution produced by PD1
$Active\_set$	dual constraints that are active as determined by PD1
$Inactive\_set$	dual constraints that are not active in as determine by PD1
π_d	optimal solution to the dual of LP relaxation of WD
$Superactive^0$	dual constraints that are active under π_d but not active under π_{PD1}
$M_{New\_Active\_set}$	ground items allocated after running component 1 of phase 2
$Dual\_active$	active constraints in the dual at the optimal dual solution
$Super\_set^0 =$	$\{S_{\sigma(j)} \mid j \in 1, .., \lvert Superactive^0 \rvert\}$ where $\sigma(j) =$ index of jth bid in the $Superactive^0$ set
$Greater\_set_i$	index set of unallocated bids that contains element i and that have greater reward and larger number of elements than the total number in the current allocated package that contains i and all other allocated packages that intersect with such unallocated bids.

PHASE II

(0) Solve the dual of LP relaxation of WD

(1) If $Superactive^0 = \phi$, then STOP

ELSE

(Component 1)

(2) For $j = 1, .., |Superactive^0|$ do

(3) Compute $best\_subset_j = \arg\max_{\substack{S \subseteq Active \cup Dual\_active \\ where\ S_{\sigma(j)} \in S \text{ and } S_a \cap S_b = \phi \ \ \forall S_a, S_b \in S}} \{\sum_{i \in S} p_i\};$

(4) Let $V_j = \sum_{i \in best\_subset_j} p_i;$

(5) if $V_j > Current\_best\_value$ then

$Current\_best\_value = V_j$;

$Current\_best\_subset = best\_subset_j$;

End

(6) $New\_Active\_set = Current\_best\_subset$;

(Component 2)

(7) For $i = 1, .., |M_{New\_Active\_set}|$ do

(8) Compute the set $Greater\_set_i$

(9) Compute

$best\_subset_i^2 = \arg\max_{\substack{S \subseteq New\_Active\_set \cup Greater\_set_i \\ where\ S_a \cap S_b = \phi\ \forall S_a, S_b \in S}} \{\sum_{k \in S} p_k\}$;

(10) Let $V_i = \sum_{k \in best\_subset_i^2} p_k$;

(11) if $V_i > Current\_best\_value$ then

$Current\_best\_value = V_i$;

$Current\_best\_subset^2 = best\_subset_i^2$;

End

(12) $New\_Allocation = Current\_best\_subset$;

Component 1 begins by solving the dual of the relaxation of WD, line (0), if there are no constraints that are binding in the dual that are not binding after Phase 1 then there are no new bids that can enter through component 1. If $Superactive^0$ is not empty then the **for** loop that starts on line (2) will enter each bid in $Superactive^0$ and then in line (3) compute the set of bids, S, in the allocation from PD1 i.e. those bids in $Active$ that are non-conflicting with this bid as well as a set of non-conflicting bids from $Dual\_active$ (provided of course that any of these bids are also disjoint with those bids selected from $Active$). If the revenue associated with S (computed in line(4)) is higher than the previous allocation then this set of bids is the tentative best allocation. Then, the process repeats, the next superactive bid is inserted and the best possible allocation that includes this bid is constructed. Component 1 stops after all bids in $Superactive^0$ have been tried in constructing an allocation.

Component 2 starts by considering an item i on line (7) that is part of the current allocation after component 1, i.e. $i \in M_{New\_Active\_set}$. Then, the set of all unallocated bids that have greater bid values than the allocated bundle that contains item i is to be identified on line (8) by computing $Greater\_set_i$. Then, line (9) attempts to find a non-conflicting set of bundles that tries to incorporate a bid from $Greater\_set_i$ and has higher revenue. This is repeated for every item $i \in M_{New\_Active\_set}$. This component is motivated by the situation where PD1 and component 1 are sub-optimal, because the allocation may have a bias toward allocating bids with high reward per items. Bids with higher bid value and with more items may improve the allocation is there is sufficient disjointness with other bundles.

5.2 Assessing Optimality at Termination of Phase I-II Method

Using the optimality bounds derived earlier it is possible at the end of the Phase I-II method to assess how close the resulting solution may be to optimality. Most of the exhaustive search based algorithms for the winner determination are not able to ensure any kind of performance guarantee at premature termination (except through duality gap measures), although it must be mentioned that the method of Sandholm (1999) is monotonically increasing in objective value as a function of time. We also address the issue of single item price construction at this point as well.

Case(1) Components 1 and 2 of Phase II do not improve the solution from PD1 and the corresponding restricted dual implied by the current allocation is feasible. From Corollary 13, the solution is at least $\max\{\frac{1}{|S|}, \frac{1}{k}\}$ percent of optimal and the single item prices are approximated by the optimal dual prices from the restricted dual. If the restricted dual (i.e. the dual where constaints are binding if its index is in the *Active* (or binding set) else constraints are left unbinding if index is in *Inactive* set) is not feasible, then DWD^Q has some dummy variables and so the solution is at least $\max\{\frac{1}{|S^*|}, \frac{1}{k}\}$ percent by corollary 15. The single item prices are in this case given by the optimal solution to DWD^Q.

case (2) Component 1 improves the solution from PD1, but not component 2. Then, the dual optimal solution of WD supports the current primal allocation with primal and dual feasibility and with primal complementary slackess holding since allocated packages are associated with tight (active) dual constraints only by construction. Thus, the optimality bounds from corollary 13 still hold. The single item prices in this case are just the prices from the optimal solution of the dual of the WD relaxation.

case (3) Component 2 improved the solution from PD1 or component 1. Then, primal complementary slackness may be violated since a new bundle may be allocated whose corresponding dual constraint is not active or will not be active in the restricted dual problem that corresponds to the most current improved allocation. This leads to two cases.

case (a) If the restricted dual associated with the *New_ Allocation* set at the end of component 2 is feasible, then the optimal solution to this program gives a set of linear prices. For this situation, the optimality bounds of corollary 13 will hold.

case (b) If the restricted dual correspoding to the final allocation is not feasible, then dual feasibility and primal complementary slackness conditions may be restored for the current allocation for the purposes of obtaining the same type of optimality guarantee as in the feasible case by constructing a modified primal and dual problem that sets all dual constraints that correspond to allocated packages to equality and adds unique variables to those constraints that were set to be active under PD1 or component 1 of Phase II, but now correspond to bundles that are not allocated by component 2.

The strategy is to create a modified primal and dual problem so that PD1 will produce the same allocation produced after component 2 of phase II in this case. In this manner, we may exploit the optimality bounds associated with PD1. Let S^{fixed} be the set of indices of bundles that remain allocated after PD1 and after component 2 of Phase II has run. Let S^{new} be the set of indices of packages that were allocated after component 2, but not under PD1. We must ensure that these bundles retain the highest average reward per item among the bundles that were allocated under PD1 that contain any of these items. The following construction, denoted as $Fitted\_PD1$, provides the mechanism through which this is accomplished.

Fitted_PD1

$item\_index = 1;$

$i = 1;$

$item\_set = M;$

while $item\_set \neq \phi$ **do**

If $i \in item\_set$ belongs to a bundle in S^{new}

denote the package by $S_{new}^{\sigma(item\_index)}$ and $\sigma(item\_index) = i$ then

(1) $item\_index = item\_index + 1;$

(2) (a) Find all other bundles that contain item i

(b) For each constraint in the dual of LP relaxation of WD that corresponds to a bundle in part (a) add the fewest possible dual variables (dummy items) to each constraint that are to be unique to each constraint so that the average reward per item (by dividing the reward by the total number of variables (items) in the modified constraint) is less than or equal to the average reward of package $S_{new}^{\sigma(item\_index)}$,

(3) Remove all items $k \in S_{new}^{\sigma(item\_index)}$ from $item\_set$

else

Remove item i from $item\_set$;

$i = i + 1;$

end **while** loop

Let S^{fixed} = set of indices of bundles that were allocated before and after component 2 of phase II from the point of last improvement (i.e. after PD1) and S_j^{fixed} be the jth such bundle

For $h = 1, .., |S^{fixed}|$ **do**

Let item i^h = item in the h^{th} bundle in S^{fixed} that was considered in the **while** loop on line (3) in PD1 during which S_j^{fixed} was allocated.

Denote the modified dual constructed by the procedure above as $fitted\_dual$ and the associated primal binary integer program as $fitted\_primal$. Also, let the total number of new items added in the above procedure as N_{item_total}.

At this point modify the PD1 algorithm **while** loop statement on line (3) so items are ordered first by $\sigma(l)$ for $l = 1, .., item\_index$, and then by the items i^h for $h = 1, .., |S^{fixed}|$. (note: we keep the original highest average reward per item M that was used in PD1 for intialization of prices in $fitted\_PD1$)

Theorem 16 *Suppose that component 2 of Phase II generates a solution that improves upon a solution generated by PD1 for an instance of WD. Then, the $fitted\_primal$ under the $fitted\_PD1$ algorithm will also produce the solution x^1.*

Now we can use the optimality bounds associated with PD1 (through $fitted\_PD1$) to derive performance bounds in this case. If the restricted dual implied by the *Active* and *Inactive* sets that result by applying $fitted\_PD1$ on the $fitted\_primal$ is feasible then the optimality performance bounds of corollary 14 apply. The single item prices are defined by the optimal solution to the restricted dual.

If not, then one can compute the WD^Q and DWD^Q corresponding to the $fitted\_primal$ and $fitted\_dual$, respectively, and then use the performance bounds from corollary 16 with the appropriate modification (augmentation) of the ground item cardinality set. Single item pricing now is non-linear as decribed in section 4.1.2 and is based on the optimal solution of DWD^Q corresponding to the $fitted\_dual$. Note that in both situations, the number of ground elements must be increased by at least by N_{item_total} elements.

Note: It is the case that the above procedure will generalize. One can guage some level of optimality quality of a solution provided by any set packing method (heuristic or exact) through the "lens" of the PD1 method. It may be the case that such optimality measures can provide better information about closeness to optimality than measures such as the duality gap or weak bounds e.g. $\frac{1}{|S|}$.

5.3 Phase I-II optimality for linear cases of WD

After solving the dual of the linear relaxation of WD one can easily separately test if the solution is optimal for the WD problem. However, this test is uncessary as the first component of Phase II will stop with the optimal solution if the relaxation of WD is integral.

Proposition 17 *If PD1 does not return an optimal solution and A^* is Totally Unimodular (TU) or the LP relaxation of the WD problem is equivalent to WD, then Phase I-II method with use of component only will find the optimal solution with the correct non-discriminatory and anonymous prices for items.*

Thus, the Phase I-II method can identify a large class of bids that admit linear solutions see (Nisan 2000). The benefit here is that the auctioneer dosn't have to assume that a linear bid structure will hold for bidders apriori which does away with the need to develop specialized polynomial time algorithms for some of these cases see (Rothkopf 1998).

6 Phase I-II Computational Results

We present some results based on running the phase I-II approximation method on a standard test suite developed for combinatorial auctions see (Sandholm 1999) and (Fujishima et al

1999) as well as on a new distribution that was created to model budget constrained bidders. The running times were obtained on a Pentium III 800 Mhz machine with 265 MB of main memory. As a benchmark we tested the same problems instances using the CPLEX MIP solver (version 7.0). In prior tests (see Andersson et al 2000),the Cplex MIP 6.5 solver was seen to be a very effective general purpose winner determination algorithm across the standard test distributions and in most cases out performed the current methods. It would be very likely that the Cplex MIP 7.0 version would be even more effective.

Experimental results for a given test size will consist of the average running time of the Phase I-II method and Cplex on the test instances as well as number of times optimal solutions were found by Phase I-II and average optimality. We decide to benchmark our approximate algorithm against the total time until Cplex finds a verifiable optimal solution. Based on our experiments running Cplex MIP 7.0 it was the case that significantly improved solutions were found at many different points in time in the duration of search and we could not reasonably conclude that most of the quality of a solution was found in the time that it took for our algorithm to run. In addition, other comments will be given that point to some additional aspects of particular test runs. In particular we define the following:

Let M = number of items in the auction

N = the number of bids (in a round)

Exp = experiment (set of problem instances drawn from the same distribution

and with same (N,M values))

trials = number of problem instances in an experiment

Ph I-II Quality = average percent of optimal solution that the Phase I-II method

achieves on an experiment

Ph II Improve = number of times in an experiment that Phase II improved on the

solution from Phase I

RT Ph I-II = Average run time of the Phase I-II method on an experiment

RT Cplex MIP 7.0 =average run time of Cplex MIP 7.0 on an experiment (time represents time to reach optimal solution)

Anonymous prices = number of times a set of linear (anonymous) prices was able to be constructed by the Phase I-II method

(1) Weighted Random Distribution

For each bid i, pick the number of items to be in the bid B_i randomly from 1,..,M. Then, without replacement choose that many items. The price for a bid is a randomly chosen from [1,.., B_i].

Table 4:

Exp	M	N	trials	Ph I-II Quality	Ph II Improve	RT Ph I-II (s)	RT Cplex MIP 7.0	Anonymous prices
1	5	30	40	96.15%	35 out of 40	$<$ 0.01	$<$ 0.01	40 out 40
2	10	100	40	97.30%	33 out of 40	$<$0.01	$<$0.01	40 out of 40
3	20	200	40	99.82%	40 out of 40	$<$ 0.01	$<$ 0.01	40 out of 40
4	30	300	40	98.38 %	37 out of 40	0.01	0.029	40 out of 40
5	40	400	40	99.94%	38 out of 40	0.015	0.04	40 out of 40
6	400	2000	40	99.98%	35 out of 40	15.30	8.1	40 out of 40
7	100	10,000	40	99.79%	38 out of 40	10.71	29.35	40 out of 40
8	100	100,000	30	$>$98.60%	26 out of 30	167.12	N/A*	30 out of 30
9	1000	5000	30	99.74%	27 out of 30	102.5	88.77	30 out of 30
10	2000	6000	20	$>$98.035*	18 out of 20	746.83	N/A*	20 out of 20

The results indicate that this distribution was not hard for Cplex 7.0 or the Phase I-II method which is accordance with the findings of (Andersson 2000). For the larger scale instances however, the branch and bound method of Cplex required too much memory and stopped prematurely. The largest test instances reported in the literature usually contained no more than 2000 bids and 400 items, but the phase I-II method easily scaled to handle up to 100,000 bids and up to 2000 items with very little loss of quality.

(2) Decay Distribution

For each bid, select one random item. Then repeatedly add a new random item with probability α until an item is not added or the bid includes all M items. The price of a bid is a number randomly selected in [0,.., # items in bid]. (We select $\alpha = .75$ as this parameter value was observed to result in challenging test instances for this distribution see (Sandholm 2000)).

Table 5:

Exp	M	N	trials	Ph I-II Quality	Ph II Improve	RT Ph I-II (s)	RT Cplex MIP 7.0	Anonymous prices
1	5	30	40	99.65%	40 out of 40	< 0.01	< 0.01	40 out of 40
2	10	100	40	99.72%	36 out of 40	< 0.01	< 0.01	40 out of 40
3	20	200	40	99.80%	29 out of 30	< 0.01	< 0.01	40 out of 40
4	30	300	40	97.74%	33 out of 40	< 0.01	< 0.001	40 out of 40
5	40	400	40	99.25%	32 out of 40	0.035	0.162	40 out of 40
6	100	1000	40	98.42%	40 out of 40	2.12	11.54	40 out of 40
7	300	3,000	10	97.83%	10 out of 10	29.75	920.74	10 out of 10
8	400	4,000	8	99.125%	8 out of 8	109.25	3375.20	8 out of 8
9	200	10,000	5	99.34%	5 out of 5	302.36	4974.50	5 out of 5
10	200	20,000	10	>99.68*	10 out of 10	945.77	n/a*	10 out of 10

In this set of experiments we observed that the for the larger instances our method was up to 2 orders of magnitude faster than Cplex.

(3) Binomial Distribution

In this distribution, a bid requesting m of M commodities has probability distribution $f(m) = \binom{M}{m} p^M\ (1-p)^{M-m}$. The valuation of a bid is drawn from 1 to 1000, multiplied by m. (note: that the probability parameter p for an experiment is indicated in the Exp column)

Table6:

Exp	M	N	trials	Ph I-II Quality	Ph II Improve	RT Ph I-II (s)	RT Cplex MIP 7.0	Anonymous prices
1, p=.2	5	30	40	98.97%	40 out of 40	$<$ 0.01	$<$ 0.01	40 out of 40
2, p=.2	10	100	40	96.62%	23 out of 40	$<$ 0.01	$<$ 0.01	23 out of 40
3, p=.2	20	200	40	94.17%	15 out of 40	$<$ 0.01	$<$ 0.01	15 out of 40
4, p=.2	20	10,000	40	99.98%	25 out of 40	$<$ 0.01	$<$ 0.02	25 out of 40
5 , p=.5	500	2,000	5	98.61%	5 out of 5	19.80	9296.47	5 out of 5
6, p=.2	1000	1,000	8	99.93%	8 out of 8	24.38	4,880	8 out of 8
7, p=.2	1000	2,000	3	97.71%	3 out of 3	73	34,710	3 out of 3
8, p=.2	1000	10,000	3	99.29%	3 out of 3	592	52,064	3 out of 3
9, p=.5	500	10,000	2	99.85%	2 out of 2	249	494,104	2 out of 2
10, p=.2	1000	5000	2	100%	2 out of 2	274	463,187	2 out of 2

This distribution for Cplex was very difficult and the phase I-II method was up to 3 orders of magnitude faster with very little drop off in quality.

(4) **Budget Bounded Distribution**

For each bid i, randomly select a bound B_i on the number of elements the bid can have from [1,..,M].. Then select randomly the number of items I in the bid from [1,..,B_i]. Now select randomly without replacement I items for the bid from the ground item set M .The price for the bid is a number randomly selected from [0,..I]. This distribution is a generalization of the uniform and bounded distributions that appears in (Sandholm). We believe that this distribution is more realistic since it is based on the assumption that bidders are budget constrained and hence will limit the size of a package accordingly. Also, it can reflect any deposit requirement placed on bidders like in the FCC SAA spectrum auctions where the deposit limits the number of items bidders can place bids on.

Table 7:

Exp	M	N	trials	Ph I-II Quality	Ph II Improve	RT Ph I-II (s)	RT Cplex MIP 7.0	Anonymous prices
1	5	30	40	98%	40 out of 40	< 0.01	< 0.01	40 out of 40
2	10	100	40	98.55%	40 out of 40	< 0.01	< 0.01	40 out of 40
3	200	30,000	15	95.68%	15 out of 15	71.25	1209.67	15 out of 15
4	300	3000	15	97.81%	15 out of 15	23	1205.75	15 out of 15
5	400	4000	10	97.15%	10 out of 10	64	2044	10 out of 10
6	600	15,000	5	99.41%	5 out of 5	1880	10538	5 out of 5
7	200	100,000	20	97.80%	20 out 20	140	n/a*	20 out of 20
8	1000	2000	5	99.85%	5 out of 5	70	3840	5 out of 5
9	300	150,000	20	98.76%	20 out of 20	115	n/a*	20 out of 20

*indicates that memory was exhausted and that the optimal solution quality was computed by assuming there were at least $|M|$ single item packages with maximum possible valuation

The phase I-II method was able to scale to problem sizes for which Cplex exploded in terms of using memory. Solution quality was never below 97% for the larger scale instances and was up to one order of magnitude faster than Cplex for several of the experiments.

7 Concluding Remarks

We have presented a mathematical programming based method to approximate the winner determination problem for combinatorial auctions. The use of linear programming duality (through the primal -dual approximation framework) as a guide to approximate the NP-complete problem has shown to be quite effective, especially for large scale instances of the problem. The advantages in using an effective approximation algorithm is that the running time is primarily a function of the problem size whereas for branch and bound methods the structure of the problem also greatly impinges on the run time. Also, the performance bounds may be used to test the validity of the phase I-II method on extremely large scale instances of combinatorial auctions where waiting for Cplex or other search based methods to finish may not be practically feasible.

In addition, the method that we have proposed enables a direct computing of single item prices that reflects the primal allocation. These prices are in every possible construction derived from a dual problem that mimimizes these quantities, and as such should be very suitable as price information for bidders at the start of the next round.

It is possible to use the Phase I-II method in other ways related to winner determination. If there is not a concern for single item prices, then one can use the procedures presented to quickly find a good primal allocation with performance guarantees. From this point, one can "hot start" many other types of algorithms like column generation, evolutionary algorithms, or general branch-and-bound methods. The computational tests have shown that the Phase I-II method scales very well as noted by the performance of the method on the binomial distribution, which is considered as a very promising test distribution DeVries et al (2001).

A possible future research direction could involve incorporating the Phase I-II method in a search based optimization framework. Since the most expensive portion of the Phase I-II method is in the computing of a linear program, the method may be embedded in a linear programming branch and bound framework where some sort of memoization could be used to keep track of prices of variables (packages) that are fixed. In addition, the optimality bounds also present additional information in addition to the lower and upper bound information provided by LP duality that may be used to prune parts of the branch-and-bound tree. In situations where the duality gap is quite large, these performance bounds may provide much better information for branch and bound.

Appendix

We present the proofs of the theorems, corollaries, and lemmas that were stated without proof in the paper.

Proof of Theorem 3:

Line (2) takes $|S|$ operations. The while loop on line (3) in the worst case must be executed $|S|$ times (without loss of generality assume that item k is the kth item in S to be covered by the while loop).For each while loop iteration to cover ground item k, then line (4) in the worst case will require a search of $|V| - k + 1$ subsets in order to determine $q = \arg\min_{j:k \in S_j}\{\sum_{i:i \in S_j} \pi_i - p_j\}$, and for each of these subsets that contain item k then at most $|S| + 1 + 1$ arithmetic operations are required, $|S|$ additions and one subraction plus one comparison to check if evaluation of $\sum_{i:i \in S_j} \pi_i - p_j$ is the current best minimum quantity. The dummy subset check on line (7) is done at most $|S|$ times. The dual price updating for π_i in the for loop on line (8) is done at most $|S|$ times. The condition for iteration on the **for** loop on line (15) can be checked while performing the computation for q on line (4) . The updating of the sets $Inactive$ and $Active$ require $|V|$ updates and the sets X, Y can be obtained from these sets. Thus, the overall time complexity is no worse than $O(|S| * \sum_{k=1}^{|S|}(|V| - k + 1) * (|S| + 2) + 3|S| + |V|) = O(|S| * (\sum_{k=1}^{|S|}(|V| + 1) - \sum_{k=1}^{|S|} k) * (|S| + 2) +$

$3|S| + |V|) = O(|S| * (|S| * (|V| + 1) - \frac{s(s+1)}{2}) * (|S| + 2) + 3|S| + |V|) = O(2|S|^3|V| - |S|^3 - |S|^4 +$

$4|S|^2|V| + 2|S|^2 + 3|S| + |V|)$ ■

Proof of Theorem 4:

Whenever a subset S_j is allocated i.e. $x_j = 1$ in the while loop on line(13) by construction subsets S_k that conflict with it are assigned to 0 i.e. $x_k = 0 \ \forall k$ such that $S_j \cap S_k$ by the for loop that starts on line (15) ■

Proof of Theorem 5:

For ever iteration of the while loop of PD1 a subset S_q is selected to cover the element k that is to be covered in the current iteration.

Case (1) If S_q is a dummy subset, then the associated dual constraint $\{\sum_{i \in S_q} a_{ij}\pi_i \geq p_q\}$ $= \{\pi_k \geq p_q = 0\}$ is set to inactive status which ensures that the dual price of the single item k is non-negative i.e. $\pi_k \geq 0$.

Case (2) If S_q is a non-dummy subset, then its associated dual constraint is set to active status i.e. $\sum_{i \in S_q} a_{ij}\pi_i = p_q$. Furthermore ,we have by the item pricing mechanism **for** loop on line (11) that $\pi_i = M - \frac{\partial_q}{|S_q|}\ \forall i \in S_q$. But since, $M =$ highest average price per item among all average item prices for subsets, then $M - \frac{p_q}{|S_q|} \geq 0$ and $\pi_i = M - \frac{\partial_q}{|S_q|} = |S_q|(M - \frac{p_q}{|S_q|}) \geq 0$ since $|S_q| > 0$ and by definition of ∂_q. Thus, the non-negativity of these dual prices are clearly satisfied.

In both cases, all dual constraints associated with subsets S_l with an item in common with S_q are kept in the original inequality form (inactive) i.e. $\sum_{i \in S_l} a_{ij}\pi_i \geq p_l\ \forall l$ such that $S_l \cap S_q \neq \emptyset$. Now as the while loop iterates until all elements are covered either through a dummy or non-dummy subset then it is clear that (1) all non-negativity requirements are satisfied, (2) all dual constraints of the LP relaxation of WD will be generated in its original inequality form (inactive) or is equality (active) form.

Next, we show that the pricing mechanism on line (11) is feasible for all constraints in the *Active* set in the restricted version of the dual of the LP relaxation of WD constructed by PD1. For $j \in Active\_set$ we have by the results of the for loop on line (10) that for each $i \in S_j$, $\pi_i = M - \frac{\partial_j}{|S_j|}$ and thus we have the following

$$\sum_{i:i \in S_j} \pi_i - p_j = \sum_{i:i \in S_j}(M - \frac{\partial_j}{|S_j|}) - p_j = \sum_{i:i \in S_j} M - \sum_{i:i \in S_j} \frac{\partial_j}{|S_j|} - p_j$$

$$= |S_j|M - \sum_{i:i \in S_j}(\frac{(\sum_{i:i \in S_j} M) - p_j}{|S_j|}) - p_j = |S_j|M - \sum_{i:i \in S_j}(\frac{|S_j|M - p_j}{|S_j|}) - p_j$$

$$= |S_j|M - |S_j|(M - \frac{p_j}{|S_j|}) - p_j = |S_j|M - |S_j|M + p_j - p_j = 0$$

Now consider the case for $j \in Inactive\_set$

case(1) the jth dual constaint does not contain any variables corresponding to items belonging to a bundle whose corresponding dual constaint $k \in Active\_set$, then for all $i \in S_j$, $\pi_i = M$ and so clearly $\sum_{i:i \in S_j} \pi_i = |S_j|M \geq p_j$ by definition of M.

case(2) suppose that the jth constraint contains elements from N^j different active constraints. Let S_l^* be that subset of items from active bundle S_l for $l = 1, ..., N_1$ that appear in the jth constraint. Also let $D =$ the number of items that belong uniquely to the jth constraint. We wish to show under what conditions that $\sum_{i:i \in S_j} \pi_i \geq p_j$ holds.

Now $\sum_{i:i \in S_j} \pi_i = \sum_{l=1}^{N^j} \sum_{i:i \in S_l^*} \pi_i + \sum_{d=1}^{D} \pi_d$

$= \sum_{l=1}^{N^j} |S_l^*|(M - \frac{\partial_l}{|S_l^*|}) + DM \qquad (*)$ since each dual constraint corresponding to S_l^* is active and thus follows from line (11) of PD1 and the second term follows since $\pi_d = M$ $\forall d$

but $(*)$ $= \sum_{l=1}^{N^j} |S_l^*|M - \sum_{l=1}^{N^j} |S_l^*| \frac{\partial_l}{|S_l^*|} + DM$

$$= \sum_{l=1}^{N^j} |S_l^*|M + DM - \sum_{l=1}^{N^j} \partial_l = |S_j|M - \sum_{l=1}^{N^j} \partial_l$$

(since $\sum_{l=1}^{N^j} |S_l^*| + D = |S_j|$)

$= |S_j|M - \sum_{l=1}^{N^j} (\sum_{i:i \in S_l} \pi_i - p_l)$ (be definition of ∂_l)

$= |S_j|M - \sum_{l=1}^{N^j} (|S_l|M - p_l)$

so $\sum_{i:i \in S_j} \pi_i \geq p_j$ holds when $|S_j|M - \sum_{l=1}^{N^j} (|S_l|M - p_l) \geq p_j$ ■

Proof of Theorem 8:

Suppose all elements of S are covered and they are from subsets that are not dummy subsets. Let X = set of indices of subsets S_j allocated in the solution by PD1. For every $j \in X$, $p_j = \sum_{i \in S_j} \pi_i$ and $\sum_{j \in X} p_j = \sum_{j \in X} \sum_{i \in S_j} \pi_i = \sum_{i \in S} \pi_i$ since by assumption all elements are covered. Now PD1 produces a feasible integer solution for SP by Theorem 3.1 (and hence a feasible solution to the linear programming relaxation of WD) and a feasible solution for the dual of the linear programming relaxation of SP by Theorem 3.2. Thus, by the strong duality theorem of linear programming the primal solution produced is optimal for WD. ■

Lemma 18 *The PD1 algorithm will allocate at least one subset of S whose average item price is M.*

proof:

Let $S^M = \{S_n | \frac{p_n}{|S_n|} = M\}$, then $\sum_{i:i \in S_n} \pi_i - p_n = 0 \ \forall S_n \in S^M$.

case (1) Suppose $S_n \in S^M$ and that all items $i \in S_n$ are unique to it i.e. $S_n \cap S_r = \emptyset$ $\forall r \neq n$, then by the while loop of PD1 on line (3) at least one item $i \in S_n$ will be such that $i = k$ at some iteration and so this subset will be allocated i.e. $n \in Active$.

case(2) Suppose that all subsets in S^M have at least one item in common with some subset not in S^M and now assume that all subsets $S_n \in S^M$ are not in the allocation produced by PD1, then for some other allocated bundle S_l such that $\exists\ t$ with $t \in S_n \cap S_l$ for some n and without loss of generality $t = k$ at the start of the while loop line(3) where S_l gets allocated (there must be such a t and S_l else all elements in S_n are unique to it and thus must get allocated) it must have been the case that l was equal to $q = \arg\min_{j:k \in S_j}\{\sum_{i:i \in S_j} \pi_i - p_j\}$ this implies that $\sum_{i:i \in S_l} \pi_i - p_l < 0$ since $\sum_{i:i \in S_n} \pi_i - p_n = 0$ which results in an infeasible solution to the dual which contradicts that PD1 always produces a dual feasible solution ■

Proof of Theorem 10:

Let OPT be the optimal objective value and X^* the corresponding optimal allocation for an instance of the winner determination and let $\pi = \{\pi_1, ..., \pi_{|S|}\}$ be the feasible solution to the dual of the LP relaxation of the WD problem produced by PD1, then

$$OPT = \sum_{j \in X^*} p_j^* \leq \sum_{i \in S} \pi_i \leq \sum_{j=1}^{|S|} M = |S| M$$

$$\leq |S| \sum_{j \in X} p_j \qquad \text{so} \quad \frac{OPT}{|S|} \leq \sum_{j \in X} p_j$$

The first inequality follows from the weak duality theorem of linear programming, the second inequality follows from the fact that PD1produces dual prices $\pi_i = M$ (for items not part of any allocated bundle) or $\pi_i = M - \frac{\partial_q}{|S_q|} \geq 0$ (by line (11) of the algorithm) , the last inequality follows from the fact that the subset with average item reward M will be allocated by the algorithm by the lemma above and thus the corresponding subset reward p_j will be greater than or equal to M.

Proof of Theorem 11:

Let U = set of elements not covered by the PD1 algorithm

X = set of indices of bundles allocated by PD1

X^* = set of indices of bundles in the optimal allocation

OPT = optimal objective value

$\varepsilon_u = \frac{|U| * M}{|X|}$

$p_j^* = p_j + \varepsilon_u \quad \forall j \in X$

k_j is such that $p_j k_j = p_j^* \quad \forall j \in X$

then, $\sum_{j \in X} p_j^* = \sum_{j \in X}(p_j + \varepsilon_u) = \sum_{j \in X} \sum_{i \in S_j} \pi_i + \sum_{k=1}^{|U|} M \geq$

$\sum_{j \in X} \sum_{i \in S_j} \pi_i + \sum_{i \in U} \pi_i = \sum_{i \in S} \pi_i \geq \sum_{j \in X^*} p_j = OPT \quad (0)$

The first equality follows from the definition of p_j^* and the second equality follows from the fact that for all allocated subsets S_j PD1 enforces primal complementary slackness for the corresponding dual constaint i.e. $p_j = \sum_{i \in S_j} \pi_i \; \forall j \in X$. The first inequality follows from the fact that $M \geq \pi_i \; \forall i$ and the second inequality follows from weak duality of linear programming. Also, we have that $\sum_{j \in X} p_j k = k \sum_{j \in X} p_j \geq \sum_{j \in X} p_j^* = \sum_{j \in X} p_j k_j$ since $k = \max_j \{k_j\}$, so by (0) $\sum_{j \in X} p_j \geq \frac{1}{k} OPT$ ■

Now let $\varepsilon_u = (|S| - |S_n|) * M$,

$p_n^* = p_n + \varepsilon_u$

k_n is such that $p_n k_n = p_n^*$.

where p_n is the reward associated with S_n

Proof of Corollary 12:

From lemma 3.4 we have that at least one subset S_n with average reward per item M will be allocated. Then clearly the PD1 algorithm will allocate at least $|S_n|$ elements . In the worst case, we will have only $|S_n|$ elements allocated and $|S| - |S_n|$ unallocated elements. So we will have $X = \{n\}$ and $\varepsilon_u = (|S| - |S_n|) * M$, $p_n^* = p_n + \varepsilon_u$, and k_n is such that $p_n k_n = p_n^*$. Now the analysis proceeds identically as in the previous a posteriori case Theorem 3.6 ■

Proof of Corollary 13: Follows immediately from Theorem 3.5 and Corollary 3.7

■

Corollary 19 *(a posteriori version)The PD1 algorithm is a* $\max\{\frac{1}{|S|}, \frac{1}{k}\} -$ *approximation algorithm.*

proof:*Follows immediately from theorems 3.5 and 3.6.* **QED**

Proof of Theorem 16:

The $fitted\_PD1$ will while loop on line (3) starting with the items defined by $\sigma(k)$ for $k = 1, .., item\_index$.

case(1) suppose that a package that contains an item defined by $\sigma(k)$ is up for allocation at start of **while** loop on line (3) of $fitted\_PD1$ for some k, let a package y that contains item $\sigma(k)$ be denoted by $S_y^{\sigma(k)}$, then that package denoted by $S_q^{\sigma(k)} \in S^g$ will be allocated since it has the highest average reward per item by contruction i.e. $q = \arg\min_{j:\sigma(k)\in S_j}\{\sum_{i:i\in S_j}\pi_i - p_j\}$ for this while loop iteration.

case(2) the **while** loop is now trying to allocate a package that includes an item i^h for some h from 1 to $|S^{fixed}|$ i.e. a package that was allocated by PD1 and remains allocated by component 2 of phase II, any package containing such an item will remain allocated under $fitted\_PD1$ since those packages retain its original average reward per item and because item i^h is the item that under the **while** loop (line (3)) led to the allocation in PD1 of that package that contains i^h, so using M at initialization for the dual prices for $fitted\_PD1$, $q = \arg\min_{j:i^h\in S_j}\{\sum_{i:i\in S_j}\pi_i - p_j\}$ must be the same under PD1 and $fitted\_PD1$.■

proof of theorem 7:

It suffices to show that the set of constraints that are in the $Active\_set$ and $Inactive\_set$ are the same for WD^0 and WD^Q under the respective applications of PD1 and that the resulting dual prices are identical for all dual variables in WD^Q that appear in WD^0 and equal to M for dual variables unique to WD^Q.

If a dual constraint j for WD^0 is in the $Active\_set$ set under the dual prices produced by PD1, then it is also feasible by the proof of Theorem 5 . Thus, no extra variables are added to the corresponding constraint in WD^Q.

Similarly, if a dual constraint in WD^0 was in the $Inactive\_set$ and feasible under dual prices produced by PD1 then no new dual variables are added and so the corresponding constraint will be unaltered in WD^Q.

Suppose a dual constraint for WD^0 associated with a subset S_j is infeasible under prices produced by PD1. Then, it must be the case that this constraint is in the $Inactive\_set$ by the proof of Theorem 5. Now by adding the variables $\pi_l^{v_j}$ for $l = 1\ to\ Q_j$ to the infeasible constraint to form the corresponding constraint for WD^Q the constraint takes the form

$\sum_{i:i\in S_j}\pi_i + \sum_{l=1}^{Q_j}\pi_l^{v_j} \geq p_j$. Thus, the quantity $\sum_{i:i\in S_j}\pi_i + \sum_{l=1}^{Q_j}\pi_l^{v_j} - p_j = \sum_{i:i\in S_j^*}\pi_i - p_j$ is strictly greater compared to $\sum_{i:i\in S_j}\pi_i - p_j$ where $S_j^* = S_j \cup \{i_l^{v_j}\}$ for $l = 1, .., Q_j$ at the start of PD1 since (1) the higest average price M for WD^0 continues to be for WD^Q and (2) all dual variables are initially assigned to M.

Also, since these added variables are unique to the infeasible dual constraint no other dual constraint k in WD^Q that was feasible under the PD1 application to WD^0 will experience a decrease (or increase) in the sum of its dual values after the dual variables are initialized to M i.e. $\sum_{i:i\in S_k}\pi_i - p_j = M|S_k| - p_j$ will not decrease (or increase) since no new dual variables were added or removed for such constraints k. Furthermore, this quantity $\sum_{i:i\in S_k}\pi_i - p_j$ must have been the same at the start of PD1 applied to WD^0 for the corresponding constraint as well since all dual variables remain identical.

Assume that the ground items are visited in the same order for the **while** loop starting on line (3). Now suppose at an iteration for item i to be covered PD1 selects by line (4) of the algorithm a dual constraint associated with subset S_q to be active in WD^0 , then it must be the case that the same constraint in WD^Q will be active as well under PD1 as there can be no other dual constraint d that contains the dual variable for item i such that $\sum_{i:i\in S_d}\pi_i - p_d < \sum_{i:i\in S_q}\pi_i - p_q$ since any dual constraint that gets changed for WD^Q is an infeasible dual constraint in WD^0 and its sum of dual variables will only increase as argued above. By the **for** loop that starts on line (10) the items in S_q will have the same pricing. By the **for** loop starting on line (15), the set of dual contraints that will be deemed to be inactive based on conflict with S_q will be the same for both WD^0 and WD^Q as well.

Thus, the constraint associated with the S_j^* must be in the *Inactive_set* when PD1 is applied to WD^Q since the corresponding constraint in WD^0 was in *Inactive* and the sum of dual variables is larger than the sum of dual variables in the corresponding constraint in WD^0. Also, PD1 will continue to set $\pi_l^{v_j} = M$ for all violated constraints j and for $l = 1, .., Q_j$ since these variables are disjoint with all dual variables that are part of active constraints. ■

proof of proposition 17: Since the LP relaxation of the WD is assumed to be equivalent to the WD problem, then all of the constraints of the dual of the relaxation of the WD that are active at the optimal solution to the dual correspond to constraints $j \in Dual\_active$ that in turn correspond to optimal decision variables x_j that are equal to 1 i.e. $\sum_{j \in Dual\_active} p_j$ is the optimal objective value.

Case (1) Suppose Superactive^0 is empty, this implies that there was no constraint that was active after solving the dual of WD in step 0 that wasn't active at end of running PD1, this means that the set of dual constraints that were set active by PD1 is exactly the same set that are active for the dual under the optimal dual solution.

Case (2) Suppose Superactive^0 is not empty. Consider the $Best\_subset\_allocation$ at the end of component 1 of phase 2. So, if $Superactive^0$ is not empty then at least one constaint was active in the dual of the WD relaxation at the optimal dual solution that was not tight under PD1 and thus it must also belong to $Dual\_active$. Therefore all packages in $Best\_subset\_allocation \subseteq Dual\_active$ since all rewards corresponding to packages in the $Dual\_active$ set are collectively optimal.

Now by line 3 S is the argmax$_{\substack{S \subseteq Active \cup Dual_active \\ where\ S_{\sigma(j)} \in S \text{ and } S_a \cap S_b = \phi \ \ \forall S_a, S_b \in S}} \{\sum_{i \in S} p_i\}$

i.e. $Best\_subset\_allocation = S$, i.e. it is that set of disjoint packages chosen from $Superactive^0 \cup Dual\_active$ that has maximum possible value so

$Dual\_active \subseteq Best\_subset\_allocation$ .Thus, $Best\_subset\_allocation = S = Dual\_active$

■

Acknowledgement

We thank David Parkes and Michael Rothkopf for helpful comments.

References

Anandalingam, G., Kwon, R., Ungar, L.(2001) "Single item approximations for combinatorial Auctions", Working Paper, Systems Engineering Department, The University of Pennsylvania.

Andersson, A.,Tenhunen,M., and Ygge, F. (2000) " *Integer Programming for Combina-*

torial Auction Winner Determination", Proceedings ofthe Fourth International Conference on Multi-Agent Systems (ICMAS00). 39-46.

Arkin, E., Hassin, R.(1998) " *Approximating weighted set packing by local search*", Mathematics of Operations Research, 23 pp.640-648.

Ausubel, L. and Cramton, P. (1998) "*Demand reduction and Inefficiency in Multi-unit Auctions*", Working Paper, Department of Economics, University of Maryland, College Park

Ausubel, L. and Cramton, P. (1998) "*The Optimality of Being Efficient*", Working Paper, Department of Economics, University of Maryland, College Park

S. Bikhchandani and J. Ostroy, (1997) (revised June 2001) "*The Package Assignment Model,*" ., Anderson School of Management, Department of Economics, UCLA

Chandra, R., Halldorsson, (2001)" *Greedy Local Improvement and Weighted Set Packing Approximation*", Journal of Algorithms, 39, pp.223-240

Cramton, P., McMillan, J., Milgrom, P., Miller, B., Vincent, D., Wilson,R. (1998), "*Simultaneous Ascending Auctions with Package Bidding*", Report to the Federal Communications Commission

DeMartini,C., Kwasnica,A.M,, Ledyard,O., Porter,D. (1999) "*A New and Improved Design for Multi-Object Iterative Auctions*", ,Working Paper, Department of Economics, California Institute of Technology

DeVries,S., and Vohra, R.(2001) "*Combinatorial Auctions: A Survey*", Manuscript, Zentrum Mathematik,TU Munchen and Dept. of Managerial Economics and Decision Sciences, Kellog School of Management, Northwestern University.

Fisher, M. and Wolsey,L. (1982) "*On the Greedy Heuristic for Continuous Covering and Packing Problems*", SIAM Journal on Algebraic and Discrete Methods,3, 584-591.

Fujishima, Y., Leyton-Brown, K., and Yoav Shoham. (1999) "*Taming the Computational Complexity of Combinatorial Auctions: Optimal and Approximate Approaches*", Computer Science Department, Stanford University

Goemans,M.X., D.P. Williamson, D.P (1997), The Primal-Dual Method for Approximation Algorithms and its Applications to Network Design. In Approximation algorithms for NP-hard problems, D.S. Hochbaum, editor. PWS Publishing Company.

Hobbs, B., Rothkopf, M.H., Hyde, L.C., and O'Neill, R.P., (2000), *"Evaluation of a truthful Revelation Auction for energy Markets with Nonconcave Benefits"*, Journal of Regulatory Economics 18(1), pp.5-32.

Hoffman, K.(1999) "*Combinatorial Optimization: Current Successes and Directions for the Future*", The Journal of Computational and Applied Mathematics, forthcoming

Kelley, F., Steinberg, R., (2000) "A Combinatorial Auction with Multiple Winners for Universal Service", Management Science, 46(4), 586-596.

Kwon, R., Anandalingam, G., Ungar, L.(2001) "*A Primal-Dual Approximation Algorithm for the Set Packing Problem*", Submitted to Operations Research Letters

Marron, D.B, and Bartels, C.W. (1996), "*The Use of Computer-Assisted Auctions for Allocating Tradeable Pollution Permits*" in Market-Based Control: A Paradigm for Distributed Resource Allocation, World Scientific Publishing Co. Pte. Ltd, River Edge, NJ.

Milgrom,P. (2000) "*Putting Auction theory to Work: The Simultaeous Ascending Auction*" Journal of Political Economy,108, forthcoming

Nemhauser,G., and Wolsey,L. (1999) "*Integer Programming and Combinatorial Optimization*", Wiley-Interscience, New York, N.Y.

Nisan,N. (2000) "*Bidding and Allocation in Combinatorial Auctions*", Working Paper, Dept. of Computer Science, Hebrew University.

Papadimitriou,C. and Steiglitz,K. (1982) "*Combinatorial optimization: algorithms and complexity*" (Prentice-Hall, second printing by Dover, 1998)

Park, S., and Rothkopf, M. (2001)"Auctions with Endogenously Determined Biddable Combinations", RUTCOR Research Report #3-2001, Rutgers University, New Brunswick, N.J.

Parkes,D., and Ungar,L.(2000) " *Iterative Combinatorial Auctions: Theory and Practice*", In Proc. 17th National Conference on Artificial Intelligence (AAAI-00), 74–81.

Parkes, D. C. 2001. "*An Iterative Generalized Vickrey Auction: Strategy-Proofness without Complete Revelation.*" In AAAI Spring Symposium on Game Theoretic and Decision Theoretic Agents Stanford University, CA. March, 2001.

Rassenti, S.J., Smith V.L., and Bulfin, R.L.(1982) "A Combinatorial Auction Mechanism for Airport Time Slot Allocation" Bell Journal of Economics, 13:402-417.

Rothkopf, M., Pekec, A., and Harstad,R. (1998) "*Computationally Manageable Combinatorial Auctions*" Management Science 44(8):1131-1147.

Sandholm, T. (1999) "*An Algorithm for Optimal Winner Determination in Combinatorial Auctions*" In IJCAI-99 (International Journal of Computational Artificial Intelligence)

Sandholm, T. (2001) "*CABOB: A fast optimal algorithm for combinatorial auctions*", Combinenet Inc. manuscript, Pittsburgh, PA 15213

Vickrey, W., (1961) "*Counterspeculation, Auctions, and Competitive Sealed Tenders*", Journal of Finance 16, 8-37.

Williamson,D.(1998) "*Lecture Notes on Approximation Algorithms*" IBM Research Report RC 21409, IBM TJ Watson Research Center, Yorktown Heights, New York