

agents follow a myopic best-response bidding strategy, bidding for bundles that maximize their utility given the prices in each round.

4.2 Linear Programming Theory

First, I provide a brief review of basic results in linear programming. See Papadimitriou & Steiglitz [PS82] for a text book introduction, and Chandru's excellent survey papers [CR99b, CR99a] for a modern review of the literature on linear programming and integer programming.

Consider the linear program:

$$\begin{aligned} \max \quad & c^T x && \text{[P]} \\ \text{s.t.} \quad & Ax \leq b \\ & x \geq 0 \end{aligned}$$

where A is a $m \times n$ integer matrix, $x \in R^n$ is a n -vector, and c and b are n - and m -vectors of integers. vectors are column-vectors, and notation c^T indicates the *transpose* of vector c , similarly for matrices. The primal problem is to compute a feasible solution for x that maximizes the value of the objective function.

The dual program is constructed as:

$$\begin{aligned} \min \quad & b^T y && \text{[D]} \\ \text{s.t.} \quad & A^T y \geq c \\ & y \geq 0 \end{aligned}$$

where $y \in R^m$ is a m -vector. The dual problem is to compute a feasible solution for y that minimizes the value of the objective function.

Let $V_{LP}(x) = c^T x$, the value of feasible primal solution x , and $V_{DLP}(y) = b^T y$, the value of feasible dual solution y .

The *weak duality theorem* of linear programming states that the value of the dual always dominates the value of the primal:

THEOREM 4.1 (weak-duality). *Given a feasible primal solution x with value $V_{LP}(x)$ and a feasible dual solution y with value $V_{DLP}(y)$, then $V_{LP}(x) \leq V_{DLP}(y)$.*

PROOF. Solution x is feasible, so $Ax \leq b$. Solution y is feasible, so $A^T y \geq c$. Therefore, $x \leq A^T b$ and $y \geq Ac$, and $c^T x \leq c^T A^T b = b^T AC \leq b^T y$, and $P \leq D$. ■

The *strong duality theorem* of linear programming states that primal and dual solutions are optimal if and only if the value of the primal equals the value of the dual:

THEOREM 4.2 (strong-duality). *Primal solution x^* and dual solution y^* are a pair of optimal solutions for the primal and dual respectively, if and only if x^* and y^* are feasible (satisfy respective constraints) and $V_{LP}(x^*) = V_{DLP}(y^*)$.*

The strong-duality theorem of linear programming can be restated in terms of *complementary-slackness* conditions (CS for short). Complementary-slackness conditions expresses logical relationships between the values of primal and dual solutions that are necessary and sufficient for optimality.

DEFINITION 4.1 [complementary-slackness] Complementary-slackness conditions constrain pairs of primal and dual solutions. *Primal* CS conditions state $x^T(A^T y - c) = 0$, or in logical form:

$$x_j > 0 \Rightarrow A^j y = c_j \quad (\text{P-CS})$$

where A^j denotes the j th column of A (written as a row vector to avoid the use of transpose). *Dual* CS conditions state $y^T(Ax - b) = 0$, or in logical form:

$$y_j > 0 \Rightarrow A_i x = b_i \quad (\text{D-CS})$$

where A_i denotes the i th row of A .

The strong-duality theorem can be restated as the *complementary-slackness theorem*:

THEOREM 4.3 (complementary-slackness). *A pair of feasible primal, x , and dual solutions, y , are primal and dual optimal if and only if they satisfy the complementary-slackness conditions.*

PROOF. P-CS iff $x^T(A^T y - c) = 0$, and D-CS iff $y^T(Ax - b) = 0$. Equating, and observing that $x^T A^T y = y^T Ax$, we have P-CS and D-CS iff $x^T c = y^T b$, or $c^T x = b^T y$. The LHS is the value of the primal, $V_{LP}(x)$, and the RHS is the value of the dual, $V_{DLP}(y)$. By the strong duality theorem, $V_{LP}(x) = V_{DLP}(y)$ is a necessary and sufficient condition for the solutions to be optimal. ■

4.2.1 Primal-Dual Algorithms

Primal-dual is an algorithm-design paradigm that is often used to solve combinatorial optimization problems. A problem is first formulated both as a primal and a dual linear program. A primal-dual algorithm searches for feasible primal and dual solutions that satisfy complementary-slackness conditions, instead of searching for an optimal primal (or dual) solution directly. Primal-dual can present a useful algorithm-design paradigm for combinatorial optimization problems. Instead of solving a single hard primal solution, or a single hard dual solution, a primal-dual approach solves a sequence of restricted primal problems. Each restricted primal problem is often much simpler to solve than the full primal (or dual) problem [PS82].

Primal-dual theory also provides a useful conceptual framework for the design of iterative combinatorial auctions. Prices represent a feasible dual solution, and bids from agents allow a search for a primal solution that satisfies complementary-slackness conditions. If the current solution is suboptimal there is enough information available to adjust dual prices in the right direction. Complementary-slackness conditions provide the key to understanding how it is possible to compute and verify optimal solutions without complete information: it is sufficient to just verify that a feasible solution satisfies CS conditions. Primal-dual algorithms are consistent with the decentralized information inherent in distributed agent-based systems. Optimality reduces to a test of feasibility and complementary-slackness, which is available from agent bids, rather than the direct solution of a primal problem, which requires information about agent valuation functions.

A standard primal-dual formulation maintains a feasible dual solution, y , and computes a solution to a *restricted primal problem*, given the dual solution. The restricted primal is formulated to compute a primal solution that is both feasible and satisfies CS conditions with the dual solution. In general this is not possible (until the dual solution is optimal),

and a relaxed solution is computed. The restricted primal problem is typically formulated to compute this relaxed solution in one of two ways:

1. Compute a feasible primal solution x' that minimizes the “violation” of complementary-slackness conditions with dual solution y .
2. Compute a primal solution x' that satisfies complementary slackness conditions with dual solution y , and minimizes the “violation” of feasibility constraints.

Method (1) is more useful in the context of iterative auction design because it maintains a feasible primal solution, which becomes the *provisional allocation* in the auction, i.e. a tentative allocation that will be implemented only when the auction terminates. The restricted primal problem can be solved as a *winner-determination problem*. I show that computing the allocation that maximizes revenue given agent bids (the solution to winner-determination) is a suitable method to minimize the violation of CS conditions between the prices and the provisional allocation in each round *i*Bundle. Prices in each round of an auction define the feasible dual solution, and agent best-response bids provide enough information to test for complementary-slackness and adjust solutions towards optimality.

As discussed in the introduction to this chapter, I first assume myopic best-response, but later justify this assumption with an extension to compute Vickrey payments at the end of the auction in addition to the efficient allocation (see Chapters 6 and 7).

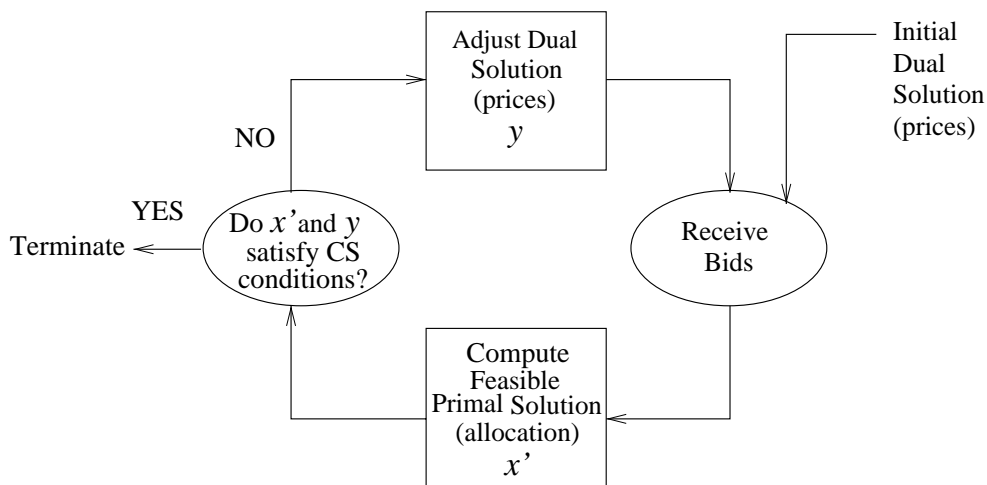


Figure 4.1: A primal-dual interpretation of an auction algorithm.

A primal-dual based auction method has the following form (see Figure 4.1):

1. Maintain a feasible dual solution (“prices”).
2. Compute a feasible primal solution (“provisional allocation”) to minimize violations with complementary-slackness conditions given agents’ bids.
3. Terminate if all CS conditions are satisfied (“are the allocation and prices in competitive equilibrium?”)
4. Adjust the dual solution towards an optimal solution, based on CS conditions and the current primal solution (“increase prices based on agent bids”)

4.3 Allocation Problems

Let us consider the particular form of an *allocation problem*, in which there are a set of discrete items to allocate to agents, and the goal is to maximize value. We assume quasi-linear preferences, and use *utility* to refer to the difference between an agent’s value for a bundle and the price. The primal and dual allocation problems can be stated as follows:

DEFINITION 4.2 [allocation problem: primal] The primal allocation problem is to allocate items to agents to maximize the sum value over all agents, such that no item is allocated to more than one agent.

DEFINITION 4.3 [allocation problem: dual] The dual allocation problem is to assign *prices* to items, or bundles of items, to minimize the sum of (i) each agents’ maximum utility given the prices, over all possible allocations; *and* (ii) the maximum revenue over all possible allocations given the prices.

Clearly, without information on agents’ values the auctioneer cannot compute an optimal primal or an optimal dual (because of term *(i)* in the dual). However, under a reasonable assumption about agents’ bidding strategies (myopic best-response) the auctioneer can verify complementary-slackness conditions between primal and dual solutions, and adjust prices and the allocation towards optimal solutions.

An auction interpretation of the complementary-slackness conditions can be stated as follows:

DEFINITION 4.4 [allocation problem: CS conditions] The CS between a feasible primal solution to an allocation problem, x , and a feasible dual solution, prices p , are:

- (CS-1) Agent i receives bundles S_i in the provisional allocation if and only if the bundle maximizes its utility given the prices, and has non-negative utility.
- (CS-2) The provisional allocation $S = (S_1, \dots, S_I)$ is the revenue-maximizing allocation given the prices.

Left deliberately vague at this stage is the exact *structure* of the prices. In a combinatorial allocation problem these might need to be non-linear and non-anonymous prices to support the optimal allocation. Similarly, the revenue-maximization concept must be defined with respect to a particular linear program formulation. Note also that CS-2 is not automatically satisfied with a provisional allocation computed to maximize revenue given agents' bids. CS-2 makes a stronger claim, that the provisional allocation must maximize revenue over *all possible* allocations given the current ask prices, not just over all allocations consistent with bids.

Primal-dual auction analysis requires the following assumption about agent strategies:

DEFINITION 4.5 [myopic best-response] A myopic best-response bidding strategy is to bid for all items or bundles of items that maximize utility at the current prices.

Best-response bids provide enough information to test CS-1, because the best-response of an agent is precisely those bundles that maximize an agent's utility given the current prices. For any feasible primal solution, the auctioneer can test CS-2 because that only requires price information.

The restricted primal has a natural auction interpretation:

DEFINITION 4.6 [auction restricted-primal problem] Given best response bids from each agent allocate bundles to maximize revenue, breaking ties in favor of including more agents in the provisional allocation.

Note well that a bundle is only allocated to an agent in the restricted primal problem if the agent bids for that bundle. This restriction ensures that CS-1 is satisfied for that agent, given the definition of myopic best-response. CS-2 is satisfied with careful price-adjustment rules, such that prices are increased "slowly enough" that the revenue-maximizing allocation can always be computed from agent bids.

Given myopic best-response, the termination condition, which tests for complementary-slackness between the provisional allocation and the prices, must check that CS-1 holds

for every agent. This is achieved when every agent to submit a bid receives a bundle in the provisional allocation, i.e. in competitive equilibrium.

Our interest is in solving the CAP, which is most immediately formulated as an integer program (see Section 4.4). In order to apply primal-dual methods it is essential that we have a linear program formulation of the CAP. We must be careful enough to use a strong enough formulation, such that the optimal solution is integral (0-1) and not fractional. The ideal situation is illustrated in Figure 4.2. The auction implements a primal-dual algorithm for a linear program that is strong enough to compute the optimal integer solution.

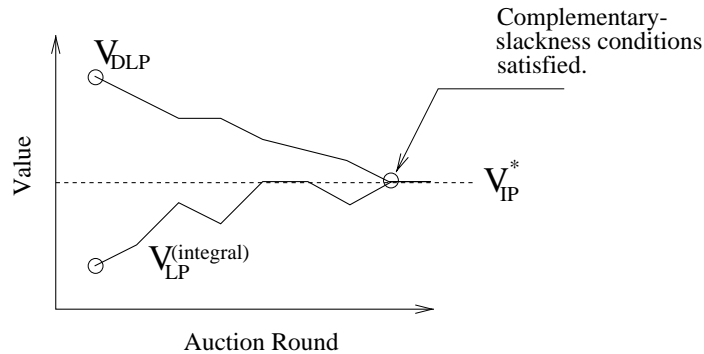


Figure 4.2: An auction-based primal-dual algorithm in which the linear program formulation is strong enough to eliminate all fractional solutions.

In comparison, consider Figures 4.3 (a) and (b), which illustrate a primal-dual algorithm and iterative auction method for a linear program that is not strong enough, and admits optimal fractional solutions. The primal-dual algorithm terminates with a fractional primal solution and value greater than the value of the best possible integer solution. The auction always maintains an integral primal solution (solving winner-determination to compute the provisional allocation), but can terminate with a primal solution that does not satisfy complementary-slackness conditions. Although the primal solution is perhaps optimal, its optimality cannot be assessed without CS information.

4.3.1 Price Adjustment

Left undefined at the moment, and the challenging part of primal-dual auction design, are the precise rules used to define price updates. The goal is to use information from agents' bids, and the current provisional allocation, to adjust prices towards an optimal dual solution—that will support an optimal primal solution. Primal-dual methods traditionally

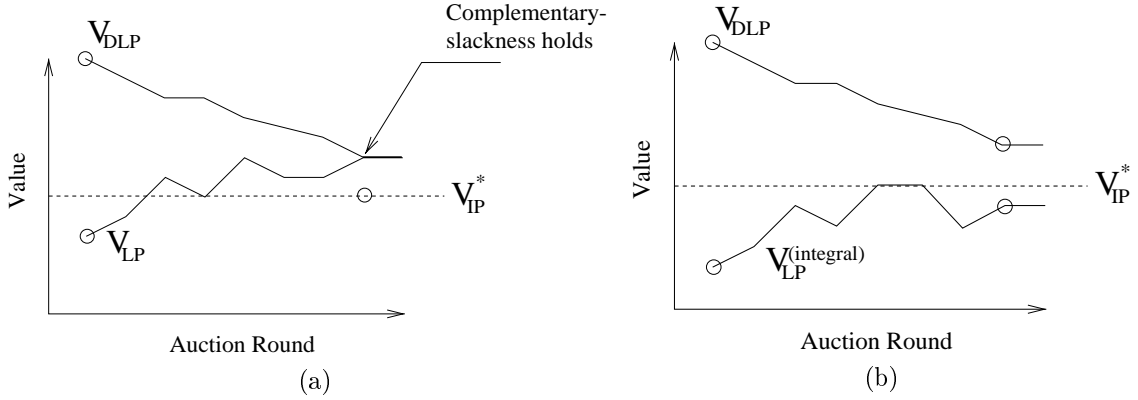


Figure 4.3: Primal-dual algorithm (a) and Primal-dual auction method (b) in which the linear program relaxation is too weak, and $V_{LPR}^* > V_{IP}^*$.

use the dual of the restricted primal to adjust the dual solution across iterations. A simpler method in allocation problems is to *increase prices on over-demanded items, or bundles of items*. The method can be explained both in terms of its effect on complementary-slackness conditions and in terms of its effect on the value of the dual solution.

The idea is to increase prices to: (a) maintain CS-2 in the next round and (b) move towards satisfying CS-1 for all agents.

PROPOSITION 4.1 (progress). *Progress is made towards satisfying CS-1 and CS-2 with the provisional allocation and the ask prices if: (1) the auctioneer increases prices on one or more bundles that receive bids in each round; and (2) the auctioneer increases prices by a small enough increment that best-response bids from agents continue to maximize revenue in the next round.*

CS-1 holds whenever every agent that bids receives a bundle in the provisional allocation. This is trivially achieved for high enough prices because no agent will bid, but we need to achieve this condition in combination with CS-2. The trick is to increase prices just enough to maintain revenue-maximization from bids CS-2 across all rounds. This is achieved in *iBundle* by ensuring that myopic agents continue to bid for bundles at the new prices, i.e. increasing price on over-demanded bundles.

An alternative interpretation is that increasing prices on over-demanded items will *reduce the value of the dual*, making progress towards the optimal solution, see Figure 4.4. Recall that the value of the dual is the sum of the auctioneer's maximal revenue and each

agent’s maximal utility at the current prices. A price increase will decrease the value of the dual if the increase in maximal revenue from the price increase is *less* than the decrease in total maximal utility summed across agents.

The auctioneer can achieve this effect of increasing revenue by less than the decrease in agent utility by selecting over-demanded items, or bundles of items, on which to increase the price. Suppose that two agents bid for bundle S_1 , and that both agents have at least $\epsilon > 0$ more utility for that bundle than any other bundle at the current prices. Increasing the price on over-demanded bundle S_1 by ϵ will decrease the maximal utility of *both* agents by ϵ , for a decrease in dual value of 2ϵ . However, increasing the price on this one bundle by ϵ can increase the auctioneer’s maximal revenue by at most ϵ . The result is that the net change in utility must a decrease of at least ϵ .

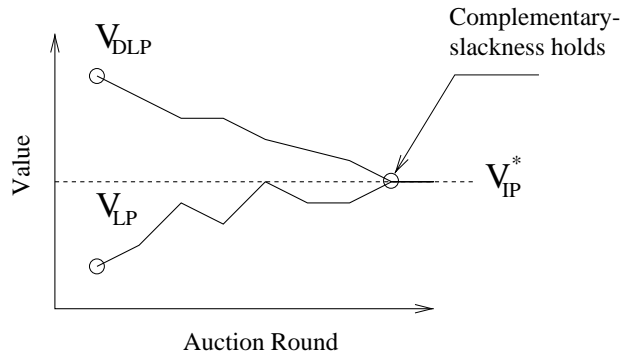


Figure 4.4: Primal-dual interpretation of an ascending-price auction.

4.3.2 Competitive Equilibrium

The optimal primal and dual solutions in an allocation problem correspond to a classic statement of *competitive equilibrium*.

DEFINITION 4.7 [competitive equilibrium] Allocation S and prices p are in competitive equilibrium when:

- (a) every agent receives a bundle in its best-response (utility maximizing) set
- (b) the allocation maximizes the revenue for the auctioneer at the prices

The allocation in competitive equilibrium is efficient, by equivalence between competitive equilibrium and primal-dual optimality:

THEOREM 4.4 (competitive equilibrium efficiency). *An allocation S is efficient if and*

only if there exists competitive equilibrium prices p , for an appropriate type of prices (e.g. linear, bundle, non-anonymous).

In the context of the combinatorial allocation problem Bikchandani & Ostroy [BO99] have characterized the structure on prices required for the existence of competitive equilibrium (and equivalently for integral solutions to linear program formulations of CAP). These formulations are introduced in Section 4.4 and discussed at length.

In some problems it is necessary that prices are both non-linear (bundle prices) and non-anonymous (different prices for the same bundle to different agents) to support a competitive equilibrium solution.

Wurman & Wellman [WW99, WW00] propose an alternative definition of competitive equilibrium, which is essentially complementary slackness condition CS-1 without CS-2. This relaxed condition is sufficient for the existence of equilibrium prices even without non-anonymous prices, but too weak to be able to claim that equilibrium prices imply an efficient allocation.

4.3.3 Example: The English Auction

The standard English auction illustrates the primal-dual framework for auction design. The English auction is an ascending-price auction for single items, where the price increases as long as more than one agent bids at the current price.

Let v_i denote agent i 's value for the item. The single-unit resource allocation problem is:

$$\begin{aligned} & \max \sum_i v_i x_i && \text{[IP}_{\text{single}}\text{]} \\ \text{s.t.} & \sum_i x_i \leq 1 \\ & x_i \in \{0, 1\} \end{aligned}$$

where $x_i = 1$ if and only if agent i is allocated the item, i.e. the goal is to allocate the item to the agent with the highest value. This can be solved as a linear program, [LP_{single}], relaxing the integral constraint

$$\begin{aligned}
& \max \sum_i v_i x_i && \text{[LP}_{\text{single}}\text{]} \\
\text{s.t. } & \sum_i x_i \leq 1 \\
& x_i \geq 0
\end{aligned}$$

and $V_{\text{LP}}^* = V_{\text{IP}}^*$, i.e. there is always an integral optimal solution to the relaxed problem. The dual formulation, $[\text{DLP}_{\text{single}}]$, is

$$\begin{aligned}
& \min \pi && \text{[DLP}_{\text{single}}\text{]} \\
\text{s.t. } & \pi \geq v_i, \quad \forall i \\
& \pi \geq 0
\end{aligned}$$

The complementary-slackness conditions are

$$\begin{aligned}
\sum x_i \geq 0 & \Rightarrow \pi = v_i, \quad \forall i \\
\pi > 0 & \Rightarrow \sum x_i = 1
\end{aligned}$$

The complementary-slackness conditions can be interpreted in terms of competitive equilibrium conditions on the allocation and the prices. An allocation and prices in a single-item auction are in competitive equilibrium, and the allocation is efficient, when:

(i) the item is sold to an agent, that agent bids for the item at the price, and no other agent bids for the item at the price.

or (ii) the item is sold to no agent, the price is zero, and no agent bids for the item.

It is straightforward to understand efficiency in these cases: in (i) the agent with the highest value receives the item; in (ii) no agent has a positive value for the item.

The English auction maintains price p on the item, initially $p = 0$. Agent i bids whenever $p < v_i$, and the provisional allocation sets $x_j = 1$ for one of the agents that bids in each round, and increases the price p whenever more than one agent bids.

Let the provisional allocation define a feasible primal solution, and the price define dual solution $\pi = \sum_i \max\{0, v_i - p\} + p$. This is feasible, $\pi \geq \max\{0, v_i - p\} + p \geq v_i$ for all agents i .

Assume that agents follow a myopic best-response bidding strategy, bidding for the item at the ask price whenever the price is below their value. The optimality of the English auction can be understood in two different ways:

- The English auction terminates with primal and dual solutions that satisfy CS-1 and CS-2.

Clearly, CS-2 is satisfied throughout the auction because the item is always allocated to one of the agents. CS-1 is satisfied when the auction terminates. Let j indicate the only agent that bids at price p . Therefore $v_i - p \leq 0$ for all agents $i \neq j$ and $v_j - p \geq 0$ for agent j (because agents follow best-response bidding strategies), and $\pi = \sum_i \max\{0, v_i - p\} + p = \max\{0, v_j - p\} + p = v_j$.

- The value of the dual strictly decreases in each round of the auction. Let $m > 1$ equal the number of agents that bid in each round of the auction except the final round. For price increment ϵ , the sum maximal utility to the agents decreases by $m\epsilon$ and the maximal revenue to the auctioneer increases by ϵ , for a net change in π of $-(m - 1)\epsilon$.

In fact, the final price in the English auction approaches the Vickrey payment (i.e. the second-highest value) as the bid increment $\epsilon \rightarrow 0$. It follows that myopic-best response is a rational *sequential* strategy for an agent, in equilibrium with myopic best-response strategies from other agents (see Chapter 7 for a full discussion of the incentive properties of iterative Vickrey auctions).

4.4 Linear Program Formulations for the Combinatorial Allocation Problem

Primal-dual based auction methods require linear programming formulations of allocation problems. Bikchandani & Ostroy [BO99] have formulated a hierarchy of linear programs for the problem, introducing additional constraints to remove fractional solutions. Although it is always possible to add enough constraints to a linear program relaxation to make the optimal solution integral [Wol81a, Wol81b, TW81], the particular formulations proposed by Bikchandani & Ostroy are interesting because the constraints have natural interpretations as prices in the dual.

The hierarchy of linear program formulations, $[LP_1]$, $[LP_2]$, and $[LP_3]$, all retain the set of integer allocations but prune additional fractional solutions. Each formulation introduces new constraints into the primal, with the dual problems $[DLP_1]$, $[DLP_2]$, and $[DLP_3]$ containing richer price structures. For example, in $[DLP_1]$ the prices on a bundle are linear in the price of items, i.e. $p(S) = \sum_{j \in S} p(j)$, where $p(j)$ is the price of item j in bundle S . Moving to $[DLP_2]$, the price on a bundle can be non-linear in the price on items, and in $[DLP_3]$ the price on a bundle can be different to different agents. Bikchandani & Ostroy prove that LP_3 solves all CAP instances, and demonstrate the existence of competitive equilibrium prices, even though they must sometimes be both non-linear and non-anonymous.

Solving the CAP with the high-level linear program formulations is likely to be less efficient computationally than direct search-based methods applied to the integer program formulation. Formulations $[LP_2]$ and $[LP_3]$ introduce an exponential number of additional primal constraints, and dual variables, effectively enumerating *all possible* solutions to the CAP. In comparison, search methods, such as branch-and-bound with LP-based heuristics, solve the problem with implicit enumeration and pruning.

However the formulations are very useful in the context of mechanism design and decentralized CAP problems. In Section 4.6 I present COMBAUCTION, a primal-dual algorithm for the CAP, which

(a) computes optimal primal and dual solutions *without* complete information about agent valuation functions.

(b) computes optimal primal and dual solutions *without* complete enumeration of all primal constraints and/or dual variables.

In fact most of the computation within COMBAUCTION occurs in winner determination, which solves the restricted primal problem in each round, and winner-determination itself is solved with a branch-and-bound search algorithm.

4.4.1 Integer Program Formulation

Introducing $x_i(S)$ to indicate that agent i receives bundle S the straightforward integer program, [IP], formulation of the combinatorial allocation problem is:

$$\max_{x_i(S)} \sum_S \sum_i x_i(S) v_i(S) \quad [\text{IP}]$$

$$\text{s.t. } \sum_S x_i(S) \leq 1, \quad \forall i \quad (\text{IP-1})$$

$$\sum_{S \ni j} \sum_i x_i(S) \leq 1, \quad \forall j \quad (\text{IP-2})$$

$$x_i(S) \in \{0, 1\}, \quad \forall i, S$$

where $S \ni j$ indicates a bundle S that contains item j . The objective is to compute the allocation that maximizes value over all agents, without allocating more than one bundle to any agent (IP-1) and without allocating a single item multiple times (IP-2). Let V_{IP}^* denote the value of the optimal allocation.

4.4.2 First-order LP Formulation

LP_1 is a direct linear relaxation, which replaces the integral constraints $x_i(S) \in \{0, 1\}$ with non-negativity constraints, $x_i(S) \geq 0$.

$$\max_{x_i(S)} \sum_S \sum_i x_i(S) v_i(S) \quad [\text{LP}_1]$$

$$\text{s.t. } \sum_S x_i(S) \leq 1, \quad \forall i \quad (\text{LP}_1\text{-1})$$

$$\sum_{S \ni j} \sum_i x_i(S) \leq 1, \quad \forall j \quad (\text{LP}_1\text{-2})$$

$$x_i(S) \geq 0, \quad \forall i, S$$

$$\min_{p(i), p(j)} \sum_i p(i) + \sum_j p(j) \quad [\text{DLP}_1]$$

$$\text{s.t. } p(i) + \sum_{j \in S} p(j) \geq v_i(S), \quad \forall i, S \quad (\text{DLP}_1\text{-1})$$

$$p(i), p(j) \geq 0, \quad \forall i, j$$

Prices $p(j)$ on items $j \in \mathcal{G}$ define a feasible dual solution, with the substitution $p(i) = \max_S \{v_i(S) - \sum_{j \in S} p(j)\}$.

PROPOSITION 4.2 (first-order dual). *The value of the first-order dual is the sum of the maximal utility to each agent plus the total price over all items (this is the auctioneer's maximal revenue).*

	A	B	AB
Agent 1	0	0	3
Agent 2	2*	0	2
Agent 3	0	2*	2

Table 4.1: Problem 1.

	A	B	C	AB	BC	AC	ABC
Agent 1	60	50	50	200*	100	110	250
Agent 2	50	60	50	110	200	100	255
Agent 3	50	50	75*	100	125	200	250

Table 4.2: Problem 2.

The dual variables define linear prices, the price for bundle $S \subseteq \mathcal{G}$ is $p(S) = \sum_{j \in S} p(j)$. From Definition 4.7 the optimal dual solution defines competitive equilibrium prices if and only if a partition of items exists at the prices that allocates each agent a bundle in its utility-maximizing set and allocates every item with positive price exactly once.

Problem 1 in Table 4.1 can be solved with $[\text{LP}_1]$; $V_{\text{LP}_1}^* = V_{\text{IP}} = 4$. The optimal allocation is $x_2(A) = 1$ and $x_3(B) = 1$, indicated by *. To see that $V_{\text{LP}_1} \leq 4$, notice that dual prices $p(A) = p(B) = 1.6$ gives a dual solution with value $V_{\text{DLP}_1} = 0 + 0.4 + 0.4 + 3.2 = 4$. Remember that $V_{\text{LP}_1}^* \leq V_{\text{DLP}_1}$ for all dual solutions by the weak-duality theorem of linear programming. These are one set of competitive equilibrium prices.

However, in general the value $V_{\text{LP}_1}^* > V_{\text{IP}}^*$ and the optimal primal solution makes fractional assignments to agents. As an example of when $[\text{LP}_1]$ fails, consider Problem 2 in Table 4.2. In this problem $V_{\text{LP}_1}^* = 300 > V_{\text{IP}}^* = 275$. The primal allocates fractional solution $x_1(AB) = 0.5$, $x_2(BC) = 0.5$ and $x_3(AC) = 0.5$, which satisfies constraints (LP₁-1) because $\sum S \ni j \sum_i x_i(S) \leq 1$ for all items $j \in \mathcal{G}$. Prices $p(A) = p(B) = p(C) = 100$ solve the dual problem DLP_1 .

Kelso & Crawford [KC82] prove that gross-substitutes (GS) preferences are a sufficient condition for the existence of linear competitive equilibrium prices, such that $V_{\text{LP}_1}^* = V_{\text{IP}}^*$.

To define gross-substitutes preferences, let $D_i(p)$ define the demand set of agent i at prices p , i.e. the set of bundles that maximize its utility (value - price).

DEFINITION 4.8 [gross-substitutes (GS)] For all price vectors p, p' such that $p' \geq p$, and all $S \in D_i(p)$, there exists $T \in D_i(p')$ such that $\{j \in S : p_j = p'_j\} \subset T$.

In words, an agent has GS preferences if an agent continues to demand items with the same price as the price on other items increases. If preferences are also *monotonic*, such that $v_i(S') \geq v_i(S)$ for all $S' \supseteq S$, then GS implies *submodular* preferences.

DEFINITION 4.9 [submodular preferences] Valuation function $v_i(S)$ is submodular if for all $S, T \subseteq \mathcal{G}$,

$$v_i(S) + v_i(T) \geq v_i(S \cup T) + v_i(S \cap T)$$

Submodularity is equivalent to a generalized statement of *decreasing returns*:

DEFINITION 4.10 [decreasing returns] Valuation function $v_i(S)$ has decreasing marginal returns if for all $S \subset T \subseteq \mathcal{G}$ and all $j \in \mathcal{G}$,

$$v_i(T) - v_i(T \setminus \{j\}) \leq v_i(S) - v_i(S \setminus \{j\})$$

In other words, the value of an item increases as it is introduced to larger sets of items. Subadditivity implies that the value for any bundle is no greater than the minimal sum of values for a partition of the bundle.

In fact, gross-substitutes preferences define the largest set of preferences that contain unit-demand preferences (see Definition 4.14) for which the existence of linear competitive equilibrium prices can be shown [GS99].

The rest of this section introduces two alternative linear program formulations of CAP, [LP₂] and [LP₃], due to Bikchandani & Ostroy [BO99].

4.4.3 Second-order LP Formulation

Introducing new constraints to the first-order linear program relaxation [LP₁] of [IP] gives a second-order linear program [LP₂] with dual [DLP₂]. The corresponding dual variables to the new primal constraints are interpreted as *bundle prices* within an auction-based primal-dual algorithm.

$$\begin{aligned}
& \max_{x_i(S), y(k)} \sum_S \sum_i x_i(S) v_i(S) && \text{[LP}_2\text{]} \\
\text{s.t. } & \sum_S x_i(S) \leq 1, \quad \forall i && \text{(LP}_2\text{-1)} \\
& \sum_i x_i(S) \leq \sum_{k \ni S} y(k), \quad \forall S && \text{(LP}_2\text{-2)} \\
& \sum_k y(k) \leq 1 && \text{(LP}_2\text{-3)} \\
& x_i(S), y(k) \geq 0, \quad \forall i, S, k
\end{aligned}$$

$$\begin{aligned}
& \min_{p(i), p(S), \pi} \sum_i p(i) + \pi && \text{[DLP}_2\text{]} \\
\text{s.t. } & p(i) + p(S) \geq v_i(S), \quad \forall i, S && \text{(DLP}_2\text{-1)} \\
& \pi - \sum_{S \in k} p(S) \geq 0, \quad \forall k && \text{(DLP}_2\text{-2)} \\
& p(i), p(S), \pi \geq 0, \quad \forall i, S
\end{aligned}$$

where $k \in K$ is a *partition* of items in set K , and $k \ni S$ indicates that bundle S is represented in partition k . A partition is a feasible “bundling” of items, e.g. $[A, B, C]$ or $[AB, C]$, etc., and K is the set of all possible partitions, e.g. $K = \{[A, B, C], [AB, C], [A, BC], \dots, [ABC]\}$ in Problem 2 (Table 4.2).

Constraints (LP₂-2) and (LP₂-3) replace constraints (LP₁-1), and ensure that no more than one unit of every item is allocated. The dual [DLP₂] has variables $p(i)$, $p(S)$ and π , which correspond to constraints (LP₂-1), (LP₂-2) and (LP₂-3), and constraints (DLP₂-1) and (DLP₂-2) correspond to primal variables $x_i(S)$ and $y(k)$.

Dual variables $p(S)$ can be interpreted as bundle prices, and with substitution $p(i) = \max_S \{v_i(S) - p(S)\}$, i.e. the maximal utility to agent i at prices $p(S)$, and $\pi = \max_{k \in K} \sum_{S \in k} p(S)$, i.e. the maximal revenue to the auctioneer at prices $p(S)$.

PROPOSITION 4.3 (second-order dual). *The value of the dual is the sum of the maximal utility to each agent with bundle prices $p(S)$ plus the auctioneer’s maximal revenue over all feasible (and non-fractional) allocations at the prices.*

	A	B	AB
Agent 1	0	0	3^*
Agent 2	2	2	2

Table 4.3: Problem 3.

The dual variables correspond to bundle prices, $p(S)$, and the optimal dual solution defines competitive equilibrium prices (by Definition 4.7) if there is an allocation that gives each agent a bundle in its utility-maximizing set at the prices, and maximizes revenue to the auctioneer over all possible allocations.

With the additional constraints $[\text{LP}_2]$ solves Problem 2. Allocation $x_1(AB) = x_2(BC) = x_3(AC) = 0.5$ is *not* feasible in $[\text{LP}_2]$ because it is not possible to allocate $y(k_1) = y(k_2) = y(k_3) = 0.5$ for $k_1 = [AB, C]$, $k_2 = [AC, B]$ and $k_3 = [AB, C]$ without violating constraint (LP₂-3) and without this we violate constraints (LP₂-2). $[\text{LP}_2]$ solves Problem 2, with $V_{\text{LP}_2}^* = V_{\text{IP}}^* = 275$. An optimal dual solution is given by bundle prices $p = (50, 60, 75, 190, 200, 200, 255)$, with total agent maximal utility $10 + 0 + 0$ and maximal auctioneer revenue $75 + 190 = 265$, i.e. $V_{\text{DLP}_2}^* = 275$.

However, Problem 3 is an example that $[\text{LP}_2]$ does not solve. The value of the optimal primal solution is $V_{\text{LP}_2}^* = 3.5$, which is greater than the value of the optimal feasible allocation, $V_{\text{IP}}^* = 3$. The primal allocates fractional bundles $x_1(AB) = 0.5$ and $x_2(A) = x_2(B) = 0.5$, which satisfies constraints (LP₂-2) and (LP₂-3) with $y(k_1) = y(k_2) = 0.5$ for partitions $k_1 = [AB, \emptyset]$ and $k_2 = [A, B]$. Prices $p(A) = 1.5, p(B) = 1.5, p(AB) = 3$ solves the dual problem DLP_2 .

4.4.4 Third-order LP Formulation

Introducing new constraints to the second-order linear program relaxation $[\text{LP}_2]$ of $[\text{IP}]$ gives a third-order linear program $[\text{LP}_3]$ with dual $[\text{DLP}_3]$. The corresponding dual variables to the new primal constraints are interpreted as *non-anonymous*, or *discriminatory* bundle prices, with different prices for the same bundle to different agents.

$$\begin{aligned}
& \max_{x_i(S), y(k)} \sum_S \sum_i x_i(S) v_i(S) && [\text{LP}_3] \\
\text{s.t.} \quad & \sum_S x_i(S) \leq 1, \quad \forall i && (\text{LP}_3\text{-1}) \\
& x_i(S) \leq \sum_{k \ni [i, S]} y(k), \quad \forall i, S && (\text{LP}_3\text{-2}) \\
& \sum_k y(k) \leq 1 && (\text{LP}_3\text{-3}) \\
& x_i(S), y(k) \geq 0, \quad \forall i, S, k
\end{aligned}$$

$$\begin{aligned}
& \min_{p(i), p_i(S), \pi} \sum_i p(i) + \pi && [\text{DLP}_3] \\
\text{s.t.} \quad & p(i) + p_i(S) \geq v_i(S), \quad \forall i, S && (\text{DLP}_3\text{-1}) \\
& \pi - \sum_{[i, S] \in k} p_i(S) \geq 0, \quad \forall k && (\text{DLP}_3\text{-2}) \\
& p(i), p_i(S), \pi \geq 0, \quad \forall i, S
\end{aligned}$$

where $k \ni [i, S]$ indicates that *agent-partition* $k \in K'$ contains bundle S designated for agent i . Variable $y(k)$ in $[\text{LP}_3]$ corresponds to an *agent-partition* k , where the set of agent-partitions in Problem 3 is $K' = \{[(1, A), (2, B)], [(1, B), (2, A)], [(1, AB), (2, \emptyset)], [(1, \emptyset), (2, AB)]\}$. It is important to note that each agent can receive at most one bundle in a particular agent-partition.

The dual variables $p_i(S)$ that correspond to primal constraints (LP₃-2) are interpreted as *non-anonymous* bundle prices, price $p_i(S)$ is the price to agent i for bundle S . As before, substitutions $p(i) = \max_S \{v_i(S) - p_i(S)\}$, i.e. the maximal utility to agent i at individual prices $p_i(S)$, and $\pi = \max_{k \in K'} \sum_{[i, S] \in k} p_i(S)$, i.e. the maximal revenue to the auctioneer at prices $p_i(S)$ given that it can allocate at most one bundle at prices $p_i(S)$ to each agent i .

PROPOSITION 4.4 (third-order dual). *The value of the dual to $[\text{LP}_3]$ is the sum of the maximal utility to each agent with bundle prices $p_i(S)$ plus the auctioneer's maximal revenue over all feasible allocations at the prices. In this case an allocation is feasible if it allocates no more than one bundle to each agent.*

The dual variables correspond to non-anonymous bundle prices, $p_i(S)$, and the optimal dual solution defines competitive equilibrium prices if there is an allocation of items that simultaneously gives each agent a bundle in its utility-maximizing set and maximizes the auctioneer’s revenue, over all possible allocations that sell at most one bundle to each agent.

Bikchandani & Ostroy [BO99] prove this important theorem:

THEOREM 4.5 (integrality). *The optimal solution to linear program $[LP_3]$ is always integral, and therefore an optimal solution to CAP, with $V_{LP_3}^* = V_{DLP_3}^* = V_{IP}^*$.*

Therefore, there are always competitive equilibrium bundles prices for CAP, although these prices must be non-anonymous in some problems.

Consider Problem 3. Allocation $x_1(AB) = 0.5$ and $x_2(A) = x_3(B) = 0.5$ is *not* feasible in $[LP_3]$ because $y(k_1) = y(k_2) = y(k_3) = 0.5$ for $k_1 = [(1, AB), (2, \emptyset)]$, $k_2 = [(1, A), (2, B)]$ and $k_3 = [(1, B), (2, A)]$ violates constraint (LP₃-3), but without this constraints (LP₃-2) are violated. In this problem $V_{LP_3}^* = V_{IP}^* = 3$. To see this, consider bundle prices $p_1 = (0, 0, 2.5)$ and $p_2 = (2, 2, 2)$, for which the value of the dual is $0.5 + 0 + 2.5 = 3$. This proves that $V_{LP_3} \leq 3$ by the weak-duality theorem of linear programming.

I will return to this hierarchy of linear-program formulations of the CAP in Section 4.6, when I introduce the COMBAUCTION primal-dual algorithm. COMBAUCTION constructs feasible primal and dual solutions to an appropriate linear program formulation, and adjusts the solution until complementary-slackness conditions are also satisfied.

4.5 Tractable Combinatorial Allocation Problems

The CAP is equivalent to the maximum weighted set packing problem (SPP), a well-studied problem in the operations research literature. In SPP there are a set of items, and a set of subsets each with non-negative weights, and the goal is to pack the items into sets to maximize total value, without using any item more than once. CAP can be reduced to SPP by introducing an additional “dummy item” for the XOR bids from each agent. de Vries & Vohra [dVV00] also note two closely related problems, the set partitioning problem (SPA), in which the goal is to select a set of subsets with minimal

cost that include all items at most once, and the set covering problem (SCP), in which the goal is to select a set of subsets with minimal cost that include all items at least once. Set covering problems find applications in railway crew-scheduling and airline scheduling, where items are flights/trains, and bundles represent possibility sets for individual workers. A considerable amount is known about the complexity of this class of problems.

A classic technique in combinatorial optimization theory is to relax an integer program to a linear one. Many tractable special cases follow by considering the conditions on the natural relaxation of the integer program that provide integer solutions. For example, one sufficient condition is that the linear program is *integral*, such that all extremal feasible points are integral, i.e. 0-1. In this case the integrality requirement can be dropped and the problem solved as a linear program in polynomial time. Restrictions on the constraint matrix, corresponding to restrictions on the kinds of subsets permitted in CAP, can provide this integrality property [dVV00].

Additional restrictions, for example on the size of bids, or on the valuation structure of bids, can also lead to tractable special cases. Given the connection with linear programming relaxations this is a good place to review known tractable special-cases in the literature. The results here are drawn from Rothkopf *et al.* [RPH98], de Vries & Vohra [dVV00], Nisan [Nis00], and earlier work due to Kelso & Crawford [KC82].

It is important to understand the characteristics of tractable special-cases of CAP because this knowledge can be leveraged within mechanism design, achieving tractable and strategy-proof solutions (see Section 3.2.1 in Chapter 3).

Restrictions on Structure of Bundles

Table 4.4 presents tractable instances of CAP that follow from restrictions on the types of bundles on which agents can submit bids. de Vries & Vohra note that the linear-ordering (or consecutive ones) condition implies that the constraint matrix satisfies *total unimodularity*,¹ and that the nested-hierarchical structure implies that the constraint matrix is *balanced*.² Nisan [Nis00] provides a proof-by-induction that the linear program has integral solutions in these cases, and also describes a method to combine two bid structures with the integral property into a single structure that retains the property.

¹A matrix satisfies total unimodularity if the determinant of every square submatrix is 0, 1, or -1.

²A 0-1 matrix is balanced if it has no square submatrix of odd order with exactly two 1's in each row and column.

linear-order	ordering $G = (g_1, g_2, \dots, g_n)$	[RPH98]
circular ones	every bid is for a contiguous sequence	
nested-hierarchical	also allow bids of form $g_n g_1 g_2$, etc.	[RPH98]
or-singletons	for every two subsets of items S_1, S_2	[RPH98]
single-item bids	that appear as part of any bid they are either disjoint or one contains the other	
bids for pairs of items	bids for single-items	-
multi-unit, decreasing returns	one item	-
	cardinality constraint on size of bids	[RPH98]
	identical items, each agent has decreasing value for each additional item	[Nis00]

Table 4.4: Tractable structure on bids

non-decreasing and supermodular	“increasing returns”	[dVV00]
two-types of agents		
gross-substitutes	“decreasing-returns”	[KC82]
unit-demand	agents only want one item	[Kuh55]
linear-additive	agents have linear values across items	[CK81]

Table 4.5: Constraints on valuation functions

Restrictions on Values on Bundles

Table 4.5 presents tractable instances of CAP that follow from restrictions on the value structure of agents bids. de Vries & Vohra [dVV00] note that the non-decreasing and supermodular preferences condition again provides the linear program relaxation of the CAP with integral solutions. Gross-substitutes were defined earlier in Definition 4.8 and have an intuitive interpretation as decreasing-returns, and also imply submodular preferences.

DEFINITION 4.11 [supermodular preferences] Bid function $b_i(S)$ is supermodular if for all $S, T \subseteq \mathcal{G}$,

$$b_i(S) + b_i(T) \leq v_i(S \cup T) + v_i(S \cap T)$$

The equivalence of supermodularity and increasing returns is well-known in the literature [GS99].

DEFINITION 4.12 [increasing returns] Bid function $b_i(S)$ has increasing marginal returns if for all $S \subset T \subseteq \mathcal{G}$ and all $j \in \mathcal{G}$,

$$b_i(T) - v_i(T \setminus \{j\}) \geq b_i(S) - v(S \setminus \{j\})$$

Note carefully that we can have any number of different types of submodular valuation functions, one from each agent, but only at most *two* different types of supermodular functions if the CAP problem is to be tractable. It is easier to solve a maximization problem, such as the CAP, with submodular (convex) objective functions than supermodular (concave) objective functions.

Exact Solutions

Rothkopf *et al.* [RPH98] also suggest a dynamic programming algorithm for CAP, which has run-time complexity independent of the number of bids actually placed, but quickly becomes intractable for large numbers of items, with scaling property $O(3^m)$ in the number of items m . Branch-and-bound search methods, either with AI-based heuristics [San99, FLBS99], or with linear-program based heuristics [ATY00] have also been studied for general CAP instances.

Approximate Solutions

The CAP is difficult to approximate, at least within a worst-case multiplicative factor. There is no polynomial time algorithm with a reasonable worst-case guarantee [Has99].

Approximation algorithms in the literature without this guarantee include a local-search approach [HB00], a simple “relax and round” method [Nis00], and iterative methods [FLBS99]. COMBAUCTION can itself be viewed as an approximate algorithm for CAP. COMBAUCTION provides a worst-case bound on the difference between the value of its solution and the value of the optimal solution. This error-term increases linearly with the minimal bid increment, which defines the rate at which prices are increased across rounds, while the number of rounds in the auction is inversely-proportional to the minimal bid increment. A larger bid increment reduces the number of rounds in the auction, reducing the number of winner-determination problems the auction must solve, in return for a loss in worst-case efficiency. Experimental results in Section 5.5.1 show the effectiveness of this approach.