Distributed Generalized Vickrey Auctions Based on the Dantzig-Wolfe and Benders Decomposition Methods for Linear Programs

A thesis presented
by

Ryan W. Davies

To
Applied Mathematics
in partial fulfillment of the honors requirements
for the degree of
Bachelor of Arts
Harvard College
Cambridge, Massachusetts

March 23, 2005

Contents

Abstract

In the field of mechanism design, the goal of a decentralized, efficient, and strategy-proof auction mechanism has been well-established. Integer and linear programs are natural ways to represent the allocation problems that mechanism design addresses, in this case the Combinatorial Allocation Problem (CAP). The goal of this thesis is to propose several frameworks for an iterative, distributed auction, based on the Dantzig-Wolfe and Benders decomposition methods for linear and mixed integer programs. The strategy is to formulate the CAP rigorously as a linear or mixed integer program, to solve using a decomposition method (Dantzig-Wolfe or Benders), and to create a reasonable auction interpretation of the process. The result is a design for a distributed implementation of a Generalized Vickrey Auction (GVA), from the family of Vickrey-Clarke-Groves (VCG) mechanisms.

Chapter 1

Introduction

In the field of mechanism design, the goal of a decentralized, efficient, and strategy-proof auction mechanism has been well established. Central mechanisms, in particular Groves mechanisms, have been well-researched, but in general fall short of simultaneously providing both the game-theoretic and computational properties that we might desire from such a system. The main motivations for a distributed implementation are threefold: 1), that such an implementation could more easily and naturally be parallelized in the face of increasing computational demands, 2), that a distributed mechanism could possibly provide stronger privacy properties than a centralized direct-revelation mechanism, and 3), that in some applications a trusted, computationally capable central agent might not be feasible.

A distributed implementation provides each agent with the specifications for its role in a distributed algorithm, which combines the work of all of the agents to produce an optimal solution to the allocation problem. The difficulty of distributed mechanism lies in the fact that, compared to a centralized design, the participating agents have more of a role in the computation and thus more of an opportunity to manipulate the outcome. A self-interested agent would, if possible, try to deviate from its prescribed actions in order to win more goods and lower its own payments. Our goal is to design a distributed algorithm that align incentives such that the agents choose, out of their own self-interest, to perform their part of the distributed algorithm correctly.

Integer and linear programs are natural ways to represent the allocation problems that mechanism design addresses, and primal-dual algorithms have been studied in the past as ways to implement effective mechanisms. In this paper we will propose two frameworks for an iterative, distributed auction based on decomposition methods for linear programs. The first is based on the Dantzig-Wolfe decomposition, which takes advantage of block structure in the coefficient matrix to decompose the problem into a master problem and one or more simpler subproblems. The second is based on Benders decomposition, which separates the variables into two classes and iterates toward a solution by fixing one class and solving a subproblem based on the second class of variables alone. The end goal of both of these frameworks is a distributed implementation of a Generalized Vickrey Auction (GVA), from the family of Vickrey-Clarke-Groves (VCG) mechanisms.

At a high level, the Dantzig-Wolfe decomposition is designed to take advantage of special structure in large problems to solve them more easily. In the case of an auction, the problem can be broken down into a separate valuation problem for each agent, linked by constraints ensuring that no good can be allocated to more than one agent. The center's role is to enforce these resource constraints and to pass prices to the agents that reflect the marginal value or demand for each good. Each agent has its own subproblem, which is to determine and report its best bundle under these prices.

The Benders is in a sense the opposite of Dantzig-Wolfe. Rather than accumulate a set of proposals, the center in Benders accumulates a core of pricing information by broadcasting proposals to the agents and receiving valuation information in return. Whereas Dantzig-Wolfe terminates when no new proposals are received from agents, in Benders the center develops a more and more accurate sense of the value of various possible partitions of bundles, and terminates when it is satisfied that its prices are accurate enough.

The combinatorial allocation problem (CAP) is widely useful model for a resource allocation problem where agents have complicated valuation schemes. The CAP is easily formulated as an integer program, but naive linear program relaxation is not guaranteed to produce feasible allocations, as it may allocate fractions of bundles which are in reality indivisible. Hence, to employ linear program solution methods, we must first formulate the program more strongly so that even under a continuous relaxation, solutions are guaranteed to be integral. Fortunately, Bikhchandani and Ostroy [2001] and Bikhchandani et al. [2001] have already undertaken this problem and from their work we have three stronger formulations - called LP2, LP3, and LP4 - which will be of tremendous value to us. The main substance of this thesis treats the application of our two chosen decomposition methods (Dantzig-Wolfe and Benders) to these three strong linear programming formulations of the CAP.

1.1 Related Work

The idea of distributed auction mechanisms is not new. Bertsekas [1988] proposed an iterative distributed process for solving optimization problems with a similar flavor to the Dantzig-Wolfe auction proposed in this thesis, wherein prices are determined centrally and at each rounds agents respond with their preferred bundle under these prices. Demange et al. [1986] propose an iterative, ascending-price auction where prices are raised each round on over-demanded goods. This can be seen as a primal-dual approach to the problem, as outlined more generally by Bikhchandani et al. [2001].

In his doctoral thesis, Parkes [2001] proposes the *i*Bundle family of iterative combinatorial auctions, designed to implement the overall welfare-maximizing outcome in the face of self-interested agents. This dissertation also connects the idea of primal-dual auction methods back to the VCG mechanism and the computation of Vickrey payments. Parkes and Shneidman [2004] take up the idea of distributed mechanism implementations, and demonstrate a number of principles to guide decentralization, particularly for VCG mechanisms for implementing outcomes that maximize total agent value.

Kameshwaran [2004] uses Benders decomposition to solve several versions of the piecewise linear knapsack problem, which is similar to the CAP. His work views the knapsack problem as a winner determination problem only, and so does not seek to decentralize it or interpret it explicitly as an auction. He also finds that an exact algorithm based on the Benders decomposition performs worse empirically on the knapsack problem than a number of other hybrid algorithms based branch-and-bound and dynamic programming techniques.

Finally, Abrache et al. [2004] directly suggest the use of decomposition methods to design iterative combinatorial auctions, and explore both Lagrangian relaxation and the Dantzig-Wolfe decomposition as bases for auctions. They formulate a Dantzig-Wolfe auction for a *continuous* allocation problem and conclude that, while the auction has the benefit of maintaining a feasible provisional allocation at all times, it runs into difficulties with protecting agent privacy and its incentive properties are weak. In their conclusion they specifically suggest exploring the Bikhchandani and Ostroy [2001] extended LP formulations as an avenue for future research in the case of indivisible (rather than continuously divisible) goods.

Building on this existing body of work, this thesis aims to explore decomposition methods as a basis for an iterative combinatorial auction, specifically the case of discrete, indivisible goods. Our hope is to progress in the direction of such a design with favorable properties in both the incentive realm and the computational realm.

1.2 Outline

Chapter 2 of this thesis presents the combinatorial allocation problem and its importance as a mathematical model for a diverse set of real-world applications, in order to motivate the remainder of this work. It also provides a brief overview of classic mechanism theory and more thoroughly discusses the reasons why a distributed approach seems attractive from that point of view.

Chapter 3 explains the role of linear programming and introduces the Dantzig-Wolfe decomposition, mentioning in particular the pricing role played by the solutions to the dual problem and the economic interpretation thereof. Chapter 4 introduces the auction design in the simpler domain of the assignment problem, and Chapter 5 and generalizes this design to the full combinatorial allocation problem. Chapter 5 also addresses the multiplicity of dual solutions that may exist, explores the possibility of optimizing the algorithm by exploiting similarities in the linear programs in consecutive rounds, and offers a concluding assessment of the Dantzig-Wolfe auction.

Chapter 6 then introduces the Benders decomposition, and Chapter 7 presents in depth the Benders auction based on the LP2 formulation. The subproblem and master problem are discussed, the problem of decentralization is investigated at length, and several concrete, small-scale examples are presented to help develop intuition for the process. Chapter 8 considers designing a Benders auction using the LP3 and LP4 formulations, but finds LP3 a poor match for the Benders technique. Applying Benders to LP4 is quite complicated and offers long-term more promise, although the analysis presented here is inconclusive.

Finally, Chapter 9 concludes, offering a comparison between the Dantzig-Wolfe and Benders auctions, thoughts on the implications of this thesis for the goal of an incentive-compatible, distributed combinatorial auction, and suggestions for further research on this topic.

Chapter 2

Classic Mechanism Design

Mechanism design is a field that combines computer science and artificial intelligence with economics and game theory to provide multi-agent systems with solutions to problems. A key feature of mechanism problems is the need to induce the agents, which are often self-interested, to collaborate to find the optimal system-wide solution.

2.1 The Combinatorial Allocation Problem

The problem that mechanism design usually attempts to solve is the combinatorial allocation problem (CAP), which in our case can also be thought of as a combinatorial auction. The CAP is a resource allocation problem in which agents have non-linear values for bundles of items, and so may value a bundle of two or more goods differently than the sum of their values for the goods that comprise that bundle. In our system, we have a set N of independent, rational, self-interested agents and a set M of items. Each agent j has a valuation function $v_j(S)$ over all possible bundles of items $S \subseteq M$ that could be allocated to that agent. The goal is to determine the allocation that maximizes the overall value to all of the agents, i.e. to maximize

$$V(N) = \sum_j v_j(S_j)$$

where S_j is the bundle of items assigned to agent j in the optimal allocation and $V(N)$ is the total value of the efficient (socially optimal) outcome.

We will make certain simplifying assumptions about agents that are common to mechanism design literature. Each agent's utility function is quasi-linear, such that

$$u_j(S) = v_j(S) - p_j(S), \forall S \subseteq M$$

where u_j is the utility function of the j-th agent, S is a bundle of items, and v_j and p_j are the value and price of bundle S to the j-th agent, respectively. Agents are assumed to be risk-neutral, and there is assumed to be no collusion between agents.

2.2 The CAP as a Model for Real-World Problems

Because of its ability to capture complementarities and other interactions between "items" of any kind, the CAP is a highly descriptive and useful mathematical model for a wide range of computational and real-world scenarios. Parkes [2001] gives a number of examples from a number of domains which require the expressivity of combinatorial valuation functions, including:

Manufacturing Scheduling: Allocating machine time in a factory producing multiple items, each with its own production sequence, quotas, deadlines, and cost/revenue structure.

Supply Chain Coordination: Allocating multiple components to competing manufacturers, each of whom needs *all* of a particular combination of components in order to produce.

Pick-up and Drop-off Routing: Designing routes for a distribution system which has limited transportational resources (i.e. a finite number of delivery trucks) - important to devise efficient combinations of pick-up and drop-off locations.

Travel Packages: Matching combinations of airline tickets, hotel reservations, car rentals, entertainment tickets, etc. to a diverse domain of clients with differing preferences with regard to location, price, entertainment, etc. In particular the travel agent needs to provide a complete combination - outward flight and return flight, hotel for appropriate nights, etc. - if any of these elements are missing the package is not valuable.

Course Registration: Matching a body of students with different academic requirements and course preferences with a set of courses with limited enrollment, ensuring that each student enrolls in an appropriate number of courses and avoids any time-scheduling conflicts.

Dynamic Resource Allocation: For example, the problem of dynamically allocating bandwidth - defining slots of bandwidth by time and size and allotting them to agents with values for bundles of slots.

Distributed Query Optimization: Responding efficiently and effectively to a set of queries with the help of a number of agents with differing expertise and restrictions on communication and/or the response capacity of individual agents.

In addition to these examples, two very prominent real-world cases of the CAP were the subject of the Harvard graduate class Computer Science 286r in the spring of 2004, the aim of which was to design and implement an iterative combinatorial exchange. The two areas of application were the FCC spectrum allocation problem and the problem of scheduling airplane landing and take-off slots at major U.S. airports.

In the 1990s, Congress mandated the FCC to achieve a "value-maximizing" allocation of wireless spectrum to wireless phone companies and other potential users. The affected spectrum was previously allotted by the FCC to potential users on a case-by-case basis. Studies showed that this had resulted in massive inefficiencies in spectrum allocation [Kwerel and Williams, 2000]. The proposed solution was a gradual transition to comprehensive licensing of the spectrum and a market-based allocation system in which spectrum licenses (over time and geographic region) could be bought or sold freely. Significant economic and political issues obstruct an immediate and complete switch to this system, but small progress can be (and has been) made by licensing small blocks of spectrum and releasing them to the public with an initial auction. The valuation of spectrum licenses, over various bandwidths and geographic areas, is quite synergistic - for example, a regional or national company might have a significantly higher value for the bundle of licenses for New York City, Philadelphia, and Washington, D.C. than for each of those licenses individually. Furthermore, given differences in strategy, geographic scope, and technology, the optimal packaging of these licenses is far from agreed upon. Viewing the problem as a combinatorial allocation problem allows the market to effectively value and compare overlapping (and hence mutually exclusive) plans and ultimately to allocate the licenses in the value-maximizing way.

The combinatorial aspect of the airline runway slot scheduling problem arises from the need to pair take-off and landing slots for each flight, subject to flight-time and flight scheduling constraints. Historically the FAA directly allocated slots to each of the major airlines; however, there are compelling economic reasons to transition to a market-based allocation system which would promote much greater efficiency. Like the FCC with regard to the spectrum allocation problem, the FAA is actively encouraging research on a combinatorial auction as a potential mechanism for both the initial transition to a market-based system and a continuous method of allocating take-off and landing slots efficiently.

The examples in this section, drawn from a wide range of problem domains, only begin to demonstrate the breadth of applicability of the CAP. Furthermore, the power of the combinatorial bidding paradigm to express complicated preferences indicates that the CAP may be an effective representation for many unforeseen future problems, as well as those under consideration today. Hopefully this sufficiently motivates the exploration of the CAP and of promising auction methods for its solution.

2.3 Incentive Compatibility

A key goal of mechanism design is *incentive compatibility* or *strategy-proofness*, which can be thought of as ensuring that the interests of the individual agents are aligned with those of the system as a whole. The goal of the system is to compute the optimal allocation given the valuation functions of each of the agents, which requires that each agent report their valuation function accurately. Since each agent is self-interested, if an agent can gain utility by misreporting its valuation function, it will choose to do so. Therefore, we wish to ensure that each agent prefers to report its valuation function truthfully. In game theoretic terms, we wish for truthfulness to be a *dominant strategy* for each agent.

Dominant strategies provide an additional benefit by reducing the *strategic complexity* for the agents, and thus the overall complexity of the problem they face in trying to maximize their utility. If there were no dominant strategy, the agent would have to perform game-theoretic reasoning to determine the best strategy. Even worse, if the exact nature of the other is hidden (which is quite probable in competitive real-world applications), the agent has to reason with incomplete information. Deciding the best strategy could be a very demanding problem computationally, which we can eliminate by providing a mechanism in which the dominant strategy is truthfulness.

A very simple example of an incentive compatible mechanism is the single-item, second-price option, also known as the Vickrey auction. Briefly, each agent's best strategy is to bid their true value for the item, since if they win, they only pay the amount of the second highest bid. The generalization of the Vickrey auction to multiple items is known as the Generalized Vickrey Auction, in the class of Vickrey-Clarke-Groves mechanisms. It is in essence a second price auction - each agent j must pay

$$\begin{aligned} p_j(S_j) &= V(N/j) - \sum_{i \neq j} v_i(S_i) \\ &= V(N/j) - V(N) + v_j(S_j) \end{aligned}$$

where $V(N/j)$ represents the total value of the optimal solution excluding agent j. This means that the utility of the optimal allocation to agent j is equal to the amount by which it increases the social optimum $V(N)$, as can be seen below:

$$
\begin{aligned}
u_j(S_j) &= v_j(S_j) - \left(V(N/j) - \sum_{i \neq j} v_i(S_i) \right) \\
&= \sum_i v_i(S_i) - V(N/j) \\
&= V(N) - V(N/j)
\end{aligned}
$$

Note that to compute the payments for the Generalized Vickrey Auction, the mechanism needs to compute solutions to $V(N/j)$ for all agents $j \in N$. This means that in a system with n agents, implementing the GVA requires solving $n+1$ combinatorial allocation problems of approximately the same size as the original problem.

2.4 Central Agent Mechanisms

The classic VCG mechanism uses a trusted central agent to implement the entire mechanism. The participating agents submit their valuation functions to the center, the center solves for the optimal allocation with all agents (which has value $V(N)$) and the optimal allocations without each agent in turn (which have values $V(N/j)$). It then reports the solutions and the payments back to the participating agents. However, with large numbers of agents and items, solving the necessary optimization problems becomes very intensive computationally - the growth is probably exponential, as the combinatorial allocation problem is NP-hard. The natural approach to this increasing computational time would be parallelization, but a mechanism with all problems being solved by one central agent doesn't easily lend itself to parallelism. This is a clue that perhaps a more fully distributed implementation of the VCG mechanism might be desirable, as it could be more easily run in parallel and thus combat complexity and efficiency issues.

Issues of complexity aside, a broader concern is that a centralized mechanism implementation may not lend itself to certain domains and real-world problems. The classic, centralized VCG mechanism serves its purpose well in some cases but has many issues. We would like to improve the mechanism's complexity and privacy and eliminate the need for a trusted center, without compromising its desirable game-theoretical properties like strategy-proofness. Furthermore, a distributed implementation would be a much better, more natural fit for many problems.

2.5 Distributed Implementations of VCG Mechanisms

One drawback inherent to classic mechanism design is that all decision-making is necessarily centralized. Parkes and Shneidman [2004] motivate and lay out a framework for the exploration of possible distributed implementations of VCG mechanisms. The fusion of mechanism design with algorithms and knowledge from distributed artificial intelligence has the potential to address the great challenge of integrating the cooperative methods of the latter with the self-interested methods of the former. The unification of these disciplines could have strong benefits for the field of multiagent systems as a whole, according to Lesser [1999].

The goal of this exploration, according to Parkes and Shneidman, should be a distributed mechanism implementation that balances good computational properties with good incentive properties. Incentive compatibility in particular is an issue with distributed implementations since the agents have a greater range of actions available to them and therefore more room for strategic action and deviation. Good incentive properties will ensure a successful, or *faithful*, implementation, in that agents choose to be truthful because it is in their best interest. Parkes and Shneidman [2004] propose three principles to guide the design of distributed mechanism implementations: the partition principle, the information-revelation principle, and the redundancy principle. The first two are relevant to the Dantzig-Wolfe auction put forth in this paper. Since the Benders auction is slightly more complicated and less rigid in its implementation at this point, it is less clear how these principles apply. Our ultimate goal is still the same, though, and hopefully the Benders auction can also be implemented in a way that clearly satisfies both principles.

2.5.1 The Partition Principle

Distributing computation in a VCG mechanism allows the agents more latitude for manipulation than in a classic, centralized mechanism. Whereas in a centralized mechanism the agents are only responsible for the private information they reveal to the trusted center, in a distributed implementation the actual winner determination rests on the agents themselves. Therefore, to ensure incentive compatibility, the distributed implementation must be designed such that no agent has an opportunity to deviate *in its computational contribution* in a way that increases its overall utility from the system. The partition principle addresses this concern:

Theorem 1. (Partition Principle) *Consider a distributed implementation d_M of the VCG mechanism in which a canonical distributed algorithm is adopted to solve the set of marginal economies $\{E_N, E_{N-1}, ...\}$. If in d_M computation is partitioned such that the center can correctly solve E_{N-j} whatever the actions of agent j, then d_M is an ex post faithful distributed implementation of the efficient choice and VCG payments. [Parkes and Shneidman, 2004]*

The Dantzig-Wolfe decomposition can be viewed as a distributed optimization algorithm, and this paper will later show how it can be used to solve the main economy E_N and each of the marginal economies $\{E_{N-1}, E_{N-2}, ...\}$. It will be shown later that in doing so the Dantzig-Wolfe auction satisfies the partition principle.

2.5.2 The Information-Revelation Principle

Another way of ensuring that agents do not manipulate the solution is to restrict the action space of agents to actions that reveal private information. The information-revelation principle formalizes this idea, relying on the concept of *information-revelation consistency.* This paper will not discuss consistency rigorously, but will merely suggest that in a Dantzig-Wolfe auction, disallowing agents from changing their preferences enforces consistency of information revelation.

Theorem 2. (Information-Revelation Principle) *The distributed mechanism d_M with consistency-checking is an* ex post *faithful implementation when the only agent actions are information-revelation actions and when the outcome rule of the mechanism is strategy-proof. [Parkes and Shneidman, 2004]*

In the Dantzig-Wolfe auction prices will be broadcast each round, and agents respond with new optimal proposals. A proposal and accompanying valuation information constitute an information-revelation action on the part of an agent. Since these are the only actions the agents undertake, and the optimal global solution is ultimately determined by the central agent with the cumulative information revealed over the course of the auction, the Dantzig-Wolfe auction naturally satisfies the information-revelation principle.

Chapter 3

LP and the Dantzig-Wolfe Decomposition

Linear programming is a technique that lends itself to a vast range of problems, but in particular to large-scale optimization problems. Linear program (LP) solvers have evolved tremendously over time and are now capable of solving problems on a scale perhaps unthinkable to mathematical programmers decades ago. However, solving some linear programs directly is still beyond the capabilities of today's cutting edge LP solvers, suggesting the desirability of a more elegant approach. Moreover, many of these problems have distinctive structure which can be exploited by a more sophisticated approach to the problem. The Dantzig-Wolfe decomposition is one such approach, laid out by Dantzig [1963]. The discussion of decomposition in this chapter will follow that of Bradley, Hax, and Magnanti [1977].

3.1 Linear Programming and the Allocation Problem

The general allocation problem can be formulated easily as an integer program (IP), but solving IPs is NP-hard and thus impractical computationally for moderate and large-sized problems. A natural approach, then, would be to construct the constraint matrix of the IP, but relax the integral requirements on the variables, resulting in a linear program, which is easier to solve. This is referred to as the *LP relaxation* of the IP formulation for the allocation problem.

The difficulty with relaxing the IP to an LP is that it allows for fractional variable values. Since the variables in the program represent whether or not a certain good is allocated to a certain agent, a fractional solution can't have meaning in an allocation problem with goods that are not continuously divisible. Since the combinatorial allocation problem as defined above has indivisible goods, we must find some way to constrain the LP to have only integral solutions. Some particularly tractable instances of the CAP are known to have this property, in particular the assignment problem where agents have unit demand, the problem for which the Dantzig-Wolfe auction will be introduced in Chapter 4.

Ultimately we wish to extend the Dantzig-Wolfe auction beyond the special tractable cases to the entire domain of combinatorial allocation problems. This requires a more sophisticated approach to formulating the LP relaxation, since cases are readily apparent where the simple relaxation of the IP formulation gives rise to non-integral solutions. Fortunately, there has been considerable work done in modifying LPs by adding constraints in order to guarantee that the optimal solutions are integral. In particular, the work of Bikhchandani and Ostroy [2001] is of interest. In their discussion of the package assignment model, they introduce more complicated LP formulations of the CAP that guarantee integral solutions, but moreover have meaningful interpretations from a pricing perspective. The Bikhchandani and Ostroy LP formulations will play a crucial role in the Dantzig-Wolfe auction when the full CAP domain is considered in Chapter 5.

3.2 Decomposition Methods and the Dantzig-Wolfe Decomposition

Some large linear programs have a distinctive constraint structure that can be exploited to solve the programs in an efficient manner. In particular, when a problem consists of independent or nearly independent sub-systems, the idea of breaking the problem down into smaller blocks, solving them separately and integrating them into the larger solution is a promising one. A decomposition method aims to break the problem into several parts, one or more with "easy" constraints and one with the "complicating" constraints. The Dantzig-Wolfe decomposition solicits partial solutions in the form of solutions to the easy subproblems, then solves a restricted master problem that is in essence a weighting problem, determine which proposals to accept and combine based on the complicating constraints that are invisible to the subproblems. Solving the dual of the master problem provides shadow prices on the complicating constraints, which provide informative feedback to the subproblems, who in turn fashion and submit new proposals based on those shadow prices. Eventually, the proposals submitted can be combined into the optimal solution to the overall problem, at which point no new proposals will be more valuable than the cost imposed by the shadow prices. Once no subproblem can submit a proposal the enhances the master problem, it can be proven that the solution to the master problem gives the solution to the original problem.

Consider the following example problem from Bradley et al. [1977]:

$$\max z = c_1 x_1 + \ldots + c_t x_t + c_{t+1} x_{t+1} + \ldots + c_n x_n$$

s.t.

$$\left[\begin{array}{ccccccccccccc} a_{i1}x_1 & + & \ldots & + & a_{it}x_t & + & a_{i,t+1}x_{t+1} & + & \ldots & + & a_{in}x_n & = & b_i & (i=1,...,m) \\ e_{s1}x_1 & + & \ldots & + & e_{st}x_t & & & & & & & = & d_s & (s=1,...,\bar{q}) \\ & & & & & & e_{s,t+1}x_{t+1} & + & \ldots & + & e_{sn}x_n & = & d_s & (s=\bar{q}+1,...,q) \\ & & & & & & & & & & x_j & \leq & 0 & (j=1,...,n) \end{array}\right]$$

This problem has two nearly independent subsystems (those containing the e_{ij} constraints) linked together be a class of overarching constraints (the a_{ij} constraints). To apply decomposition, we will view each of the two subsystems as a separate subproblem, which will each submit their own proposals, and the master problem will solve for the optimal weights to give the proposals it has already received and solicit more proposals if necessary. For a given proposal $X_1 = (x_1, ..., x_t)$ from subproblem 1, define resource coefficients r_{i1} and profit coefficient p_1 as follows:

$$\begin{array}{ccccccc} r_{i1} & = & a_{i1}x_1 & + & \ldots & + & a_{it}x_t & (i=1,...,m) \\ p_1 & = & c_1x_1 & + & \ldots & + & c_tx_t \end{array}$$

so that r_{i1} represents the resource usage of X_1 in the i-th a_{ij} constraint and p_1 represents the value of X_1. For any proposal $X_2 = (x_{t+1}, ..., x_n)$ from subproblem 2, define r_{i2} and p_2 similarly:

$$\begin{array}{ccccccc} r_{i2} & = & a_{i,t+1}x_{t+1} & + & \ldots & + & a_{in}x_n & (i=1,...,m) \\ p_2 & = & c_{t+1}x_{t+1} & + & \ldots & + & c_nx_n \end{array}$$

At any given point in the algorithm, with k proposals from subproblem 1 and l proposals from subproblem 2, the master problem is the following weighting problem:

$$\max z = p_1^1\lambda_1^1 + \ldots + p_1^k\lambda_1^k + p_2^1\lambda_2^1 + \ldots + p_2^l\lambda_2^l$$

s.t.

$$\left[\begin{array}{cccccccccccccc} r_{i1}^1\lambda_1^1 & + & \ldots & + & r_{i1}^k\lambda_1^k & + & r_{i2}^1\lambda_2^1 & + & \ldots & + & r_{i2}^l\lambda_2^l & = & b_i & \pi_i \\ \lambda_1^1 & + & \ldots & + & \lambda_1^k & & & & & & & = & 1 & \sigma_1 \\ & & & & & & \lambda_2^1 & + & \ldots & + & \lambda_2^l & = & 1 & \sigma_2 \\ & & & & & & & & & & \lambda & \geq & 0, & \forall\lambda \end{array}\right]$$

The variables λ_1^j and λ_2^j refer to the weights placed on the j-th proposals from subproblems 1 and 2, respectively. Note that the π_i, σ_1 and σ_2 denote the optimal shadow prices for each of the constraints in the weighting problem, and are obtained by solving the dual to the restricted master problem as expressed above. The next round of proposals X_1 and X_2 will be priced out using these shadow prices:

$$\bar{p}_1 = p_1 - \sum_{i=1}^{m} \pi_i r_{i1} - \sigma_1$$

$$\bar{p}_2 = p_2 - \sum_{i=1}^{m} \pi_i r_{i2} - \sigma_2$$

The shadow prices are reported to each subproblem, who evaluate their new subproblems, based on the latest information. For subproblem 1,

$$v_1 = \max \sum_{j=1}^{t} \left(c_j - \sum_{i=1}^{m} \pi_i a_{ij} \right) x_j$$

$$\text{s.t.}$$

$$\begin{bmatrix} e_{s1}x_1 & + & \dots & + & e_{st}x_t & = & d_s & (s = 1, ..., \bar{q}) \\ & & & & x_j & \geq & 0 & (j = 1, ..., t) \end{bmatrix}$$

and for subproblem 2,

$$v_2 = \max \sum_{j=t+1}^{n} \left(c_j - \sum_{i=1}^{m} \pi_i a_{ij} \right) x_j$$

$$\text{s.t.}$$

$$\begin{bmatrix} e_{s,t+1}x_{t+1} & + & \dots & + & e_{sn}x_n & = & d_s & (s = \bar{q}+1, ..., q) \\ & & & & x_j & \geq & 0 & (j = t+1, ..., n) \end{bmatrix}$$

If $v_1 > 0$, then the optimal proposal from the first subproblem is added to the master problem, and if $v_2 > 0$ then the optimal proposal from the second subproblem is added. If $v_i \leq \sigma_i$, then $\bar{p}_i \leq 0$ for subproblem i, and no new proposal is submitted from that subproblem in the current round. If no new proposal from any subproblem exceeds the cost established by the shadow prices, the algorithm terminates and the optimal solution has been obtained by the final solution to the master weighting problem.

To summarize, the Dantzig-Wolfe decomposition algorithm starts by breaking the original problem into a master problem and one or more subproblems. In each iteration, the master program receives new proposals from one or more of the subproblems. These proposals are incorporated into the restricted master problem, which is solved along with its dual. The shadow prices given by the dual solution form meaningful pricing feedback which is sent to the subproblems, who again respond with new potentially valuable proposals if they are able. Once no more valuable proposals have been received, the algorithm terminates (this will provably happen in a finite number of steps, although exponential in the worst case) and the solution to the master primal problem provably holds the optimal solution for the original problem.

3.3 Economic Interpretation of the Decomposition Procedure

Decomposition has a very natural economic interpretation when the overall problem is viewed as a resource allocation problem and the prices as reflecting the marginal values of each resource. The above concrete example could be seen as the profit maximization problem for a corporation with two subdivisions. Each subdivision has its own internal constraints and knowledge of its valuation function over the possible resource allocations. The corporate headquarters (the master problem) need not have explicit knowledge of the internal constraints or valuation function of any subdivision.

Bradley et al. [1977] point out that in many such situations it would be very costly for the headquarters to gather detailed information about the subdivisions, and furthermore that often the expertise required to make the best local decisions is at the subdivision level, rather than the command level of the headquarters. For these reasons, it is desirable for each subdivision to operate separately and semi-autonomously, while the headquarters is responsible for coordinating the activities of the various subdivisions, in this case by ensuring that the firm's resources are allocated across subdivisions as efficiently as possible.

During the decomposition procedure, information is passed between the master and the subdivisions via *proposals* (from the subdivisions to the master) and *prices* (from the master to the subdivisions). The nature of the dual solution to the master problem is such that the shadow prices on the resource constraints of the master primal problem represent the marginal cost of each resource at the firm level. The shadow prices on the proposal constraints for each subdivision represent the value of the current optimal weighted proposal from that subdivision in the current provisional solution maintained by the master program. Bradley et al. give the following very clear interpretation of the subdivision problem:

$$
\begin{aligned}
\text{(Net profit)} &= \text{(Gross revenue)} - \text{(Resource Cost)} \\
v_1 &= p_1 - \pi_1 r_{11} - \pi_2 r_{21} - \ldots - \pi_m r_{m1} \\
&= \sum_{j=1}^{t} \left(c_j - \sum_{i=1}^{m} \pi_i a_{ij} \right) x_j
\end{aligned}
$$

Here, c_j is the per-unit gross profit for activity x_j. The shadow price π_i is the value of the i-th corporate resource, $\pi_i a_{ij}$ is the cost of resource i for activity j, and $\sum_{i=1}^{m} \pi_i a_{ij}$ is the total corporate resource cost to produce each unit of activity x_j. [Bradley et al., 1977]

Over iterations of the Dantzig-Wolfe algorithm, the resource prices will be adjusted according to the new proposals received by the headquarters. Eventually, the prices will reach an equilibrium, where economically speaking the marginal cost (price) of each resource exceeds the marginal revenue for that resource for all subdivisions. When prices have thus stabilized, no useful new proposals will be generated and the headquarters will allocate resources based on the optimal final allocation.

A competitive auction connotes more adversarial agents than subdivisions of a firm, but in fact an auction interpretation follows quite easily from the decomposition method. Instead of resources, the auctioneer has single indivisible goods, which it wishes to allocate in the socially optimal manner, i.e. in the way that produces the highest cumulative utility to all agents participating in the auction. Prices are announced on items (or bundles, in combinatorial domains), and agents submit bids (proposals) based on those prices. The prices are adjusted via the dual solution to reflect the new best information on the marginal desirability of each good, and the process is repeated until no two agents are willing to meet the price for any conflicted good (or bundle). It is this natural interpretation of the decomposition method that this paper hopes to introduce, and a more concrete discussion begins in the following section.

Chapter 4

A Simple Domain - the Assignment Problem

Ultimately, I will extend the Dantzig-Wolfe auction framework introduced in this section to the generalized combinatorial allocation problem (CAP). However, for clarity I will first present and analyze the auction with a simplified domain - the assignment problem, noted as a tractable instance of the CAP by Bikhchandani et al. [2001], who refer to it as the case of heterogeneous goods and unit demand. The assignment problem is a natural starting point for a linear programming approach to auction design because of one salient feature in particular - the linear program relaxation of the integer program formulation is guaranteed to have integral optimal solutions. This allows for linear programming techniques (in this case the Dantzig-Wolfe decomposition) to be used to optimize the objective function with respect to the constraint matrix, while having faith that there exist optimal solutions that are purely integral and therefore not nonsensical from the point of view of the original problem.

4.1 Definition of the Assignment Problem

The assignment problem is a restricted instance of the combinatorial allocation problem in which the agents have unit demand for goods - this means that an agent's value for a bundle is its highest value for any single item in the bundle (in other words, no agent derives utility from a second item in the assignment problem). Say we have a set M of goods which we wish to assign to a set N of agents. Using v_{ij} to denote the value of agent $j \in N$ for good $i \in M$, we state mathematically that an agent's value for a bundle S is

$$v_j(S) = \min\{v_{ij} : i \in S\}$$

Clearly this is a version of the CAP with a very restricted valuation space available to the agents. In this problem no agent can benefit from being assigned more than one good, hence the name "the assignment problem."

4.2 LP Representation of the Assignment Problem

Assuming then that no agent need be allocated more than one good in an optimal allocation, we can represent the full assignment problem as a linear program as follows:

$$
\begin{aligned}
V(N) = \max \sum_{j \in N} \sum_{i \in M} v_{ij} x_{ij} & \\
\text{s.t.} \quad \sum_{j} x_{ij} \leq 1, \quad \forall i \in M & \qquad (4.1)\\
\sum_{i} x_{ij} \leq 1, \quad \forall j \in N & \qquad (4.2)\\
x_{ij} \geq 0, \quad \forall i, j &
\end{aligned}
$$

The first set of constraints (4.1) restricts each good to be assigned no more than once, and the second set (4.2) constrains each agent to be assigned no more than one good. It is well known that all extreme points of the above LP formulation are integral. Its dual is

$$
\begin{aligned}
\min \sum_{j \in N} \pi_j + \sum_{i \in M} p_i & \\
\text{s.t.} \quad \pi_j + p_i \geq v_{ij}, \quad \forall j \in N, \forall i \in M & \\
\pi_j, p_i \geq 0, \quad \forall j \in N, \forall i \in M &
\end{aligned}
$$

Here the dual variables π_j refer to the profit of agent j, and the dual variables p_i to the price on good i. Note that this does not allow for non-linear prices, and only allows for non-anonymous prices through the π_j term. This is not an issue for the assignment problem, but there exist combinatorial allocation problems such that non-linear and non-anonymous prices may be necessary to reach an optimal solution. This will be discussed at greater length later in this paper.

Another point of interest mentioned by Bikhchandani et al. [2001] is that while multiple solutions to the dual almost certainly exist, the dual solution that maximizes $\sum_{j \in N} \pi_j$ (or equivalently minimizes $\sum_{i \in M} p_i$) yields the Vickrey payments for the optimal allocation. The multiplicity of dual solutions and the preferability of the solution that maximizes agent utility will appear later as important considerations in the Dantzig-Wolfe auction.

4.3 The Dantzig-Wolfe Auction on the Assignment Problem

Having defined the assignment problem as a linear program (and importantly a linear program with integral solutions), I will now outline the application of the Dantzig-Wolfe decomposition to the solution of that linear program, and show how the process can be used and translated into an iterative auction.

4.3.1 The Restricted Master Problem

In formulating the assignment problem for the Dantzig-Wolfe decomposition, we can view the central, coordinating agent (i.e. the winner determination problem) as the master problem and each agent's utility maximization as a subproblem. In each round, the restricted master problem will be as follows:

$$\max \sum_{j \in N} \sum_{k} v_j^k \lambda_j^k$$

$$\text{s.t.} \quad \sum_{j \in N} \sum_{k} r_{ij}^k \lambda_j^k \leq 1, \quad \forall i \in M \tag{4.3}$$

$$\sum_{k} \lambda_j^k \leq 1, \quad \forall j \in N \tag{4.4}$$

$$\lambda_j^k \geq 0, \quad \forall j \in N, \forall k$$

where λ_j^k represents the weight to be placed on the k-th proposal submitted by agent j, and the λ_j^k's are the variables that the master problem is maximized over. Each coefficient v_j^k represents agent j's value for its k-th proposal. The r_{ij}^k's are the *resource coefficients* from the Dantzig-Wolfe master problem. Each r_{ij}^k is defined to be 1 if agent j is allocated good i in its k-th proposal, and zero otherwise. Thus the first set of constraints (4.3) (the resource constraints, one for each good $i \in M$) is analogous to the constraints $\sum_j x_{ij} \leq 1, \forall i \in M$ in the original LP formulation of the assignment problem - simply, that no good can be allocated more than once. The second constraint (4.4) ensures that no agent is given more than one of its proposals.

The dual is analogous to the dual in the original LP formulation:

$$\min \sum_{i \in M} p_i + \sum_{j \in N} \pi_j$$

$$\text{s.t.} \quad \sum_{i \in M} r_{ij}^k p_i + \pi_j \geq v_j^k, \quad \forall j \in N, \forall k$$

$$\pi_j, p_i \geq 0, \quad \forall j \in N, \forall i \in M$$

It is worth noting that in the assignment problem, the dual constraints are quite simple - since agents have unit demand, no agent proposal contains more than one good, so for each constraint one of the r_{ij}^k's will be equal to 1, and the rest will be equal to 0, so each dual constraint will be of the form

$$p_i + \pi_j \geq v_{ij}^k$$

where i is the good that agent j gets in its k-th proposal.

The structure of dual solutions is therefore quite transparent - at the beginning of the decomposition process, when few proposals are on the table and few goods have been bid on by more than one agent, any p_i and π_j such that $p_i + \pi_j = v_i^k j$ will produce a feasible solution. However, as more agents submit proposals for each good, competing bids will place an upper bounds on the profit that an agent can reap from a good (because lower prices would result in overdemand for the good, which we know is not present in the optimal allocation). Again, it may be that the dual solution that maximizes agent profit and minimizes prices could support the quickest possible convergence of the decomposition algorithm.

4.3.2 Initial Solutions

The Dantzig-Wolfe algorithm starts from a minimal set of feasible agent proposals that nevertheless contain enough information for an informative initial solution to the dual and primal problems. I propose the following starting point for the Dantzig-Wolfe auction in the assignment problem. Let proposal 0 exist for all agents, where each agent is allocated no goods and has no value for proposal 0. Then, solicit proposals from each agent, assuming zero prices on all goods ($p_i = 0, \forall i$) and zero shadow prices for all agents ($\pi_j = 0, \forall j$) - assuming myopic best-response, each agent will submit a proposal (bid) for the good it values the most.

The first meaningful version of the master problem will then be complete, and is laid out here. The objective function for the primal problem is:

$$\max \quad v_1^0\lambda_1^0 \quad + \quad v_1^1\lambda_1^1 \quad + \quad v_2^0\lambda_2^1 \quad + \quad v_2^1\lambda_2^1 \quad + \quad \ldots \quad + \quad v_n^0\lambda_n^0 \quad + \quad v_n^1\lambda_n^1$$

and this objective function is optimized subject to the following resource, proposal, and non-negativity constraints:

$$
\left[\begin{array}{ccccccccccccccc}
r_{11}^0\lambda_1^0 & + & r_{11}^1\lambda_1^1 & + & r_{12}^0\lambda_2^0 & + & r_{12}^1\lambda_2^1 & + & \dots & + & r_{1n}^0\lambda_n^0 & + & r_{1n}^1\lambda_n^1 & \leq & 1 \\
 & & & & & & \vdots & & & & & & & & \\
r_{m1}^0\lambda_1^0 & + & r_{m1}^1\lambda_1^1 & + & r_{m2}^0\lambda_2^0 & + & r_{m2}^1\lambda_2^1 & + & \dots & + & r_{mn}^0\lambda_n^0 & + & r_{mn}^1\lambda_n^1 & \leq & 1 \\
 & & & & & & & & & & & & & & \\
\lambda_1^0 & + & \lambda_1^1 & & & & & & & & & & & \leq & 1 \\
 & & & & \lambda_2^0 & + & \lambda_2^1 & & & & & & & \leq & 1 \\
 & & & & & & & & \ddots & & & & & & \\
 & & & & & & & & & & \lambda_n^0 & + & \lambda_n^1 & \leq & 1
\end{array}\right]
$$

$$\lambda_j^k \geq 0, \quad \forall j \in N, \forall k$$

The dual problem has the same objective function in each round:

$$\min \sum_{i \in M} p_i + \sum_{j \in N} \pi_j$$

and this objective function is optimized each round subject to a growing number of constraints, one corresponding to each proposal that an agent has submitted. In the first round, with two proposals per agent, the dual constraint matrix is:

$$
\left[\begin{array}{ccccccccccc}
r_{11}^0 p_1 & + & \dots & + & r_{m1}^0 p_m & + & \pi_1 & & & \geq & v_1^0 \\
r_{11}^1 p_1 & + & \dots & + & r_{m1}^1 p_m & + & \pi_1 & & & \geq & v_1^1 \\
 & & \vdots & & & & & & & & \\
r_{1j}^0 p_1 & + & \dots & + & r_{mj}^0 p_m & + & & \pi_j & & \geq & v_j^0 \\
r_{1j}^1 p_1 & + & \dots & + & r_{mj}^1 p_m & + & & \pi_j & & \geq & v_j^1 \\
 & & \vdots & & & & & & & & \\
r_{1n}^0 p_1 & + & \dots & + & r_{mn}^0 p_m & + & & & \pi_n & \geq & v_n^0 \\
r_{1n}^1 p_1 & + & \dots & + & r_{mn}^1 p_m & + & & & \pi_n & \geq & v_n^1
\end{array}\right]
$$

$$p_i, \pi_j \geq 0, \quad \forall i \in M \forall j \in N$$

To initiate the first round of the auction, the winner determination agent solves the initial primal and dual problems, resulting in an initial feasible allocation, a set of prices p_i on all goods $i \in M$, and a set of agent shadow prices π_j for each agent $j \in N$. These agent shadow prices represent the marginal contribution of agent j to the value of the initial feasible allocation.

With only two proposals per agent, the initial problem is very sparse, and the initial allocation is easy to understand - each good that is requested by at least one agent will be allotted to the agent with the highest value for it, and goods which are not requested by any agent will go unallocated.

Since the primal constraints corresponding to these initially unallocated goods do not constrain the initial solution, the corresponding dual variables will be zero in an optimal dual solution. This means that any good i that is not bid on in the first round will have $p_i = 0$ in the first otherwise non-zero vector of prices. The vector of good prices and agent shadow prices will be communicated to the agents, who will then submit new best-response proposals to the master.

4.3.3 Subsequent Rounds

Each round of the auction will consist of the following steps.

Step 1: Receive new proposals from each agent (if there are no new proposals, then terminate)

Step 2: Incorporate the new proposals into the current formulation of the master primal and dual problems.

Step 3: Solve the master primal and dual problems for the current best-known allocation and a new vector of prices.

Step 4: Report new good prices and agent shadow prices to agents, requesting new proposals.

This process, repeated until no new proposals are submitted from any agent, will terminate with an optimal solution to the complete primal problem, by the completeness of the Dantzig-Wolfe algorithm. In the assignment problem with m goods, since only m proposals are possible from each agent, the maximum number of rounds needed for a solution is m.

From the optimal solution reached after the final round we know the optimal allocation of goods to agents, but we don't yet know the appropriate Vickrey prices to charge them. In simple domains these prices can be simply determined by finding the so-called "optimal" dual solution, but for more complicated problems, that will not be possible, so instead the Dantzig-Wolfe auction, having determined $V(N)$, will proceed to solve each marginal economy and determine the Vickrey payments once it knows $V(N/j)$ for all j. An alternate method of attempting to determine prices from the dual solutions will also be discussed later in this section.

Step 3 has some room for interpretation, in the way that both the primal and dual problems are solved. In fact, solving the primal may not be necessary at each round - the most substantive step each is calculating a new price vector to report to the agents. So, it would be possible for a skeletal implementation of the Dantzig-Wolfe auction to leave solving the restricted primal until the round where no new proposals are submitted by the agents, at which point solving the restricted primal will compute the overall optimal allocation. The computational cost of the auction could be considerably reduced in this way.

There are two reasons why solving the primal each round could be desirable despite the additional computation involved. First, an interim solution to the restricted primal provides a provisional allocation, a "best-so-far" view of which goods will go to which agents. While this information isn't of any strategic use to the agents in this auction, it is conceivable that in a practical implementation the central agent would want to maintain a provisional allocation for bookkeeping or transparency purposes. Second, and more compelling, the value of the current restricted primal solution can be used to solve for the "optimal" dual solution, which may have desirable properties from a pricing perspective. This can be achieved by modifying the dual to maximize $\sum_{j \in N} \pi_j$ and adding the additional constraint that $\sum_{i \in M} + \sum_{j \in N} \pi_j = V(N)$, where $V(N)$ is the value of the current optimal solution to the restricted primal problem.

As mentioned previously, the dual will have a multiplicity of solutions. It may be desirable to modify the version of the dual that is solved each round in order to obtain certain solutions that provide more meaningful price feedback to agents, which could ultimately speed up the convergence of the auction. This will be discussed more thoroughly in Section 5.5.

Finally, a vector of prices is broadcast to the agents, who then formulate and price out new proposals. This vector will consist of a good price p_i for each good and an agent shadow price π_j for each agent.

Note carefully that notation has changed from the original Dantzig-Wolfe example in Chapter 3, where the complicating (resource) constraints in the master problem had shadow prices π_i and the proposal constraints had shadow prices σ_j. Here, the π_j's represent the marginal profit of the agents, so they denote the dual variables associated with the proposal constraints, and the p_i's represent the marginal cost of the goods, so they denote the dual variables associated with the resource (i.e. good/bundle) constraints.

Upon receiving the pricing vector, agents will calculate their optimal proposal (assuming they have already submitted k proposals) under the new good prices as follows:

$$v_j = \max\{v_{ij} - p_i, \quad \forall i\}$$

After determining the optimal proposal, and checking that it is new (if it is a proposal that has already been submitted the agent need not resubmit it), the agent compares the value of the proposal v_j to its proposal shadow price, π_j. If $v_j > \pi_j$, then the proposal is a valid one and the agent submits it as a bid to the center. If $v_j \leq \pi_j$, the proposal is less valuable than agent j's current marginal contribution to the optimal solution, and should be discarded.

4.3.4 Example of the Dantzig-Wolfe Decomposition on the Assignment Problem

Here is a simple example of the Dantzig-Wolfe auction on an assignment problem with three agents, Agents 1, 2, and 3, and three goods, A, B, and C. The valuations are:

i	$\varnothing$	$\{A\}$	$\{B\}$	$\{C\}$
$v_1 i$	0	10	8	1
$v_2 i$	0	6	6	3
$v_3 i$	0	4	12	5

Initial Solutions: Say that each Agent j has submitted the null proposal in Round 0, which will be represented by the weighting variables λ_j^0.

Round 1, Proposals: To begin, the agents assume prices $p_i = 0$ for all goods i, so each submits its value for its highest-valued good, which is Good A for Agents 1 and 2 and Good B for Agent 3. These will each be weighted by the variables λ_j^1 for agent j. The coefficients in the master problem will be $v_1(A) = 10$, $v_2(A) = 6$, and $v_3(B) = 12$, respectively.

Round 1, Master Primal: The master problem in the first round is:

$$\max \quad 0\lambda_1^0 + 10\lambda_1^1 + 0\lambda_2^0 + 6\lambda_2^1 + 0\lambda_3^0 + 12\lambda_3^1$$

$$\text{s.t.} \quad \lambda_1^1 + \lambda_2^1 \leq 1 \tag{4.5}$$

$$\lambda_3^1 \leq 1 \tag{4.6}$$

$$\lambda_1^0 + \lambda_1^1 \leq 1 \tag{4.7}$$

$$\lambda_2^0 + \lambda_2^1 \leq 1 \tag{4.8}$$

$$\lambda_3^0 + \lambda_3^1 \leq 1 \tag{4.9}$$

Constraints (4.5) and (4.6) enforce that Goods A and B be allocated no more than once, respectively (there is currently no proposal for Good C). Constraints (4.7)-(4.9) enforce that no agent be given more than a full unit of weight across all of its proposals.

The optimal solution is $\lambda_1^1 = 1$, $\lambda_2^0 = 1$, $\lambda_3^1 = 1$, and all other $\lambda_j^k = 0$, allocating Good A to Agent 1 and Good B to Agent 3, with total value $10 + 12 = 22$. Note that the null proposals are not strictly necessary, unless we were to make constraints Constraints (4.7)

- (4.9) equality constraints, rather than inequalities. There is no practical difference - for that reason I'll keep the inequalities and omit the null proposals beyond this point.

Round 1, Master Dual: The dual problem of the first round is:

$$\begin{aligned} \min \quad & p(A) + p(B) + p(C) + \pi_1 + \pi_2 + \pi_3 \\ \text{s.t.} \quad & p(A) + \pi_1 \geq v_1(A) \\ & p(A) + \pi_2 \geq v_2(A) \\ & p(B) + \pi_3 \geq v_3(B) \end{aligned}$$

which yields solutions (for example) $p(A) = 6$, $p(B) = 6$, $p(C) = 0$, $\pi_1 = 4$, $\pi_2 = 0$, $\pi_3 = 6$. These prices are reported back to the agents.

Round 2, Proposals: Under the prices $p(A) = 6$, $p(B) = 6$, $p(C) = 0$, each agent solves

$$\max_i \left(v_j(i) - p(i)\right)$$

This gives the best-response proposals as Good B for Agent 1 and Good C for Agents 2 and 3. Their values are $v_1(B) = 8$, $v_2(C) = 3$, and $v_3(C) = 5$, respectively.

Round 2, Master Primal: The Round 2 master problem is:

$$\begin{aligned} \max \quad & 10\lambda_1^1 + 8\lambda_1^2 + 6\lambda_2^1 + 3\lambda_2^2 + 12\lambda_3^1 + 5\lambda_3^2 \\ \text{s.t.} \quad & \lambda_1^1 + \lambda_2^1 \leq 1 && (4.10) \\ & \lambda_3^1 + \lambda_1^2 \leq 1 && (4.11) \\ & \lambda_2^2 + \lambda_3^2 \leq 1 && (4.12) \\ & \lambda_1^0 + \lambda_1^1 \leq 1 && (4.13) \\ & \lambda_2^0 + \lambda_2^1 \leq 1 && (4.14) \\ & \lambda_3^0 + \lambda_3^1 \leq 1 && (4.15) \end{aligned}$$

Constraints (4.10)-(4.12) enforce that Goods A, B and C be allocated no more than once, and constraints (4.13)-(4.15) enforce that no agent be given more than a full unit of weight across all of its proposals.

The optimal solution is $\lambda_1^1 = 1$, $\lambda_2^2 = 1$, $\lambda_3^1 = 1$, allocating Good A to Agent 1, Good C to Agent 2, and Good B to Agent 3, with total value $10 + 12 + 3 = 25$.

Round 2, Master Dual: The dual problem of the second round is:

$$\begin{aligned}
\min \quad & p(A) + p(B) + p(C) + \pi_1 + \pi_2 + \pi_3 \\
\text{s.t.} \quad & p(A) + \pi_1 \geq v_1(A) \\
& p(B) + \pi_1 \geq v_1(B) \\
& p(A) + \pi_2 \geq v_2(A) \\
& p(C) + \pi_2 \geq v_2(A) \\
& p(B) + \pi_3 \geq v_3(B) \\
& p(C) + \pi_2 \geq v_3(A)
\end{aligned}$$

which yields solutions (for example) $p(A) = 6$, $p(B) = 10$, $p(C) = 3$, $\pi_1 = 4$, $\pi_2 = 0$, $\pi_3 = 2$. Again, these prices are reported back to the agents.

Round 2, Proposals: Under the prices $p(A) = 6$, $p(B) = 10$, $p(C) = 3$, each agent solves $\max_i v_j(i) - p(i)$ to find their best-response proposals. Under these prices, it happens that $\max_i v_j(i) - p(i) \leq 0$ for all three agents, so they have no new proposals to submit. This indicates that the algorithm is finished, and the final, optimal allocation from the Round 2 master problem is optimal.

Computing Vickrey prices: In order to calculate Vickrey prices for this allocation, one approach wold be to solve each marginal economy (E_{N-1} without Agent 1 has value $V(N/1)$, etc.) Methods for solving the marginal economies will be discussed later - for now we can tell by inspection that $V(N/1) = 18$, $V(N/2) = 22$, $V(N/3) = 16$. The Vickrey payments are then easily computed:

$$\begin{aligned}
p_1(A) &= V(N/1) - V(N) + v_1(A) = 18 - 25 + 10 = 3 \\
p_2(C) &= V(N/2) - V(N) + v_2(C) = 22 - 25 + 3 = 0 \\
p_3(B) &= V(N/3) - V(N) + v_3(B) = 16 - 25 + 12 = 3
\end{aligned}$$

These payments may seem strangely low, but this is specific to this small example with as many agents as bundles.

4.3.5 Auction Termination and the Calculation of Payments

Once the problem has been solved by the decomposition method, the overall optimal allocation is known to the central agent. However, to ensure that the agents are truthful in their proposal submission (and their reported values in particular), the mechanism must ensure that it implements an incentive compatible outcome, in this case by computing Vickrey prices for each bundle and thus implementing a GVA, which is in *ex post* Nash Equilibrium. There are two methods by which these Vickrey prices could be determined, which are introduced below and will be expanded in the following sections.

One way to determine the prices would be for the central agent to solve each of the marginal economies for the optimal allocation $V(N/j)$, using the same method of iterative price feedback and proposal elicitation as in the main auction. This entails solving n additional problems of similar complexity to the main problem, but the solution to the main economy can be used to start searching for solutions to the marginal economies at a highly advanced point, much nearer in the solution space to the optimal marginal allocation than the very first round with only two proposals from each agent. It may be that, given the information the central agent obtained during the main auction, these additional problems can be solved at a very low computational cost.

A second way to determine the prices, perhaps less clear-cut but potentially more elegant, is to attempt to use all of the cumulative pricing information to determine a set of prices which support the optimal allocation in each of the marginal economies, from which Vickrey prices could easily be determined. Although it is possible that such prices could be found within the history of solutions to the dual over the rounds leading to the main solution, it seems that in some cases additional queries would have to be sent to the agents in the form of price vectors in order to ensure that the competitive equilibrium prices for the optimal allocation also supported the optimal allocation in each marginal economy.

Both methods have their apparent strengths and weaknesses, and deserve further exploration. They are addressed in turn in the following sections.

4.3.6 Computing Payments by Solving Marginal Economies

One way to compute Vickrey prices is to take advantage of the information the central agent already has in terms of agent proposals and valuations and efficiently solve each of the marginal economies. Once $V(N/j)$ is calculated for each agent j, the Vickrey payments, $p_j(N) = V(N/j) - \sum_{i \neq j} v_i(N)$, can be calculated in a straightforward manner.

The prospect of solving n additional combinatorial allocation problems may seem daunting, and moreover crude and excessive for the purpose of computing Vickrey payments; however, upon further inspection it becomes clear that the optimal allocation $V(N)$ provides a greatly advanced starting point for any optimal marginal allocation $V(N/j)$. Over the course of the solution of the main problem, many or most of the highest-value packages will be elicited from each agent, and it seems in most cases, especially those with more than a few agents, that many packages from $V(N)$ will be allocated to the same winner in many of the marginal $V(N/j)$. Therefore, starting the auction algorithm for each marginal economy from the final master problem of the main economy will be far more efficient than solving each marginal economy from scratch using the same method. It may in fact be the case, as indicated by limited experimentation on small test cases, that the number of new proposals elicited during the solution of all n marginal economies is actually much less than the number of proposals elicited during the solution of the main problem.

The original master problem can be easily converted into a master problem for the marginal economy without agent j, in fact by zeroing a single right-hand side constraint in the primal problem (this amounts to removing that row from the primal problem constraint matrix, and removing the corresponding column from the dual constraint matrix and eliminating the associated dual variable). This constraint is the proposal constraint for agent j, which after zeroing the right-hand side will be $\sum_k \lambda_j^k \leq 0$, which along with non-negativity constraints on the λ_j^k effectively remove agent j from any possible allocation. The structure is therefore almost identical to that of the original Dantzig-Wolfe set-up, and its solution is straightforward.

It is here, in the solution of marginal economies, that the partition principle for distributed VCG mechanisms is relevant, and it can be seen that since agent j is unable to influence the outcome of the marginal economy E_{N-j}, the partition principle is satisfied and this distributed implementation of the VCG mechanism is therefore faithful from an incentive standpoint.

4.3.7 Computing Payments by Examining Past Prices

An alternate approach to the straightforward method of solving the primal problem for each marginal economy E_{N-j} is to utilize the pricing history of the main problem to find feasible solutions for the dual problem for each E_{N-j}. After the optimal solution to the main problem is reached, the center has a sequence of pairs $(p, (v, S))$, where p is a vector of prices and (v, S) is a vector of the new proposals received under those prices and the values of those proposals. It may be possible from this record to identify prices $p_i, \forall i \in M$, and agent shadow prices $\pi_k, k \neq j$, that satisfy the dual problem for E_{N-j}. If this is true, then it may be possible for the Dantzig-Wolfe algorithm to terminate with an optimal solution $V(N/j)$ without further queries to agents.

However, it also seems likely that in some cases second-best proposals would not be elicited in the main part of the auction, and that the optimal solution to certain marginal economies would not be possible until these proposals were incorporated into the restricted master program. So, it may be that further rounds will be necessary in any case to accumulate enough information for Vickrey prices to be computed, even after the optimal allocation is known. The best method to pursue Vickrey prices is not clear at this time and warrants further scrutiny.

4.3.8 Price Movement Across Rounds

The idea of designing auctions based on linear programs and primal-dual algorithms is certainly not a new one - Bikhchandani et al. [2001] mention several in their work on linear programming and Vickrey auctions. Crawford and Knoer [1981] first proposed an ascending-price auction based on a primal-dual algorithm that duplicates the outcome of a sealed bid Vickrey auction for the assignment proble. This idea was developed by Demange et al. [1986] and Bikhchandani et al. [2001] detail it in their section on the case of heteregeneous goods and unit demand. More recently, Parkes [2001] developed *i*Bundle, an iterative combinatorial auction that provably terminates with an efficient allocation, without placing any restrictions on agent valuations.

The original Crawford-Knoer algorithm and *i*Bundle are both ascending-price auctions, meaning that the central agent maintains an ask price on each bundle, and possibly for every agent (non-anonymous prices). The Dantzig-Wolfe auction departs from the requirement of ascending prices, in part because it does not allow agents to change their values for a proposal once they have submitted it. The price vector produced by a master dual solution represents a lower bound on the value of each bundle in the optimal allocation, based on the information already held in the master program.

It is an interesting question whether, if the master restricts itself to the dual solution that maximizes $\sum_{j \in N} \pi_j$, the prices generated by the Dantzig-Wolfe auction will be necessarily ascending, or not. This warrants further investigation, and in general a more thorough understanding of the behavior and significance of the prices generated by the dual solutions should be sought.

4.4 Incentive Compatibility and *ex post* Nash Equilibrium

The goal of the Dantzig-Wolfe auction is to implement the efficient outcome for whatever problem it is given, be it the assignment problem or a more general instance of the CAP. Truthful behavior of the agents is a necessary condition for this to be achieved, and this in turn hinges on the incentive compatibility of the mechanism. If the auction can guarantee the efficient outcome and correct Vickrey payments, it will be in *ex post* Nash equilibrium and no agent will be able to benefit from deviating.

Clearly one issue that must be resolved concretely is the question of how best to compute the Vickrey prices after the efficient allocation is determined. However, even once the preferable methodology is chosen, *ex post* Nash equilibrium is still contingent on each agent behaving truthfully and not manipulating the outcome. Since the Dantzig-Wolfe auction is a distributed implementation of a VCG mechanism, the partition principle and the information-revelation principle from Parkes and Shneidman [2004] both apply.

The partition principle is satisfied for both methods of final price determination. In the first, that of solving each marginal economy E_{N-j} directly, the principle is satisfied because agent j is excluded from contributing to the solution of the marginal master problem. In the second, the principle is also satisfied because after the main problem is solved no additional information will be required from any agent j to determine prices that solve the dual problem for E_{N-j}.

The information-revelation principle is also satisfied for both methods of final price determination. The consistency requirement is satisfied because agents are not allowed to modify any proposal that they have already submitted, and all actions available to the agents constitute information-revelation actions. As was noted in Chapter 2, the Dantzig-Wolfe approach lends itself very naturally to an information-revelation-based mechanism, and is therefore very promising as a generalized approach to distributed mechanism design. The next section will extend the auction developed here to the more sophisticated domain of the CAP.

Chapter 5

Full Domain - Dantzig-Wolfe and the CAP

While Chapter 4 showed that the implementation of the Dantzig-Wolfe auction is straightforward for the assignment problem, we wish to generalize it to more difficult cases than the particularly tractable one of agents restricted to unit demand. In this section the methodology introduced in Chapter 4 will be extended to the general combinatorial allocation problem (CAP). However, substantial challenges arise when all manner of agent valuations become admissible. In particular, certain problems give rise to fractional solutions to the linear program relaxation of the integer program representation of the allocation, and while this can be addressed within the framework of the Dantzig-Wolfe auction, significant modifications are required.

5.1 IP Representation of CAP and Simple LP Relaxation

The combinatorial allocation problem is naturally represented as an integer program, similar to the assignment problem. We still have a set M of items and a set N of bidders. Define $v_j(S)$ to be agent j's value for the bundle of items $S \subseteq M$. Bikhchandani et al. [2001] introduce a set of binary variables $y(S,j)$ such that $y(S,j) = 1$ if the bundle $S \subseteq M$ is allocated to agent $j \in N$ and zero otherwise. An important distinction is that if the set of items $\{i, i'\}$ is allocated to agent j, then $y(\{i, i'\}, j) = 1$, but $y(\{i\}, j) = 0$. The problem formulation (as a linear program) is

$$
\begin{aligned}
V(N) = \max \sum_{j \in N} \sum_{S \subseteq M} v_j(S) y(S,j) \\
\text{s.t.} \quad \sum_{S \ni i} \sum_{j \in N} y(S,j) \leq 1, \quad \forall i \in M \qquad (5.1)\\
\sum_{S \subseteq M} y(S,j) \leq 1, \quad \forall j \in N \qquad (5.2)\\
y(S,j) \geq 0 \quad \forall S \subseteq M, \forall j \in N
\end{aligned}
$$

Constraints (5.1) keep goods from being assigned to more than one bundle, and constraints (5.2) prevent agents from being assigned more than one bundle. This formulation is known as LP1, and while it is a natural representation of the problem, it admits fractional solutions, so we must find stronger constraints to produce an LP with integral extreme points.

5.2 Integral LP Representations of CAP

Bikhchandani and Ostroy [2001] create formulations LP2 and LP3 to strengthen the linear program formulation of the CAP. More recently, Bikhchandani et al. [2001] produce an alternate integral LP formulation known as LP4.

We will being by considering LP2 - while it is an intermediate formulation and still admits fractional solutions, it is a building block for stronger formulations and is worth introducing. Let Π be the set of all possible partitions of the set M of items, and for any $\sigma \in \Pi$, let $S \in \sigma$ signify that the bundle $S \subset M$ is part of the partition σ. LP2 introduces variables z_σ, where $z_\sigma = 1$ if the partition σ is chosen and $z_\sigma = 0$ otherwise. The LP2 formulation is

$$
\begin{aligned}
V(N) = \max \sum_{j \in N} \sum_{S \subseteq M} v_j(S) y(S,j) \\
\text{s.t.} \quad \sum_{S \subseteq M} y(S,j) \leq 1, \quad \forall j \in N \qquad (5.3)\\
\sum_{j \in N} y(S,j) \leq \sum_{\sigma \ni S} z_\sigma, \quad \forall S \subseteq M \qquad (5.4)\\
\sum_{\sigma \in \Pi} z_\sigma \leq 1 \qquad (5.5)\\
y(S,j), z_\sigma \geq 0 \quad \forall S \subseteq M, \forall j \in N, \forall \sigma \in \Pi
\end{aligned}
$$

where the first set of constraints (5.3) limits each agent to one bundle, the second set (5.4) limits bundles to those present in the chosen partition, and the third constraint (5.5) limits the solution to a single partition. LP2 is stronger than LP1, but still admits fractional solutions in certain cases.

5.2.1 Bikhchandani and Ostroy - the LP3 Integral Formulation

In order to further strengthen the LP relaxation, Bikhchandani and Ostroy [2001] extend the partition variables to include every possible agent-partition. Denote the set of all such partitions Γ, and an agent partition $\mu \in \Gamma$ consists of both a partition of items into bundles and an assignment of each bundle to a specific agent. $S^j \in \mu$ means that agent j receives bundle S in agent-partition μ. Let $\delta_\mu = 1$ if the agent-partition $\mu \in \Gamma$ is selected, and zero otherwise. Now we have the building blocks for the LP3 formulation, below:

$$
\begin{aligned}
& V(N) = \max \sum_{j \in N} \sum_{S \subseteq M} v_j(S) y(S,j) \\
\text{s.t.} \quad & \sum_{S \subseteq M} y(S,j) \leq 1, \quad \forall j \in N && (5.6) \\
& y(S,j) \leq \sum_{\mu \ni S^j} \delta_\mu, \quad \forall j \in N, \forall S \subseteq M && (5.7) \\
& \sum_{\mu \in \Gamma} \delta_\mu \leq 1 && (5.8) \\
& y(S,j), \delta_\mu \geq 0 \quad \forall S \subseteq M, \forall j \in N, \forall \mu \in \Gamma
\end{aligned}
$$

Here the first set of constraints (5.6) again limits each agent to one bundle, the second (5.7) only allows bundles to be assigned as per the chosen agent-partition, and the third (5.8) limits the solution to a single agent-partition.

Define for each constraint from the first set (5.6) the dual variable π_j, with each from the second set (5.7) the variable $p_j(S)$, and with the constraint on agent-partitions (5.8) the variable π^s. These dual variables should be familiar - π_j and π^s represent the surpluses of the agents and the seller, respectively, and $p_j(S)$ gives a non-anonymous price for bundle S to agent j. The dual, which will be referred to as DLP3, is

$$
\begin{aligned}
& \min \sum_{j \in N} \pi_j + \pi_s \\
\text{s.t.} \quad & p_j(S) + \pi_j \geq v_j(S) \quad \forall j \in N, \forall S \subseteq M \\
& -\sum_{S^j \in \mu} p_j(S) + \pi_s \geq 0, \quad \forall \mu \in \Gamma \\
& p_j(S), \pi_j, \pi^s \geq 0 \quad \forall j \in N, \forall S \subseteq M
\end{aligned}
$$

Demonstrating the integrality of LP3 is straightforward [Bikhchandani et al., 2001]. It is not without its drawbacks, however, not the least of which is the astronomical number of variables - a simple calculation gives that the number of possible agent-partitions is m^n. While the Dantzig-Wolfe approach does not require all of these variables to be enumerated, it is still tempting to search for a more compact formulation.

5.2.2 Bikhchandani et al - the LP4 Integral Formulation

Bikhchandani et al. [2001] give a new integral LP formulation LP4 with fewer variables than LP3. Recall the set Π of anonymous partitions. Let $y^\sigma(S,j) = 1$ if partition σ is chosen from Π and $S \in \sigma$ is given to agent j, and zero otherwise. Recalling the variables z_σ from LP2, we assemble LP4:

$$V(N) = \max \sum_{\sigma\in\Pi}\sum_{j\in N}\sum_{S\subseteq M} v_j(S) y^\sigma(S,j)$$

$$\text{s.t.} \quad \sum_{S\in\sigma} y^\sigma(S,j) \le z_\sigma, \quad \forall j \in N, \forall \sigma \in \Pi \tag{5.9}$$

$$\sum_{j\in N} y^\sigma(S,j) \le z_\sigma, \quad \forall S \in \sigma, \forall \sigma \in \Pi \tag{5.10}$$

$$\sum_{\sigma\in\Pi} z_\sigma \le 1 \tag{5.11}$$

$$\sum_{\sigma\in\Pi}\sum_{S\in\sigma} y^\sigma(S,j) \le 1, \quad \forall j \in N \tag{5.12}$$

$$y^\sigma(S,j), z_\sigma \ge 0 \quad \forall S \subseteq M, \forall j \in N, \forall \sigma \in \Pi$$

The first set of constraints (5.9) limits each agent to one bundle from a partition σ if that partition is chosen and none otherwise. The second set (5.10) mirrors the first and limits each bundle to be assigned to only one agent if that bundle is in the partition σ and zero otherwise. The third constraint (5.11) limits the solution to a single partition σ, and the fourth set (5.12) is in fact redundant with the previous constraints, but generates dual variables that correspond to agent j's marginal product.

To formulate DLP4, we define μ_j^σ, w_S^σ, π^s and π_j to be the dual variables associated with the first, second, third, and fourth classes of primal constraints, respectively. DLP4 is:

$$\min \sum_{j\in N} \pi_j + \pi_s$$

$$\text{s.t.} \quad \pi^s \ge \sum_{S\in\sigma} w_S^\sigma + \sum_{j\in N} \mu_j^\sigma \quad \forall \sigma \in \Pi$$

$$\pi_j + \mu_j^\sigma + w_S^\sigma \ge v_j(S) \quad \forall j \in N, \forall S \in \sigma, \forall \sigma \in \Pi$$

$$\mu_j^\sigma, w_S^\sigma, \pi_j, \pi^s \ge 0 \quad \forall j, S, \sigma$$

The π_j and π^s dual variables retain their familiar interpretations as bidder and seller surplus. To obtain bundle-specific prices, we can set

$$p_j(S) = \min_{\sigma\in\Pi}(\mu_j^\sigma + w_S^\sigma)$$

This allows us to break the agents' payments into a non-anonymous component μ_j^σ and a non-linear component w_S^σ. [Bikhchandani et al., 2001]

Bikhchandani et al. also give a proof that the LP4 formulation also has integral solutions.

LP4 is uses anonymous partition variables z_σ rather than non-anonymous variables δ_μ, which are more compact but nevertheless still exponential. This gain is somewhat offset by the fact that the agent-bundle variables $y^\sigma(S,j)$ must be partition-specific as well, adding exponential size in that dimension. The exact trade-off is unclear and merits experimental examination.

In summary, due to Bikhchandani and Ostroy [2001] and Bikhchandani et al. [2001] we have at our disposal LP3 and LP4, two LP formulations of the CAP that do not admit fractional solutions. Both formulations should be explored for use in the Dantzig-Wolfe auction.

5.3 The Dantzig-Wolfe Auction on LP3

It is not immediately clear which of the two integral LP formulations of the CAP is preferable for use in a Dantzig-Wolfe auction. One apparent advantage of LP3 is that bundle prices $p_j(S)$ spring more readily from the its dual, DLP3, than they do from LP4.

As with the assignment problem, for the restricted master program we introduce the weight variables λ_j^k, each of which denotes the weight assigned to the k-th proposal from the agent j in the master problem. Recall that agent j has value v_j^k for proposal λ_j^k. Furthermore, analogous to the original LP3 formulation, the restricted master will have variables δ_μ corresponding to each agent-partition μ, although these variables do not appear in the objective function of the weighting problem. Define $\lambda_j^k \in \mu$ to mean that the bundle in agent j's k-th proposal is assigned to agent j in agent-partition $\mu \in \Gamma$. The restricted master problem for the LP3 formulation is then given by

$$\max \sum_{j \in N} \sum_k v_j^k \lambda_j^k$$

$$\text{s.t.} \quad \lambda_j^k \leq \sum_{\mu \ni \lambda_j^k} \delta_\mu, \quad \forall j \in N, \forall k \tag{5.13}$$

$$\sum_{\mu \in \Gamma} \delta_\mu \leq 1 \tag{5.14}$$

$$\sum_k \lambda_j^k \leq 1, \quad \forall j \in N \tag{5.15}$$

$$\lambda_j^k, \delta_\mu \geq 0 \quad \forall j \in N, \forall k, \forall \mu \in \Gamma$$

The dual of the master problem is identical to DLP3 except that not all bundles have variables $p_j(S)$ - only those that appear as proposals in the primal problem do. Let Λ_j denote the set of proposals submitted by agent j, and say $S \in \Lambda_j$ if bundle S appears in a proposal already submitted by agent j. The dual price variable $p_j(S)$ is generated by the constraint from (5.13) corresponding to agent j and the proposal k in which agent j proposed the bundle S. The dual payoff variables π^s and π_j correspond with constraints (5.14) and (5.15), respectively. The dual is then

$$\begin{aligned} \min \sum_{j \in N} \pi_j + \pi_s \\ \text{s.t.} \quad p_j(S) + \pi_j \geq v_j(S) \quad \forall j \in N, \forall S^j \in \Lambda_j \\ - \sum_{S^j \in \mu, \Lambda_j} p_j(S) + \pi_s \geq 0, \quad \forall \mu \in \Gamma \\ p_j(S), \pi_j, \pi^s \geq 0 \quad \forall j \in N, \forall S^j \in \Lambda_j \end{aligned}$$

with variables p_i, π_j and π^s that can easily be interpreted as bundle prices and agent surpluses.

Each round, each agent will receive an updated vector of non-anonymous prices $p_j(S)$ and payoffs π_j, although only the prices are necessary for the Dantzig-Wolfe subproblem. The agents price out potential proposals and determine the best proposal by solving

$$v_j^k = \max_S (v_j(S) - p_j(S))$$

If $v_j^k \geq 0$, then there is a valid proposal to be made and the agent submits that proposal $(S, v_j(S))$ to the center, where it will be incorporated and given the weighting variable λ_j^k in the master problem. If, on the other hand, $v_j^k < 0$, the best-response proposal is not an improvement on agent j's current best proposal and the agent need not report anything for the round.

This formulation is rigorous and will have an integral solution, and a Dantzig-Wolfe auction on it would terminate in a finite number of iterations. However, the number of variables is immense - the number of δ_μ's is m^n - so conducting the auction with this formulation may require considerable refinement in practice. The key to practicality will be reducing the number of variables that must be explicitly stated in the master program, as well as taking advantage of prior information to solve each addition step of the master program, which will be discussed more in Section 5.6. In particular, it may be desirable to generate and consider the non-anonymous partition variables δ_μ only as they are required by the current set of agent-proposals under consideration. While the number of δ_μ variables thus generated will still grow very quickly as more and more proposals are submitted, this is strongly preferable to representing all m^n of them through the entire computation.

5.4 The Dantzig-Wolfe Auction on LP4

This section will present an alternate version of the Dantzig-Wolfe auction, based on the Bikhchandani et al. LP4 formulation. The restricted master program will use the weight variables $\lambda_j^k(\sigma)$, each of which denotes the weight assigned to the k-th proposal from the agent j in the master problem, under the anonymous partition of goods σ. Agent j has value v_j^k for proposal $\lambda_j^k(\sigma)$, for all partitions σ - we are assuming that an agent's value for a bundle does not change based on how the other goods are partitioned or allocated. Let $\lambda_j^k(\sigma) \sim S$ indicate that agent j's k-th proposal is a proposal for bundle S.

It may seem strange to have $\lambda_j^k(\sigma)$ weighting variables for each partition σ, all with identical values v_j^k, but this redundancy in LP4 is the mechanism by which solution integrality is ensured. Glancing at the original LP4 formulation, we can see how the $\lambda_j^k(\sigma)$ variables in the restricted master problem have replaced the agent-, bundle-, and partition-specific variables $y^\sigma(S, j)$ in the original.

Analogous to the original LP4 formulation, the restricted master will have variables z_σ corresponding to each anonymous partition σ which do not appear in the objective function of the weighting problem. The restricted master problem is then given by

$$\max \sum_{j \in N} \sum_k v_j^k \lambda_j^k$$

$$\text{s.t.} \quad \sum_k \lambda_j^k(\sigma) \leq z_\sigma, \quad \forall j \in N, \forall \sigma \in \Pi \tag{5.16}$$

$$\sum_{j \in N, \lambda_j^k(\sigma) \sim S} \lambda_j^k(\sigma) \leq z_\sigma, \quad \forall S \in \sigma, \forall \sigma \in \Pi \tag{5.17}$$

$$\sum_{\sigma \in \Pi} z_\sigma \leq 1 \tag{5.18}$$

$$\sum_k \sum_\sigma \lambda_j^k(\sigma) \leq 1, \quad \forall j \in N \tag{5.19}$$

$$\lambda_j^k(\sigma), z_\sigma \geq 0 \quad \forall j \in N, \forall k, \forall \sigma \in \Pi$$

The dual is very similar to the original, DLP4. It has four classes of variables: μ_j^σ, corresponding with primal constraints (5.16); w_S^σ, corresponding with constraints (5.17); π^s, corresponding with constraint (5.18); and π_j, corresponding with constraints (5.19). The dual formulation is:

$$\min \sum_{j \in N} \pi_j + \pi_s$$

$$\text{s.t.} \quad \pi^s \geq \sum_{S \in \sigma} w_S^\sigma + \sum_{j \in N} \mu_j^\sigma \quad \forall \sigma \in \Pi \tag{5.20}$$

$$\pi_j + \mu_j^\sigma + w_S^\sigma \geq v_j(S) \quad \forall j \in N, \forall S \sim \lambda_j^k(\sigma), \forall \sigma \in \Pi \tag{5.21}$$

$$\mu_j^\sigma, w_S^\sigma, \pi_j, \pi^s \geq 0 \quad \forall j, S, \sigma$$

where constraints (5.20) correspond with the z_σ primal variables and the constraints (5.21) correspond with the $\lambda_j^k(\sigma)$ variables. To offer an intuitive interpretation, the constraints (5.20) require that the seller's profit exceed the minimum for each possible partition. When the dual is solved, the constraints from (5.20) which are binding at the solution will correspond to the partitions σ which can give optimal allocations. The constraints (5.21) ensure that the combination of prices and payoffs exceeds agents' values across all submitted proposals and possible partitions.

Note one slight difference with DLP4 - the constraints (5.21) are only needed for agent-bundles in a partition that have been proposed by those agents, rather than all agent-bundles possible in this partition. This is the meaning of $\forall S \sim \lambda_j^k(\sigma)$ - for all bundles contained in a proposal from some agent j in some round k. There will be considerably fewer constraints in (5.21) than there were in the original dual, where there was one for each $S \in \sigma$.

The agent subproblem is the same as in the LP3 auction: each agent will receive an updated vector of non-anonymous prices $p_j(S)$ in each round, and price out its best-response proposal by solving

$$v_j^k = \max_S \left(v_j(S) - p_j(S)\right)$$

If $v_j^k \geq 0$, then there is a valid proposal to be made and the agent submits that proposal $(S, v_j(S))$ to the center, which will then be associated with the weighting variables $\lambda_j^k(\sigma)$ in the master problem. Otherwise (if $v_j^k < 0$) there is no good *new* proposal for the agent to make in this round.

One disadvantage of LP4, mentioned earlier, is that the dual does not directly yield a single set of non-anonymous prices $p_j(S)$. Rather, the prices must be extrapolated from the partition-specific μ_j^σ and w_S^σ variables. In Section 5.2.2 it was proposed that this be done by solving

$$p_j(S) = \min_{\sigma \in \Pi} (\mu_j^\sigma + w_S^\sigma)$$

This pricing structure is very flexible - it can allow for both non-anonymous and non-linear prices if necessary. The downside is the additional computational burden imposed, if we wish to solve thusly for $p_j(S)$ for all agent-bundle combinations.

Like the Dantzig-Wolfe auction on LP3 proposed in the Section 5.3, the LP4 version as formulated here is rigorous and will terminate in a finite number of steps with an integral solution. Again, program size is a concern, given the exponential numbers of variables floating around. Two ideas to streamline the process jump to mind immediately. The first is to generate and represent the z_σ variables only as needed by the set of agent proposals under consideration, analogous to generating the δ_μ as proposed in the previous section. The second is to try to harness the parallel structure of the $\lambda_j^k(\sigma)$ weighting variables across partitions σ (recall that they all have identical value v_j^k) in order to consider fewer variables and thus solve the problem more efficiently. This would require a deeper practical understanding of the structure of the dual and would likely require substantial empirical work to explore and verify.

Brief Summary

It is clearly possible to translate advanced LP relaxations of the integer program representations of the CAP into the master problem and subproblems required for decomposition to take place. A theoretical Dantzig-Wolfe auction for the full combinatorial valuation domain is near realization. However, the exponential number of variables required to constrain the LP relaxation to integral solutions pose an enormous computational challenge, and the theoretical auction hinted at here may not be capable of addressing large problems in practice. The key to a smaller, more practical implementation of the Dantzig-Wolfe auction is a deeper understanding of the integral LP formulations, how they can be represented more compactly, and the way in which their primal and dual solutions interact with the master and subproblems over the course of a Dantzig-Wolfe optimization.

5.5 Multiple Dual Solutions

We have seen, first in Section 4, that the solution to the master dual program at each round of the bidding may not be unique - on the contrary, is seems that a continuum of dual optimal solutions will often be available, taking advantage of an equal tradeoff in the dual constraint matrix between good or bundles prices p_i and agent shadow prices π_j. This adds a degree of freedom to the progression of the auction - the algorithmically correct path taken by the central agent is not uniquely defined even for a specific instance of the CAP. The Dantzig-Wolfe algorithm provably terminates with the optimal solution, so the auction will have an efficient outcome, whatever dual solutions are chosen; however, it would be best to thoroughly understand the behavior of the master dual solutions, and the impact of the specific solutions chosen in any instance on the path of the auction. One conceivable scenario is that restricting the dual problem to a certain dual solution supports uniformly faster convergence of the auction; alternately, it is possible that the specific dual solution chosen does not have any effect on convergence time, or that its effect on convergence is not uniform, and there is no incentive for the designer of the auction to try to restrict the dual solution in any way.

Bikhchandani et al. [2001] find that for the purposes of computing Vickrey payments from the dual solution (in the tractable cases where that is possible), the preferred dual solution is that which maximizes profit to the bidders, $\sum_{j \in N} \pi_j$. The can be understood intuitively by thinking that the play between optimal dual solutions is in the tradeoff between the p_i's and the π_j's, and that in a terminal state of the Dantzig-Wolfe auction the p_i's must be at or greater than the "second prices" of the goods or bundles to which they refer (otherwise valuable proposals would still be found by at least one agent and the auction would not have terminated). Implementing the Vickrey auction requires the second-price principle, or else the auction loses its *ex post* strategy-proofness. Maximizing profit to the bidders forces the prices to the minimum of their acceptable range in the dual solution, which corresponds to the second price.

The way in which an inferior dual solution to the master in a given round could be detrimental to convergence is that it could encourage the submission of proposals which are not part of the efficient outcome and would not be elicited under a different dual solution. Prices in the Dantzig-Wolfe auction represent lower bounds on the values of goods or bundles, and intuitively tighter lower bounds could eliminate extraneous bidding in some cases. The agent-optimal dual solution has been shown to be of theoretical interest in other situations and the question of whether it is provides tighter lower bounds through prices and is thus preferable in the Dantzig-Wolfe auction is worthy of investigation.

If it turns out to be the case that any dual solution to the master problem is equally valid from a convergence standpoint, there arises a question of whether it is actually necessary to solve the master primal problem at every round of the auction, or even at all before all of the worthy bids have been submitted. The current value of $V(N)$ is useful in specifying a dual solution - as we have seen earlier, this is achieved by modifying the objective function of the dual and adding a dual constraint to the effect of $\sum_{i \in M} p_i + \sum_{j \in N} \pi_j = V(N)$. If no specific dual solution need be specified, though, the only benefit to solving the master primal during intermediate rounds of bidding is maintaining a provisional feasible allocation, which does not provide any useful information to the agents. Therefore, if the auction converges equally fast regardless of which dual solutions are chosen, the computational aspect could be sped up considerably by only solving for the dual solution to the master until no new proposals are submitted and the optimal solution is contained in the restricted master primal problem.

5.6 Linear Program Similarities

The main computational demand of the Dantzig-Wolfe auction is solving the primal and dual linear programs after each round of new agent proposals. In settings with large numbers of agents and goods, both the primal and dual formulation will be very large LPs, that must be solved many times. Furthermore, after the main economy $V(N)$ is solved, each marginal economy $V(N/j)$ must also be solved, so running the entire auction requires solving $n+1$ problems, each requiring many iterations of LP solving.

5.6.1 Similarities Between Consecutive Primal and Dual Formulations

The computational situation may be greatly mitigated, however, by the fact that the master primal and dual problems differ only slightly from round to round. Each round, one or more new proposals are submitted from agents and incorporated as columns and variables λ into the master primal program and as rows (i.e. constraints) into the master dual program. The master programs maintain the same basic structure as in the previous round, and, even more promisingly, all of the variables and coefficients from the previous round, which may dwarf the new proposals, especially late in the auction. It seems intuitive that the solutions to the master primal and dual problems for a given iteration will incorporate much, even most of the solutions from the previous round. Thus, a simple technique to take advantage of this similarity would be to start searching for solutions to the master primal and dual problems in round k from the optimal solutions to those problems in round $k-1$. If the simplex method were being used to solve these LPs, this might entail seeding the beginning tableau for the k-th round solution with the final basis from the round before.

It is also possible that other techniques exist that would allow for the solutions to the master problem in round $k+1$ to be determined very efficiently, given the solutions to rounds 1 through k. If effective techniques were known, they could greatly enhance the practicality and computational speed of a Dantzig-Wolfe auction.

5.6.2 Current Research

To the best of my knowledge, there is no current research in the area of optimizing the solution time of a sequence of similar linear programs. It seems an interesting problem with a wide range of applications in decomposition and elsewhere, and worth focused pursuit if it is indeed the case that it has not been investigated thoroughly before.

5.7 Assessing the Dantzig-Wolfe Auction

Although the Dantzig-Wolfe auction has not been fully fleshed out, it is possible to have some idea of how it compares to other implementations of the Generalized Vickrey Auction, in particular other distributed implementations. In terms of incentive compatibility, the Dantzig-Wolfe auction comfortably satisfies the partition principle and the information-revelation principle, and therefore as long as it implements a VCG mechanism with payments $p_j = V(N/j) - \sum_{i \neq j} v_i(N)$ it is a faithful distributed implementation, avoiding the problem of manipulation by agents effectively. We can therefore say that the Dantzig-Wolfe auction as a distributed implementation of the VCG mechanism will surely meet the second main goal of Parkes and Shneidman [2004], namely that it has good incentive properties.

The current picture of the computational properties of the Dantzig-Wolfe auction, on the other hand, is much more cloudy. The size of the linear programs that the central winner determination agent needs to solve could quickly grow beyond the capacity of current methodology and hardware. In particular, the problem formulations for the full combinatorial allocation problem, based on the LP3 and LP4 linear relaxations, utilize an exponential number of variables in the size of the problem, making a complete solution of the problem very difficult. The key to addressing this issue is by carefully formulating the restricted master program so that it is as compact as possibly while still faithfully implementing the strengthened LP formulation of the original CAP.

A second computational consideration, discussed in Section 5.6, is the solution method used to solve the master linear program round after round. There are tremendous similarities between consecutive iterations of the master program and if these are harnessed by the solution technique huge computational benefits would accrue. The state of current research on such techniques is unclear and bears further investigation.

A final interesting question about the Dantzig-Wolfe auction regards the behavior of prices. Many auction implementations of the VCG mechanism (such as *i*Bundle [Parkes, 2001]) have ascending prices over the course of the auction. The Dantzig-Wolfe auction, on the other hand, is certainly not constrained to ascending prices - prices are free to fluctuate up or down over rounds depending on the specific solution to the master dual problem. However, the prices may operate as lower bounds on current perceived bundle values for each round, and as more competitive proposals are submitted these values will rise, so it may be that prices do end up weakly ascending on some forms of the Dantzig-Wolfe auction. This may happen particularly if the central agent prefers a specific optimal solution of the dual problem, such as the bidder-optimal solution, for convergence reasons. Further exploration and analysis is necessary to more thoroughly comprehend the behavior and significance of the prices in the Dantzig-Wolfe auction.

Chapter 6

The Benders Decomposition

The overarching goal of this thesis is to create new ways of implementing combinatorial auctions by representing them as linear programs and applying two well-known decomposition techniques to them. The center and the agents will interact by exchanging information and over time progress toward an optimal allocation. In the Dantzig-Wolfe decomposition, the information flow is such that the center announces a set of prices and the agents submit in response possible allocations supported by those prices. Benders decomposition is in many ways the opposite approach to Dantzig-Wolfe, and it is thus not surprising that the center-agent information flow is reversed: when we apply the Benders decomposition to the CAP, it will be the center that is announcing allocations (or, more accurately, partitions) and the agents who are responding with price information. The reasons for this will become more clear once we have thoroughly examined the process of applying the Benders decomposition to various LP formulations of the CAP.

6.1 The Benders Decomposition Procedure

The Benders decomposition is a decomposition method that is often used to attack problems which have both an integral and a continuous component. It does this by separating the problem into two parts: a linear subproblem with only continuous variables, and an integral master problem with the complicating integral variables and their constraints. The subproblem can be solved easily using established LP methods - the Benders strategy is to fix the integral variables, solve the dual of the subproblem, and from that dual solution generate constraints to add to the master problem, until the re-written master problem is sufficiently constrained to yield an optimal solution. This discussion of Benders follows those in Kameshwaran [2004] and Lasdon [2002].

In general, the Benders decomposition could be used on a problem of the following form:

$$\begin{aligned} \text{MIP}: \quad & \max \ c\mathbf{x} + f\mathbf{y} \\ \text{s.t.} \quad & A\mathbf{x} + B\mathbf{y} \le d \\ & \mathbf{x} \in R^m_+, \mathbf{y} \in \{0,1\}^n \end{aligned}$$

where c, f, A, B, and d are coefficient matrices of appropriate dimensions. The $\mathbf{x}$ variables are continuous and the $\mathbf{y}$ variables integral. In Benders we will rewrite the MIP as the Benders master problem MP with a single continuous variable, z. To construct MP we will also reformulate the constraints using the Benders subproblem, which is the dual of the linear program when the integer variables are fixed. Consider the problem when the integral variables $\mathbf{y}$ have been fixed at $\bar{\mathbf{y}}$:

$$\begin{aligned} \text{SP}(\bar{\mathbf{y}}): \quad & \max \ c\mathbf{x} + f\bar{\mathbf{y}} \\ \text{s.t.} \quad & A\mathbf{x} \le d - B\mathbf{y} \\ & \mathbf{x} \in R^m_+ \end{aligned}$$

Its dual is the Benders subproblem, DSP$(\bar{\mathbf{y}})$.

$$\begin{aligned} \text{DSP}(\bar{\mathbf{y}}): \quad & \min \ \mathbf{u}(d - B\mathbf{y}) + f\bar{\mathbf{y}} \\ \text{s.t.} \quad & \mathbf{u}G \ge c \\ & \mathbf{u} \in R^l_+ \end{aligned}$$

We define $E = \{(\mathbf{u})^e\}$ to be the set of extremal points of the polyhedron DSP$(\bar{\mathbf{y}})$. Note, though, that this polyhedron is independent of $(\bar{\mathbf{y}})$. Using E we can reformulate the original problem to create the Benders MP:

$$\begin{aligned} \text{MP}: \quad & \max \ z \\ \text{s.t.} \quad & z \le f\mathbf{y} + \mathbf{u}^e(d - B\mathbf{y}), \quad e \in E \\ & z \in R^1_+, \mathbf{y} \in \{0,1\}^n \end{aligned}$$

The constraints are the dual cuts generated by the extreme points. Lasdon [2002] and others have proven the equivalence of MP to the original MIP.

Since there are generally a large number of extremal points of the Benders subproblem, MP will have a correspondingly large number of constraints. However, the Benders decomposition takes advantage of the fact that only a few of these constraints will be binding at the optimal solution and generates them as needed, rather than enumerating all of them. The goal of Benders is to solve the modified master problem, MP2, with a subset of cuts from E.

To begin Benders, MP2 is solved with $E = \varnothing$ to obtain a feasible $\bar{\mathbf{y}}$. Then we solve the subproblem DSP($\bar{\mathbf{y}}$) to obtain a dual cut. We add this cut to E and solve the new MP2 to obtain a better $\bar{\mathbf{y}}$. This process of solving in turn the subproblem DSP($\bar{\mathbf{y}}$) and the modified master problem MP2 is continued until E contains a sufficient set of cuts and the optimal solution is obtained (we can tell because the values of DSP($\bar{\mathbf{y}}$) and MP2 will converge).

Here is an outline of the Benders algorithm:

Step 1: $E = \varnothing$. Choose an initial feasible $\bar{\mathbf{y}}$ by solving MP2 with no complicating constraints. (If MP2 is infeasible than there is no feasible solution, so STOP).

Step 2: Solve the Benders subproblem DSP($\bar{\mathbf{y}}$) and obtain a solution $\mathbf{u}^e$ with objective value DSP($\bar{\mathbf{y}}$).

Step 3: Add the dual solution $\mathbf{u}^e$ to E.

Step 4: Call MP2, which gives a new solution $\bar{\mathbf{y}}$ with objective value z. If $z = \text{DSP}(\bar{\mathbf{y}})$, then we have an optimal solution, so STOP.

Step 5: Otherwise, $\text{DSP}(\bar{\mathbf{y}}) < z$ and we have not yet converged to an optimal solution, so return to Step 2 with the new $\bar{\mathbf{y}}$.

Since there are a finite (but exponential) number of extremal points $\mathbf{u}$, the Benders decomposition is guaranteed to converge in a finite number of steps. However, in the worst case convergence still takes exponential time, if we are forced to solve the subproblem for every possible $\bar{\mathbf{y}}$ in order to sufficiently constrain MP2. Kameshwaran [2004] suggests several ways to accelerate convergence, including starting with a dual cut generated from a *good* feasible solution, rather than an empty E. Another option is to attempt to judiciously select the best dual cuts in each step, which is treated by Magnanti and Wong [1979]. A third option is to restrict the space of possible $\bar{\mathbf{y}}$ using some prior information, which we will consider later in the form of pricing only agent-declared "interesting" bundles. As convergence time is a potentially important issue in the application of Benders to the CAP, we will explore these options to address it.

Chapter 7

Mapping the Benders Decomposition to LP2

Whereas with the Dantzig-Wolfe decomposition we began on the simpler domain of the assignment problem, this domain does not provide an interesting problem for the Benders decomposition. (The reason, which will become more clear later, is that the partition of goods into bundles is already fixed, which will be the work of the Benders master problem under the natural decomposition). Therefore, we will start immediately with LP representations of the full CAP. This section will focus on the results of using Benders decomposition on Bikhchandani and Ostroy's LP2 formulation [2001]. Bikhchandani et al. [2001] give LP2 as an intermediate formulation of the CAP, a stepping stone from the simple LP formulation to stronger formulations that guarantee an integral solution. Solving LP2 as a strict LP does not guarantee an integral solution; however, we can take advantage of the ability of the Benders decomposition to mesh continuous and discrete variables in a MIP to structure the problem such that we are guaranteed an integral solution.

The LP2 formulation of the CAP is as follows:

$$V(N) = \max \sum_{j \in N} \sum_{S \subseteq M} v_j(S) y(S,j)$$

$$\text{s.t.} \quad \sum_{S \subseteq M} y(S,j) \leq 1, \quad \forall j \in N \tag{7.1}$$

$$\sum_{j \in N} y(S,j) \leq \sum_{\sigma \ni S} z_\sigma, \quad \forall S \subseteq M \tag{7.2}$$

$$\sum_{\sigma \in \Pi} z_\sigma \leq 1 \tag{7.3}$$

$$y(S,j), z_\sigma \geq 0, \quad \forall S, \forall j, \forall \sigma$$

N is the set of agents, M is the set of goods, $v_j(S)$ is agent j's value for bundle S, and

$y(S, j)$ equals 1 or 0 depending on whether agent j is allocated bundle S. We also have a variable z_σ for each possible partition $\sigma \in \Pi$, where Π is the set of all possible partitions of the set of goods M across $n = |N|$ agents.

The intuition for the constraints is as follows. The first set, (7.1), ensures that no agent receives more than one bundle. The second set, (7.2), ensures that no agent gets a bundle unless that bundle appears in a selected partition σ - that is, that $y(S, j)$ can only be 1 if there is some agent partition σ such that $S^j \in \sigma$ and $z_\sigma = 1$. The third, (7.3), ensures that only one partition is active in the solution.

Note that the z_σ's do not appear in the objective function, but they do appear in the constraints and constrain the solution to a single partition. To use Benders, we will impose the restriction that $z_\sigma \in \{0, 1\}$ and allow the $y(S, j)$'s to be continuous.

7.1 Overview of the Benders Decomposition on LP2

The $y(S, j)$ are agent-bundle decision variables. Since fixing all $y(S, j)$ effectively chooses a partition σ and fixes all z_σ, the logical Benders breakdown of LP2 is to instead fix all z_σ (analogous to the $\mathbf{y}$ in Lasdon [2002] and Chapter 6). This means setting one z_σ to 1 and the rest to 0 - equivalent to choosing (temporarily fixing) a partition. This allows a degree of freedom in the $y(S, j)$ variables - an agent can choose between any of the bundles in the partition, and this choice will be made within the Benders subproblem. Note that once a partition of goods has been chosen, the allocation problem is equivalent to the assignment problem (since we now have n bundles that cannot be divided or combined to allot to n agents).

For clarity, we will make explicit the connection with the general Benders problem structure as outlined in Chapter 6. The $y(S, j)$ are the continuous variables analogous to $\mathbf{x}$ in the general example, and the z_σ are the integral variables analogous to $\mathbf{y}$. The values $v_j(S)$ are the objective coefficients of the continuous variables, analogous to c, and since the z_σ do not appear in the objective function, $f = \mathbf{0}$.

We then solve the dual with the z_σ fixed and use the solution to generate constraints in our modified master problem, MP2. We will represent the modified master problem in round t as MP^t. The process of solving the dual and adding constraints to the modified master problem is iterated until MP^t is sufficiently constrained to yield a feasible optimum. We'll know this when the objective values of the dual and of MP^t coincide.

Since LP2 as a linear program is not guaranteed to have integral solutions, it is important to show that making $z_\sigma \in \{0, 1\}$ constrains the solution of the entire problem to integral solutions.

Theorem 3. *Solving* LP2 *with the* $z_\sigma \in \{0, 1\}$ *guarantees an integral solution and thus a feasible allocation.*

Proof. Note that once the partition σ is fixed, the restricted problem $L(\sigma)$ is an assignment problem and thus has integral optimal solutions for any σ. When we enforce that $z_\sigma \in \{0,1\}$ and $\sum_{\sigma\in\Pi} z_\sigma \leq 1$, it is clear that no feasible solution can have more than one z_σ such that $z_\sigma > 0$. Therefore we can re-write the problem as

$$V(N) = \max_{\sigma\in\Pi} L(\sigma)$$

which clearly has an integral solution since $L(\sigma)$ is integral for all $\sigma \in \Pi$. $\square$

The remainder of this chapter will address the Benders subproblem (the dual $D(\sigma)$), the modified master problem MP^t, the decentralization of the subproblem and subsequent auction interpretation, and finally a few examples of the Benders auction in action on very small domains.

7.2 Subproblem: Primal and Dual of LP2 with z_σ fixed

Choosing a partition σ and fixing the $z_\sigma \in \{0,1\}$ accordingly, we obtain the following linear program $L(\sigma)$:

$$\begin{aligned} L(\sigma) = \max & \sum_j \sum_S v_j(S) y(S,j) \\ \text{s.t.} \quad & \sum_S y(S,j) \leq 1, \quad \forall j \in N \\ & \sum_j y(S,j) \leq 1, \quad \forall S \in \sigma \\ & \sum_j y(S,j) \leq 0, \quad \forall S \notin \sigma \\ & y(S,j) \geq 0, \quad \forall S, \forall j \end{aligned}$$

This program has the dual:

$$\begin{aligned} D(\sigma) = \min & \sum_{j\in N} \pi_j + \sum_{S\in\sigma} p(S) \\ \text{s.t.} \quad & p(S) + \pi_j \geq v_j(S), \quad \forall j \in N, \forall S \subseteq M \\ & \pi_j, p(S) \geq 0, \quad \forall j, \forall S \end{aligned}$$

The term $\sum_{S\in\sigma} p(S)$ is intuitively equivalent to π_s, the seller's surplus. Note also that we still consider the $p(S)$ variables for all bundles, not just bundles $S \in \sigma$.

Solving this dual will give us an objective value $D(\sigma)$, which provides a lower bound on the value of the final allocation. Also, importantly, it provides us with a set of prices that support the given partition σ and provide the necessary information to add a meaningful constraint to MP^t. This means that even though $p(S)$ for some $S \notin \sigma$ does not appear in the objective function of the dual, all $p(S)$ are important, as they contribute to a set of supporting prices.

The need to retain $p(S)$ for $S \notin \sigma$ makes the dual as written above different than the assignment problem, because there are bundles whose prices do not appear in the objective function which nevertheless need to be priced. However, if we do solve the dual as an assignment problem and obtain a solution $(\pi, \tilde{p})$ where $\tilde{p} = \{p(S) : S \in \sigma\}$, we can artificially construct from $\tilde{p}$ a set of prices p that support the partition σ over all bundles $S \subseteq M$, and then use (π, p) as a dual solution to generate an MP^t constraint within the Benders scheme.

7.3 The Modified Master Problem, MP^t

If we denote as E the set of solutions to the dual $D(\sigma)$ over all partitions σ, the complete Benders master problem is given by:

$$\begin{aligned}
\mathrm{MP}: \quad & \max \ \theta \\
& \text{s.t.} \quad \theta \leq \sum_j \pi_j^e + \sum_{\sigma \in \Pi} \left(z_\sigma * \sum_{S \in \sigma} p^e(S) \right), \quad \forall (\pi^e, p^e) \in E \\
& \sum_{\sigma \in \Pi} z_\sigma \leq 1 \\
& \qquad \theta \geq 0, \quad z_\sigma \in \{0,1\}, \quad \forall \sigma \in \Pi
\end{aligned}$$

Define E^t to be the set of solutions generated to the dual problem $D(\sigma)$ in rounds $\tau \in \{1, ..., t\}$. The modified master problem MP^t problem for round t is then given by:

$$\begin{aligned}
\mathrm{MP^t}: \quad & \max \ \theta^t \\
& \text{s.t.} \quad \theta^t \leq \sum_j \pi_j^e + \sum_{\sigma \in \Pi} \left(z_\sigma * \sum_{S \in \sigma} p^e(S) \right), \quad \forall (\pi^e, p^e) \in E^t \\
& \sum_{\sigma \in \Pi} z_\sigma \leq 1 \\
& \qquad \theta^t \geq 0, \quad z_\sigma \in \{0,1\}, \quad \forall \sigma \in \Pi
\end{aligned}$$

Having solved the dual $D(\sigma^t)$ for round t and received back a set of prices $(\hat{\pi}, \hat{p})$ that supports the given partition σ^t over all bundles $S \subseteq M$, we formulate and add a new constraint of the form

$$\sum_j \hat{\pi}_j + \sum_{\sigma \in \Pi} \left(z_\sigma * \sum_{S \in \sigma} \hat{p}(S) \right) \tag{7.4}$$

Solving this updated version will give us a new setting for the z_σ, which will dictate a new partition σ to fix, and we'll repeat the Benders process for round $t+1$ unless the solution is optimal.

Since MP^t is a MIP and the z_σ give a doubly exponential number of 0-1 variables, we might in practice represent the z_σ variables with decision variables $x(S)$, each corresponding to a bundle S. Instead of a single constraint on partitions, there would be a constraint for each good $i \in M$ that it not be contained in more than one bundle S. Thus MP^t would be given by:

$$\begin{aligned}
\text{MP}^t: \quad & \max \ \theta^t \\
\text{s.t.} \quad & \theta^t \le \sum_j \pi_j^e + \sum_S x(S) p^e(S), \quad \forall (\pi^e, p^e) \in E^t \\
& \sum_{S \ni i} x(S) \le 1, \quad \forall i \in M \\
& \theta^t \ge 0, \quad x(S) \in \{0,1\}, \quad \forall S
\end{aligned}$$

and we could determine from the decision variables $x(S)$ the new partition σ for the next round. However, in order to preserve clarity and best illustrate the connection with the Benders decomposition, we will continue with the first representation of MP^t for the remainder of this chapter, and leave the substitution of decision variables $x(S)$ as a suggestion for a practical implementation of this auction.

Here is the complete process of applying the Benders decomposition to LP2:

Step 1: $t = 1, E^0 = \varnothing, \theta^0 = \infty$. Choose an initial partition σ^1 at random.

Step 2: Call the dual problem $D(\sigma^t)$ and obtain a solution (π^t, p^t) with objective value $L(\sigma^t)$. If $L(\sigma^t) = \theta^{t-1}$ then we have an optimal solution and σ^t is an optimal partition, so STOP.

Step 3: Add the payoffs and prices (π^t, p^t) to E^{t-1} to create E^t.

Step 4: Call MP^t, which gives a new solution θ^t and a new partition σ^{t+1}.

$t \leftarrow t + 1$ and return to Step 2.

We can see that this is a valid decomposition by noticing that it has the following two important properties. First, if solving MP^t gives a solution z_σ that has already been selected, the dual $D(\sigma)$ has already been solved and a corresponding constraint added to MP^t with the same value for this particular σ. This means that we cannot select the same partition more than once (unless it turns out to be optimal), ensuring that progress will be made at each step. Second, the constraint added to MP^t for a cut (π^e, p^e) does not overconstrain the solution value for any other partition σ' - in other words, it is a valid inequality. Note that (π^e, p^e) is a feasible solution for $D(\sigma')$, and that the objective function for $D(\sigma')$ is comprised of $\sum_j \pi_j^e$ and $\sum_{S\in\sigma'} p^e(S)$. This second property ensures that the master problem will not be constrained below the optimal value and thus that the decomposition algorithm is optimal.

7.3.1 Exploitable Structure of MP^t

Having formulated the master problem and examined its behavior in several examples, it becomes apparent that the structure of the problem is perhaps simpler than it initially appears. It seems that after each round, the modified master problem MP^t could incorporate the new dual cut into existing constraints, rather than generating new ones. This could be to our advantage if we are somehow able to exploit this special structure.

Definition 7.3.1. Define $V(\sigma)$ to be the true total value of the optimal allocation under partition σ.

Proposition. *The complete master problem as formulated at the beginning of Section 7.3 can be equivalently written as the "re-written master problem" RMP:*

$$\begin{aligned} \text{RMP}: \quad & \max \ \theta \\ \text{s.t.} \quad \theta \le & \sum_{\sigma\in\Pi} (z_\sigma * V(\sigma)) \\ & \sum_{\sigma\in\Pi} z_\sigma \le 1 \\ & \theta \ge 0, \quad z_\sigma \in \{0,1\}, \quad \forall \sigma \in \Pi \end{aligned}$$

The intuition behind this idea is that only one of the z_σ variables can be 1 (the rest must be 0), so the value of θ is effectively constrained by the selection of σ and bound from above by *all* of the constraints on that σ. In an optimal solution, only the constraint with the minimal value on σ will be binding (this will be the true value, the rest being high estimates from various dual solutions). Therefore, once there is a new lower bound on the value of σ in another constraint, all other constraints are effectively obsolete *with regard to that* σ. This leads to the idea of combining all non-obsolete bounds into a single constraint, rather than keeping them around at some cost and no benefit.

We can extend this proposition to the modified master problem:

Definition 7.3.2. Define $V(\sigma, \tau)$ to be the total value of partition σ under the payoffs and prices (π^τ, p^τ) from Round τ:

$$V(\sigma, \tau) = \sum_j \pi_j^\tau + \sum_{S \in \sigma} p^\tau(S)$$

With $V(\sigma, \tau)$ thus defined, after each round, we can re-write the modified master problem as:

$$\begin{aligned}
\mathrm{MP^t}: \quad & \max \ \theta^t \\
& \text{s.t.} \quad \theta^t \leq \sum_{\sigma \in \Pi} \left(z_\sigma * V(\sigma, \tau)\right), \quad \forall \tau \in \{1, ..., t\} && (7.5) \\
& \sum_{\sigma \in \Pi} z_\sigma \leq 1 && (7.6) \\
& \theta^t \geq 0, \quad z_\sigma \in \{0, 1\}, \quad \forall \sigma \in \Pi
\end{aligned}$$

Due to constraint (7.6) in a solution at most one of the z_σ will be 1, and the rest will be 0. Call the chosen partition, whichever partition it turns out to be, σ'. Since $z_\sigma = 0$ for all $\sigma \neq \sigma'$, and due to constraints (7.5), the value of θ^t is bound to be $\leq V(\sigma', \tau)$ for all $\tau \in \{1, ..., t\}$. Of all the constraints (7.5), only the one with the minimal $V(\sigma', \tau)$ will be active at the optimal solution, and the rest are irrelevant. (It is possible for more than one to be minimal, in which case some are redundant). Thus we could re-write the modified master problem again with only one constraint (aside from $\sum_{\sigma \in \Pi} z_\sigma \leq 1$) as below. Call it the RMP^t.

$$\begin{aligned}
\mathrm{RMP}^t: \quad & \max \ \theta^t \\
& \text{s.t.} \quad \theta^t \leq \sum_{\sigma \in \Pi} \left(z_\sigma * \left(\min_{\tau \in \{1, ..., t\}} V(\sigma, \tau) \right) \right) \\
& \sum_{\sigma \in \Pi} z_\sigma \leq 1 \\
& \theta^t \geq 0, \quad z_\sigma \in \{0, 1\}, \quad \forall \sigma \in \Pi
\end{aligned}$$

In other words, each round, when a constraint of the form of (7.4) is added to MP^t from (π^t, p^t), the information contained in that constraint is the value of each partition σ under that set of payoffs and bundle prices, which is an *upper bound* on the true value $V(\sigma)$ of that partition. The master problem is a collection of upper bounds on the values of all partitions, and the solution to the master problem is the partition σ with the highest upper bound. Therefore, if for some partition σ the pre-existing upper bound was $V(\sigma, \tau')$ and the Round t dual solution (π^t, p^t) gives $V(\sigma, t) < V(\sigma, \tau')$, then this new, better upper bound will supersede the old from Round τ' and the old bound is obsolete and no longer contributes any information relevant to the solution of MP^t. The master problem is thus a minimax problem over the $V(\sigma, \tau)$:

RMP^t: Maximize over all $\sigma \in \Pi$ the value of $\min_{\tau \in \{1,\ldots,t\}} V(\sigma, \tau)$.

This lends itself to an updating strategy where, instead of maintaining all of the $V(\sigma, \tau)$ we only maintain the current minimum for each σ, and update that minimum at each round if necessary.

From this view of the master problem, we can see that for a non-optimal solution corresponding to a partition σ to be ruled out (without its being fixed as a partition and its dual solved), there must be a constraint in MP - the minimal value of $V(\sigma, \tau)$ - which binds its true value below the current best-known solution. This gives intuition to the impact of our selection of dual cuts (prices and payoffs) on the convergence of the algorithm. Even though in practice we might use decision variables $x(S)$ instead of z_σ, in which case the updating strategy above might not make sense, viewing the master problem in this way has yielded a valuable insight into its structure.

7.4 A Simple Example of Benders on LP2

To help illustrate the process of solving LP2 using the Benders decomposition, here is a very simple example. In this instance of the problem there are two agents, Agent 1 and Agent 2, and two goods, Good A and Good B. Their valuations are as follows:

S	$\varnothing$	$\{A\}$	$\{B\}$	$\{A, B\}$
$v_1(S)$	0	10	9	19
$v_2(S)$	0	5	6	20

So, Agent 1 values each separate item more than Agent 2 does, but Agent 2 values the two together more than Agent one does.

There are two possible partitions - call them $\sigma^X = \{\{A\}, \{B\}\}$ and $\sigma^Y = \{\{A, B\}, \varnothing\}$. We can tell by inspection that the optimal solution is with the partition $\sigma^Y = \{\{A, B\}, \varnothing\}$, Agent 1 receiving $\{\varnothing\}$ and Agent 2 receiving $\{A, B\}$.

To begin Benders, we fix a partition for round one - let's choose $\sigma^1 = \sigma^X = \{\{A\}, \{B\}\}$.

Round 1, Dual: It can be seen by inspection that the optimal solution under partition σ^1 is for Agent 1 to receive A and Agent 2 to receive B, with a total value of $10 + 6 = 16$. The restricted dual is

$$\begin{aligned} D(\sigma^1) = \min\,&(\pi_1 + \pi_2 + p(\{B\}) + p(\{A\})) \\ \text{s.t.} \quad & \pi_1 + p(S) \geq v_1(S), \forall S \subseteq M \\ & \pi_2 + p(S) \geq v_2(S), \forall S \subseteq M \\ & \pi_j, p(S) \geq 0, \quad j = 1, 2, \forall S \subseteq M \end{aligned}$$

Notice that even though the $p(S)$ terms associated with our chosen partition(σ^1) are the only $p(S)$ terms to appear with non-zero coefficients in the objective function, all of them appear in the constraints and will have non-trivial values in a solution. This is how this problem differs from a simple assignment problem.

Solving $D(\sigma^1)$ directly, one possible solution is:

$\pi_1 = 5$, $\pi_2 = 1$
$p(\varnothing) = 0$, $p(\{A\}) = 5$, $p(\{B\}) = 5$, $p(\{A, B\}) = 19$

with an objective value of $\pi_1 + \pi_2 + p(\{A\}) + p(\{B\}) = 5 + 1 + 5 + 5 = 16$.

We now use the $p(S)$ and π_j to add a constraint to the modified master problem, MP^1.

Round 1, Modified Master: To the modified master problem, which began as simply maximizing a variable θ subject to the condition that no more than one partition is active, we add a constraint of the following form:

$$\theta \leq \sum_{\sigma \in \Pi} z_\sigma * \left(\sum_{S \in \sigma} p(S) \right) + \sum_{j \in N} \pi_j$$

With our solution from the dual, we get:

$$\theta^1 \leq z_{\sigma^X} * (5 + 5) + z_{\sigma^Y} * (0 + 19) + 5 + 1$$

Solving MP^1 gives us the values $\theta^1 = 25$, $z_{\sigma^X} = 0$, $z_{\sigma^Y} = 1$. We compare $\theta^1 = 25$ to the dual solution 16 and conclude that we need iterate another round, with the partition fixed as $\sigma^2 = \sigma^Y$.

Note that, if we had not solved for $p(\{A,B\})$ in the dual and instead plugged in $p(\{A,B\}) = 0$, the solution would have been $\theta^1 = 16$, matching the dual and terminating the process. This shows how all prices are important, not just those for $S \in \sigma$.

Round 2, Dual: Now the restricted dual is

$$\begin{aligned} D(\sigma^2) &= \min\left(\pi_1 + \pi_2 + p(\{A,B\}) + p(\varnothing)\right) \\ \text{s.t.} \quad & \pi_1 + p(S) \geq v_1(S), \forall S \subseteq M \\ & \pi_2 + p(S) \geq v_2(S), \forall S \subseteq M \\ & \pi_j, p(S) \geq 0, \quad j = 1,2, \forall S \subseteq M \end{aligned}$$

One possible solution is:

$\pi_1 = 0$, $\pi_2 = 0$
$p(\varnothing) = 0$, $p(\{A\}) = 10$, $p(\{B\}) = 9$, $p(\{A,B\}) = 20$

with an objective value of 20.

Round 2, Modified Master: With our solution from the dual, we add to MP^2 the constraint:

$$\theta^2 \leq z_{\sigma^1} * (10 + 9) + z_{\sigma^2} * (0 + 20) + 0 + 0$$

Solving this gives us the values $\theta^2 = 20$, $z_{\sigma^X} = 0$, $z_{\sigma^Y} = 1$. Since this matches our dual value, we know it's the best solution.

Notice that our additional constraint that we added in Round 2 is in fact only a more binding version of the constraint we added in Round 1. This reflects the possible "one-constraint" structure of the modified master problem as discussed in Section 7.3.1, where that the complete version MP can be written with only one additional constraint, namely

$$\theta \leq \sum_{\sigma \in \Pi} \left(z_\sigma * V(\sigma)\right)$$

where $V(\sigma)$ is the maximal value of the allocation under partition σ.

7.5 Decentralizing and Solving the Benders Subproblem

Having seen an example, we now come to the most challenging and interest part of the implementation of a Benders auction - decentralization. Applying Benders decomposition to LP2 is an iterative process, and each round has two parts - fixing a partition σ^t and solving the dual $D(\sigma^t)$, and solving the updated modified master problem MP^t using the updated set of (π^e, p^e) solutions. From the auction perspective, it is naturally the central agent (the auctioneer or seller) who will solve MP^t and the bidding agents who will contribute to solving $D(\sigma^t)$. As in Dantzig-Wolfe, solving the subproblem by collaboration between the buyer agents and the central agent provides the meat of the auction interpretation of the entire process. The subproblem can be broken down and addressed in a number of ways, which in turn lend themselves to different auction interpretations.

A high-level auction interpretation of the dual is that the central agent announces a partition σ, and requires in order to proceed a set of prices and payoffs that supports that partition over *all* possible partitions. That means that these prices and payoffs must satisfy two properties:

> **Property 1:** The prices $p(S)$ are explicitly defined for all bundles $S \subseteq M$, and these prices and the agent payoffs support an efficient allocation of the partition σ to the bidding agents (note that this is the assignment problem).
>
> **Property 2:** The prices on bundles $S' \notin \sigma$ are *high enough* that no buyer agent could increase its surplus by switching the bundle it is assigned from the partition σ to another bundle not in σ.

At each round, for a fixed partition σ^t, the Benders subproblem is:

$$
\begin{aligned}
D(\sigma^t) = \min \sum_{j \in N} \pi_j + \sum_{S \in \sigma} p(S) \\
\text{s.t.} \quad p(S) + \pi_j \geq v_j(S), \quad \forall j \in N, \forall S \subseteq M \\
\pi_j, p(S) \geq 0, \quad \forall j, \forall S
\end{aligned}
$$

The term $\sum_{S \in \sigma} p(S)$ is intuitively equivalent to π_s, the seller's surplus. Note that there are constraints on the $p(S)$ variables for all bundles, not just bundles $S \in \sigma$, although only those prices for $S \in \sigma$ appear in the objective function. We can see from this that in an optimal solution, $p(S')$ for $S' \notin \sigma$ need only be high enough that the corresponding constraints $p(S') + \pi_j \geq v_j(S'), \forall j$ are not violated. For instance, if we had an optimal assignment for π_j and $\tilde{p}(S), S \in \sigma$, we could then set $p(S') = \infty, \forall S' \notin \sigma$, and the entire set of prices would support an efficient allocation over the partition σ because clearly all constraints are satisfied. Intuitively, no agent would switch to a bundle not in the partition because the cost would clearly be too high.

This points to the idea that we could construct a solution to the subproblem without solving the entire problem explicitly.

Although we require a dual solution (π^t, p^t) to support the partition σ^t over all bundles $S \subseteq M$, solving $D(\sigma^t)$ would be much simpler if we didn't need to worry about calculating $p(S)$ for $S \notin \sigma^t$. This is attractive from a computational standpoint because the total number of bundles S is exponential, but in a problem with n agents only n bundle prices appear in the objective function of the subproblem, along with n π_j variables. As mentioned before, the problem of determining an efficient allocation (with supporting prices) of a fixed partition σ is a case of the assignment problem, which is quite well-studied and easily tractable. Therefore, a viable alternate approach to solving the complete subproblem (which involves an exponential number of variables) would be to solve the underlying assignment problem and then construct the remainder of a comprehensive set of prices that would support that assignment.

However, while there exist a multiplicity of solutions to the subproblem, it is clear that with the Benders decomposition some dual cuts are superior to others from the point of view of constraining the master problem and thus speeding up convergence. Magnanti and Wong [1979] address this issue for Benders decomposition generally, and Kameshwaran [2004] adapts some of their proposals specifically for Benders decomposition on the knapsack problem. In the auction setting, there will exist a tension between quality of the dual cuts we generate and the amount of computational effort we're willing to expend to ensure that certain level of quality.

Another idea to address the complexity issue is to have agents report at the beginning of the auction the set of bundles that they are "interested" in. For example, it may be that for agent j, the set of bundles for which $v_j(S') < v_j(S)$ for all $S' \subset S$ is small, and the agent is thus only interested in those bundles which are not supersets of equally valuable bundles. Eliminating the "uninteresting" bundles at the beginning could result in much greater computational efficiency later, depending on the valuation domain of the agents. This technique would clearly be more effective in problems where the agents had sparse valuation functions and were interested in only a small fraction of the available bundles. A simple example where this would *not* make a difference is an agent with a non-zero value for each good and a linear additive valuation function - for this agent every bundle is "interesting" by the definition given above. However, it is also easy to imagine examples of more complex, combinatorial, and sparse valuation functions which would lend themselves quite naturally to the "interesting bundle" approach.

This section will first discuss the impact of selection of dual cuts on the convergence of the Benders algorithm as a whole with reference to some prior papers that treat the subject [Magnanti and Wong, 1979, Kameshwaran, 2004]. It will then explore both the direct approach to solving the subproblem $D(\sigma)$ (which ultimately yields little of interest) and the assignment problem approach, which itself has several different potential avenues to pursue and proves to be the more interesting of the two. We will continually keep in mind the issue of convergence and the effect that our various approaches might have on the number of iterations of the Benders process required to come to an optimal overall allocation.

7.5.1 Selecting Dual Cuts to Optimize Benders

Since its creation, the Benders decomposition method has been used on a very wide range of problems, with varying degrees of success. Magnanti and Wong [1979] examine the Benders decomposition in general and propose several ideas to minimize the number of iterations required for convergence, including making a good selection of initial cuts and selecting good cuts to add to the master problem at every step. They introduce the notions of *dominance* and *Pareto optimality* to the consideration of dual cuts in the Benders subproblem, which turn out to be effective ways of defining the "best" cuts to add to the master problem. Kameshwaran [2004] utilizes these definitions specifically for Benders on the knapsack problem.

Definition 7.5.1. Define $V(\hat{\pi}, \hat{p}, \sigma)$ to be the total value of any partition σ under the dual cut $(\hat{\pi}, \hat{p})$, so

$$V(\hat{\pi}, \hat{p}, \sigma) = \sum_j \hat{\pi}_j + \sum_{S \in \sigma} \hat{p}(S)$$

Definition 7.5.2. We say that the cut $(\hat{\pi}, \hat{p})$ *dominates* the cut $(\bar{\pi}, \bar{p})$ if $V(\hat{\pi}, \hat{p}, \sigma) \leq V(\bar{\pi}, \bar{p}, \sigma)$ for all $\sigma \in \Pi$, with strict inequality holding for at least one σ.

Definition 7.5.3. We say that a cut $(\hat{\pi}, \hat{p})$ is *Pareto optimal* if it is not dominated by any other cut.

It is difficult to generalize across a wide range of problem structures, but it seems clear intuitively that we would always prefer to add a Pareto optimal constraint to the modified master problem MP^t rather than a dominated constraint, if possible, for the sake of faster convergence. Unfortunately, even given the transparent structure of MP^t in this case, the wide range of possible agent valuation functions makes it difficult to make strong claims about which dual cuts are the "best."

Searching for Pareto Optimal Cuts

Let us examine for a moment only the payoffs and prices associated with bundles $S \in \sigma$. Since there is a clear trade-off in the dual constraints between price and agent payoff,

$$p(S) + \pi_j \geq v_j(S), \quad \forall j, S$$

a natural question to ask is whether one is preferred to the other from the standpoint of convergence. In particular, is it possible to make general statements about whether buyer-optimal prices or seller-optimal prices have the best convergence properties?

Looking again at the form the modified master problem constraints take,

$$\theta \leq \sum_j \pi_j^e + \sum_{\sigma \in \Pi} \left(z_\sigma * \sum_{S \in \sigma} p^e(S) \right), \quad \forall (\pi^e, p^e) \in E$$

it would seem that in many cases, since prices are bundle-specific and agent payoffs get added to all partitions, that we would prefer higher prices to higher payoffs because the latter would inflate the partition values for all partitions, rather than just those including certain bundles.

It is tempting to claim that this is always the case. We could define a general seller-optimal solution $(\hat{\pi}, \hat{p})$ as follows. Solve the assignment problem under partition σ for the optimal assignment and determine the seller-optimal payoffs and prices $\hat{\pi}_j$ and $\tilde{p}(S)$ - that is, the prices for $S \in \sigma$ that minimize $\sum_j \hat{\pi}_j$, the payoff to the bidding agents. Then, for all bundles $S' \notin \sigma$, set the prices $\hat{p}(S') = \max_j(v_j(S') - \hat{\pi}_j)$. This will choose the minimum possible prices while still satisfying the constraints that $\pi_j + p(S) \geq v_j(S)$.

Fitting this into the framework of Magnanti and Wong [1979], we would like to assert that the dual cut $(\hat{\pi}, \hat{p})$ is always Pareto optimal. A stronger assertion still would be that $(\hat{\pi}, \hat{p})$ is the *best* possible cut - that no other cut provides a *tighter* upper bound on *any* partition σ'. Analytically, there is no other feasible dual cut $(\bar{\pi}, \bar{p})$ in that round for which the strict inequality $V(\hat{\pi}, \hat{p}, \sigma') > V(\bar{\pi}, \bar{p}, \sigma')$ holds for any partition σ'.

Unfortunately, while this holds in many plausible examples, it is not universally true. Consider the following counterexample.

Example. *There are two agents, Agent 1 and Agent 2, and two goods, Good A and Good B. Their valuations are as follows:*

S	$\varnothing$	$\{A\}$	$\{B\}$	$\{A, B\}$
$v_1(S)$	*0*	*4*	*3*	*8*
$v_2(S)$	*0*	*6*	*5*	*10*

Again there are two possible partitions - call them $\sigma^X = \{\{A\}, \{B\}\}$ *and* $\sigma^Y = \{\{A, B\}, \varnothing\}$. *Fix partition* $\sigma^1 = \sigma^Y = \{\{A, B\}, \varnothing\}$ *for Round 1. Consider the following two dual solutions, both of which are valid and the first of which has been formulated to be the seller-optimal solution as described above:*

	π_1	π_2	$\varnothing$	$\{A\}$	$\{B\}$	$\{A, B\}$
$(\hat{\pi}, \hat{p})$	*0*	*0*	*0*	*6*	*5*	*10*
$(\bar{\pi}, \bar{p})$	*0*	*2*	*0*	*4*	*3*	*8*

It can be seen that, counter to what we would hope, $(\bar{\pi}, \bar{p})$ *dominates* $(\hat{\pi}, \hat{p})$*, because*

$$V(\hat{\pi}, \hat{p}, \sigma^Y) = 10 = V(\bar{\pi}, \bar{p}, \sigma^Y)$$

and

$$V(\hat{\pi}, \hat{p}, \sigma^Y) = 11 > 9 = V(\bar{\pi}, \bar{p}, \sigma^Y)$$

If we start with the dual solution $(\hat{\pi}, \hat{p})$, in general, it will be impossible to decrease the price on a bundle $S' \notin \sigma$ without increasing the corresponding agent payoff π_j for the agent j with the maximal value for that bundle. However, this counterexample has been created by "overloading" Agent 2, who has the highest values for both Good A and Good B, and therefore increasing Agent 2's payoff π_2 by 1 allows us to decrease *both* $p(A)$ and $p(B)$ by 1 and attain a lower total value, even though it would be impossible for Agent 2 to receive both A and B as separate bundles.

Given the somewhat contrived nature of this counterexample, it seems plausible that if we impose certain "niceness" restrictions on the problem - for example, perhaps no agent's valuation function can be allowed to complete dominate another's - we may be able to make more forceful claims about the strength of the seller-optimal cut $(\hat{\pi}, \hat{p})$. I leave this as a proposition for future exploration.

Proposition. *Under certain well-constructed assumptions about agent valuations, the seller-optimal dual cut* $(\hat{\pi}, \hat{p})$ *is guaranteed to be Pareto optimal.*

Ultimately, it seems from this analysis that it will be difficult to make sweeping domain-independent claims about Pareto optimality and buyer-optimal versus seller-optimal dual solutions.

Negative Claims about Pareto Optimality

As it turns out, we can make the negative assertion that any cut from a large, easily-defined class of cuts is *not* Pareto optimal.

Theorem 4. *Let* $(\bar{\pi}, \bar{p})$ *be a cut for which* $\bar{p}(S') > \max_j(v_j(S') - \bar{\pi}_j)$ *for some bundle* $S' \notin \sigma$*. This cut is* not *Pareto optimal because it is dominated by the cut* $(\hat{\pi}, \hat{p})$*, if we define* $(\hat{\pi}, \hat{p})$ *to be identical to* $(\bar{\pi}, \bar{p})$ except *that* $\hat{p}(S') = \max_j(v_j(S') - \hat{\pi}_j)$*.*

Proof. Since $\bar{p}(S) = \hat{p}(S)$ for any $S \neq S'$, we know that for any partition σ such that $S' \notin \sigma$, $V(\bar{\pi}, \bar{p}, \sigma) = V(\hat{\pi}, \hat{p}, \sigma)$. Now consider any partition $\sigma' \ni S'$. Since $(\bar{\pi}, \bar{p})$ and $(\hat{\pi}, \hat{p})$ are identical except with respect to S',

$$V(\bar{\pi}, \bar{p}, \sigma') - \bar{p}(S') = V(\hat{\pi}, \hat{p}, \sigma') - \hat{p}(S')$$

Therefore, since $\bar{p}(S') > \hat{p}(S')$, $V(\bar{\pi}, \bar{p}, \sigma') > V(\hat{\pi}, \hat{p}, \sigma')$. Thus, $V(\bar{\pi}, \bar{p}, \sigma) \geq V(\hat{\pi}, \hat{p}, \sigma)$ for all σ, with strict inequality for those $\sigma' \ni S'$. Therefore $(\hat{\pi}, \hat{p})$ dominates $(\bar{\pi}, \bar{p})$. □

This is useful theoretically because it makes clear the optimal value of the $p(S')$ for $S' \notin \sigma$ once the agents payoffs π_j are set. However, in practice, it will be hard to determine every price optimally in a computationally satisfactory manner, so it is likely that most cuts in practice will come from this class that is verifiably not Pareto optimal.

We now have a better understanding of what constitutes a good cut; unfortunately, determining the best cut for a round is tantamount to conducting an auction over the entire bundle space, since we must determine an exact price for every single bundle, whether or not it is contained in σ. A more pragmatic approach would be to rely on effective and much faster heuristics that generate a dual cut approaching the optimal cut. It is clear that the lower the partition values a cut gives, the better it is (it provides a tighter upper bound). This means that we will aim for low prices that still support the fixed partition σ. The desired tradeoff between agent payoffs π_j and the prices $\tilde{p}(S)$ on bundles $S \in \sigma$ is even less clearcut once we have surrendered the idea of obtaining a Pareto optimal cut.

Having determined the cuts we should aim for, we now must address the inevitable tension between the quality of the cuts we use and the efficiency we're willing to sacrifice to guarantee that quality.

7.5.2 Decentralized Solution of the Complete Subproblem

The most straightforward approach to the subproblem is to distribute the computation to agents by the appropriate variables, and aggregate the results in the center. The problem $D(\sigma)$ breaks down in the following way, for a setting with n agents:

$$\min \sum_j \pi_j + \sum_{S \in \sigma} p(S)$$

$$\text{s.t.}$$

$$\begin{bmatrix} \pi_1 & + & & \dots & & + & p(S) & \geq & v_1(S) & \forall S \subseteq M \\ & & \pi_2 & + & \dots & + & p(S) & \geq & v_2(S) & \forall S \subseteq M \\ & & & \ddots & & & & & & \\ & & & & \pi_n & + & p(S) & \geq & v_n(S) & \forall S \subseteq M \\ & & & & & \pi_j, & p(S) & \geq & 0 & \forall j \in N, S \subseteq M \end{bmatrix}$$

To solve this in a distributed way, each agent j would solve the problem:

$$
\begin{aligned}
D_j(\sigma^t) = \min\ & \pi_j + \sum_{S\in\sigma} p_j(S) \\
\text{s.t.}\quad & p_j(S) + \pi_j \geq v_j(S), \quad \forall S \subseteq M \\
& \pi_j, p_j(S) \geq 0, \quad \forall S
\end{aligned}
$$

and submit a solution (π_j, p_j) to the center, which would then construct an overall solution to $D(\sigma^t)$ using the π_j's and according to the rule $p(S) = \max_j p_j(S)$ (this will guarantee that none of the pricing constraints are violated and thus the overall solution is valid).

It would be possible to obtain a similar result by conducting an actual auction over all bundles - but, as our goal is to design such an auction, it hardly makes sense to include one at every single step! An important question is whether there exists a more effective method to obtain this comprehensive set of prices. For example, if the number of "interesting" bundles were small, agents could only submit prices on bundles they found interesting.

If we wish to decentralize further, the determination of the final $p(S)$ from the $p_j(S)$ could presumably done in a distributed fashion as well, although there could be some complications if privacy and information revelation to other agents were an issue.

If we look at this approach critically, we notice that each agent submitting prices on all bundles (or all interesting bundles) has the same complexity as agent's entire valuation function. This means that we could conduct, in similar time, the entire simplistic auction wherein each agent submits its complete valuation function to the center and the center computes the optimal allocation in one shot. This begs the question of whether we are really gaining anything from decomposing the original problem in this fashion. It seems that using Benders decomposition on LP2 and solving the dual in this manner does not provide sufficient advantages over the simplistic one-shot complete-revelation auction described above to justify the computational cost. We are therefore motivated to search for alternate ways to solve the Benders subproblem. A promising idea to pursue is the structure of the allocation problem once a partition has been fixed, which is identical to that of the assignment problem, which is well-explored.

7.5.3 Decentralizing and Solving Using the Assignment Problem Reduction

The overview of this approach should be familiar by now. The first step is to solve the allocation problem under the fixed partition σ, obtaining efficient prices $\tilde{p}(S)$ on all $S \in \sigma$. This step will also yield agent payoffs π_j which are an important component of the subproblem solution. The second step is to use this and other information to construct a set of prices $p(S')$ on the bundles $S' \notin \sigma$. Since these prices don't affect the objective value of the subproblem, we only require that they not violate any of the constraints, i.e. none of them are so low that an agent would prefer a bundle S' to one in the partition σ. Finally we'll combine all of these prices and payoffs into a single solution, (π, p) that is an optimal solution for $D(\sigma)$.

There are a number of ways to attack the assignment problem, each with slightly different properties. Demange et al. [1986] describe an ascending-price auction which could be used with prices on bundles $S \in \sigma$. Their auction will yield the buyer-optimal (i.e. payoff maximizing) prices. Bikhchandani et al. [2001] outline some other methods that attain similar results. From the previous discussion of Pareto optimal cuts comes interest in an algorithm for the assignment problem that yields the seller-optimal prices (those that minimize the agent payoffs). This outcome can be obtained as a result of more recent work on descending-price auctions by Garg and Mishra [2004]. However, once we retreat from demanding Pareto optimality of our dual cuts, it is less clear that the seller-optimal prices are the ones we desire. Our pricing needs will likely vary depending on the nature of heuristics that we choose to employ and thus, our choice of method to solve the assignment problem will be influenced by the following discussion which further explores the problem of best extrapolating a larger set of supporting prices p from prices $\tilde{p}$ that support the efficient assignment of a fixed partition.

Assuming the assignment problem has been solved, we want to take the solution $(\pi, \tilde{p})$ (with prices $\tilde{p}(S)$ only for $S \in \sigma$) and construct a complete set of prices $p(S)$ for all $S \subseteq M$ such that (π, p) is a solution for $D(\sigma^t)$. One way to do so would be the following scheme:

Step 1: Let $p(M) = \tilde{p}(M)$ if $M \in \sigma$, and otherwise let $p(M) = V - \min_{j \in N} \pi_j$, where V is an upper bound on the values of all agents. One way to compute V would be to solicit before the first round from each agent j its valuation $v_j(M)$ of the set of all goods. Then $V = \max_j v_j(M)$ is a valid upper bound. Also let $p(\varnothing) = 0$.

Step 2: For all $S' \notin \sigma$, set $p(S')$ to

$$p(S') = \min_{S \in \sigma, S \supseteq S'} \tilde{p}(S).$$

The set of prices p thus constructed from $\tilde{p}$ create a complete and valid solution (π, p) to $D(\sigma^t)$. This can be seen from two properties of the $p(S)$. First, no bidder would have a greater payoff for a bundle $S' \subseteq S$ at the same price as the bundle $S \in \sigma$, which ensures that no agent can increase their profit π_j. Second, the objective value of $D(\sigma^t)$ is unaltered, because the only prices that appear in it are for bundles $S \in \sigma$.

In practice, the issue that we must confront is that while all sets of prices $p(S'), S' \notin \sigma$ that support the allocation under the fixed partition can be part of a complete solution (π, p), the higher the prices p are above the true values of their bundles, the less useful information they contain to be incorporated into the modified master problem. As discussed previously in this section, for faster convergence we want tighter bounds and therefore less inflated prices. From the point of view of convergence, the "best" dual solution is the solution under a given partition with the lowest possible supporting prices. For example, if the dual solution had prices p such that $p(S) = \max_j v_j(S)$, the new master constraint would perfectly price each partition and the optimal partition would be determined in the subsequent solution to the modified master problem.

On the other hand, if we employ the extrapolation scheme described above, wherein each bundle $S \notin \sigma$ is given the same value as minimally-valued superset in σ (or M, the set of all goods), our dual cut is likely to be weak. It is almost guaranteed to be non-Pareto optimal by Theorem 4, and the bounds it provides leave much room for improvement by more sophisticated heuristics in many cases. In a later concrete example, we will see how this strategy leaves bundles not included in the fixed partition with excessively high prices, which impedes convergence. Of course, it is worth noting that every domain is different, there will always be exceptions to heuristic rules, and the empirical testing and verification of heuristics across a wide test suite of cases is highly desirable.

As we already discussed in Section 7.5.1, since there is a clear trade-off in the dual constraints between price and agent payoff:

$$p(S) + \pi_j \geq v_j(S), \quad \forall j, S$$

a natural question to ask is whether one is preferred to the other from the standpoint of convergence. In particular, is it possible to make general statements about whether buyer-optimal prices or seller-optimal prices have the best convergence properties?

Looking again at the form the modified master problem constraints take:

$$\theta \leq \sum_j \pi_j^e + \sum_{\sigma \in \Pi} \left(z_\sigma * \sum_{S \in \sigma} p^e(S) \right), \quad \forall (\pi^e, p^e) \in E$$

it would seem that in many cases, since prices are bundle-specific and agent payoffs get added to all partitions, that we would prefer higher prices to higher payoffs because the latter would inflate the partition values for all partitions, rather than just those including certain bundles.

Selectively Maintaining Pricing and Payoff Information

Jumping to another level of sophistication, another idea to maximize the use of information that has already been gathered would be for the center to maintain information on the reported prices for all bundles over all rounds, in particular the minimal price thus far on every bundle. Rather than immediately aggregating the information into full-partition valuations, we could maintain information on current minimal prices, their associated payoffs, and also maximal cumulative agent payoffs, and combine these to get a current "best" valuation for every partition. We can take advantage of the "one-constraint" property of the modified master problem (from Section 7.3.1) and combine all of the pricing and payoff information from all rounds into a single bound for each partition.

The advantage of this approach is that valuable bundle-specific information is not lost. Here is a pared down example to illustrate the potential benefits:

Example. *Imagine that there is a partition $\sigma = \{\alpha, \beta\}$ with two bundles, α and β. Say that two rounds have been conducted, both of which yielded solutions with $\pi_j = 0$ for all agents j. Say that price on bundle α in the Round 1 was lower than in Round 2, and vice versa for bundle β. So, $p^1(\alpha) < p^2(\alpha)$, and $p^2(\beta) < p^1(\beta)$. With the original setup, the modified master problem would bound the value of partition σ and indirectly the value of θ from above at:*

$$V(\sigma) \leq p^1(\alpha) + p^1(\beta)$$

and

$$V(\sigma) \leq p^2(\alpha) + p^2(\beta)$$

However, had we employed a scheme that updated prices, we might better be able to take advantage of the fact that Round 1 generated a lower supporting price for α and Round 2 generated a lower supporting price for β. Combining this information into one value, we can get a strictly better bound:

$$\begin{aligned} V(\sigma) serious &\leq p^1(\alpha) + p^2(\beta) \\ &< \min\{p^1(\alpha) + p^1(\beta), p^2(\alpha) + p^2(\beta)\} \end{aligned}$$

where the final strict inequality follows from $p^1(\alpha) < p^2(\alpha)$, and $p^2(\beta) < p^1(\beta)$.

Clearly, there are cases where valuable bundle-specific price information gets subsumed by the partition-valuing structure of the MP constraints and is thus lost. A price-updating scheme would hope to retain and take advantage of this information. It would also dovetail well with the conversion of the master problem from partition variables z_σ to bundle decision variables $x(S)$, as mentioned in Section 7.3.

Any scheme of maintaining prices is greatly complicated by the addition of agent payoffs into the mix. In general, it seems that lower agent payoffs are preferred, but as we saw in the example from Section 7.5.1, if one agent has the highest values on multiple mutually exclusive bundles, raising that agent's surplus and decreasing the prices on those bundles will lead to a better solution than the payoff-minimizing one. More analysis and empirical evidence are required to better understand these tradeoffs and develop appropriate heuristics for pricing non-included bundles.

Another idea to improve efficiency would be to have, after the first round, agents only submit bundle prices which have *changed* since the last round. If the prices for a majority of bundles remain static across rounds, it makes sense to cut that dead weight from the informational exchange between the bidding agents and the central agent. Again, this is an idea that merits empirical experimentation to determine its effectiveness across a variety of settings.

Returning to the broad view, since Benders and Dantzig-Wolfe are in a sense duals of each other, we might expect the opposite of the Dantzig-Wolfe auction, in which the center announces prices and the bidding agents respond with best-response bundles under those prices. Indeed, in the Benders auction as it has taken shape, it is the master problem (at the center) which finds an allocation across all feasible allocations, and the subproblem (the agents) who respond to that proposal with prices.

7.6 Second Example of Benders on LP2, Using the Assignment Problem Reduction

Now I'll step through an example borrowed from Bikhchandani and Ostroy [2001] to demonstrate the idea of solving a reduced $D(\sigma^t)$ as the assignment problem and using a simple technique to construct the remainder of a full dual solution. Additionally, Bikhchandani and Ostroy use this specific example to as a counterexample to show that as a pure LP, the LP2 formulation does not always yield integral results. In this example, though it is cast as a mixed integer program and results in a valid integral allocation, as predicted by Theorem 3.

In this example there are two agents, Agent 1 and Agent 2, and three goods, A, B, and C. The valuations are:

S	$\varnothing$	$\{A\}$	$\{B\}$	$\{C\}$	$\{A,B\}$	$\{A,C\}$	$\{B,C\}$	$\{A,B,C)\}$
$v_1(S)$	0	4	4	4.25	7.5	7	7	9
$v_2(S)$	0	4	4.25	4	7	7.5	7	9

There are four possible partitions in this example. Call them:

$$\sigma^W = \{\{A,B,C\},\varnothing\},\ \sigma^X = \{\{A,B\},\{C\}\}$$
$$\sigma^Y = \{\{A,C\},\{B\}\},\ \sigma^Z = \{\{B,C\},\{A\}\}$$

For Round 1, let's fix $\sigma^1 = \sigma^W = \{\{A, B, C\}, \{\varnothing\}\}$.

Round 1, Dual: Having fixed σ^1, solving the assignment problem under that partition would assign $\{A, B, C\}$ to one of the agents (say Agent 1), $\varnothing$ to the other agent (Agent 2), and give $D(\sigma^1) = 9$, $\pi_1 = \pi_2 = 0$, $\tilde{p}(\{A, B, C\}) = 9$, $\tilde{p}(\varnothing) = 0$.

Now, assume we have obtained the information that $v_1(\{A, B, C\}) = v_2(\{A, B, C\}) = 9$ (this is plausible and could be an argument for always starting with the partition that includes M as a single bundle). We can therefore use $V = 9$ as an upper bound on the value of any bundle.

Now using the technique described in Section 7.5.3, we can construct a complete set of prices p^1 over all of the bundles:

S	$\varnothing$	$\{A\}$	$\{B\}$	$\{C\}$	$\{A, B\}$	$\{A, C\}$	$\{B, C\}$	$\{A, B, C)\}$
$p(S)$	0	9	9	9	9	9	9	9

We'll use these to add a constraint to the modified master.

Round 1, Modified Master: With our dual solution, we'll add to create MP^1 the following constraint, given first in the general form and then in this instance:

$$\theta \leq \sum_{\sigma \in \Pi} z_\sigma * \left(\sum_{S \in \sigma} p(S) \right) + \sum_{j \in N} \pi_j$$

$$\theta^1 \leq z_{\sigma^W} * (9 + 0) + z_{\sigma^X} * (9 + 9) + z_{\sigma^Y} * (9 + 9) + z_{\sigma^Z} * (9 + 9) + 0 + 0$$

Solving MP^1 gives us the values $\theta^1 = 18$, $z_{\sigma^Z} = 1$, all other $z_\sigma = 0$ (ties broken arbitrarily). $\theta^1 > D(\sigma^1)$, so we iterate another round, with $\sigma^2 = \sigma^Z$.

Round 2, Dual: Solving the assignment problem under σ^Z, the center gives $\{A\}$ to Agent 1 and $\{B, C\}$ to Agent 2. $D(\sigma^2) = 11$, $\pi_1 = \pi_2 = 0$, $\tilde{p}(\{A\}) = 4$, $\tilde{p}(\{B, C\}) = 7$.

Reusing our bound of $V = 9$, we construct a complete set of prices p^2 over all of the bundles:

S	$\varnothing$	$\{A\}$	$\{B\}$	$\{C\}$	$\{A, B\}$	$\{A, C\}$	$\{B, C\}$	$\{A, B, C)\}$
$p(S)$	0	4	7	7	9	9	7	9

Round 2, Modified Master: With the new constraint from (π^2, p^2), MP^2 is:

$$\begin{aligned} \text{MP}^2: \quad & \max \ \theta^2 \\ \text{s.t.} \quad & \theta^2 \le 9\sigma^W + 18\sigma^X + 18\sigma^Y + 18\sigma^Z \\ & \theta^2 \le 9\sigma^W + 16\sigma^X + 16\sigma^Y + 11\sigma^Z \\ \sum_{\sigma \in \Pi} & z_\sigma \le 1 \\ & \theta^2 \ge 0, \quad z_\sigma \in \{0,1\}, \quad \forall \sigma \in \Pi \end{aligned}$$

with solution $\theta^2 = 16$, $z_{\sigma^X} = 1$, all other $z_\sigma = 0$. Again $\theta^2 > D(\sigma^2)$, so we iterate another round, with $\sigma^3 = \sigma^X$.

Round 3, Dual: Under σ^X, the center gives $\{A, B\}$ to Agent 1 and $\{C\}$ to Agent 2. $D(\sigma^3) = 11.5$, $\pi_1 = \pi_2 = 0$, $\tilde{p}(\{A, B\}) = 7.5$, $\tilde{p}(\{C\}) = 4$.

S	$\varnothing$	$\{A\}$	$\{B\}$	$\{C\}$	$\{A, B\}$	$\{A, C\}$	$\{B, C\}$	$\{A, B, C)\}$
$p(S)$	0	7.5	7.5	4	7.5	9	9	9

Round 3, Modified Master:

$$\begin{aligned} \text{MP}^3: \quad & \max \ \theta^3 \\ \text{s.t.} \quad & \theta^3 \le 9\sigma^W + 18\sigma^X + 18\sigma^Y + 18\sigma^Z \\ & \theta^3 \le 9\sigma^W + 16\sigma^X + 16\sigma^Y + 11\sigma^Z \\ & \theta^3 \le 9\sigma^W + 11.5\sigma^X + 16.5\sigma^Y + 16.5\sigma^Z \end{aligned}$$

with solution $\theta^3 = 16$, $z_{\sigma^Y} = 1$, all other $z_\sigma = 0$. $\theta^3 > D(\sigma^3)$, so we iterate another round with $\sigma^4 = \sigma^Y$.

Round 4, Dual: Under σ^Y, the center gives $\{B\}$ to Agent 1 and $\{A, C\}$ to Agent 2. $D(\sigma^4) = 11.5$, $\pi_1 = \pi_2 = 0$, $\tilde{p}(\{B\}) = 4$, $\tilde{p}(\{A, C\}) = 7.5$.

S	$\varnothing$	$\{A\}$	$\{B\}$	$\{C\}$	$\{A, B\}$	$\{A, C\}$	$\{B, C\}$	$\{A, B, C)\}$
$p(S)$	0	7.5	4	7.5	9	7.5	9	9

Round 4, Modified Master:

$$\begin{aligned} \text{MP}^4: \quad & \max \ \theta^3 \\ \text{s.t.} \quad & \theta^4 \le 9\sigma^W + 18\sigma^X + 18\sigma^Y + 18\sigma^Z \\ & \theta^4 \le 9\sigma^W + 16\sigma^X + 16\sigma^Y + 11\sigma^Z \\ & \theta^4 \le 9\sigma^W + 11.5\sigma^X + 16.5\sigma^Y + 16.5\sigma^Z \\ & \theta^4 \le 9\sigma^W + 16.5\sigma^X + 11.5\sigma^Y + 16.5\sigma^Z \end{aligned}$$

with solution $\theta^4 = 11.5$, which is equal to $D(\sigma^4)$, so the final solution is the assignment from the dual in Round 4.

7.7 Third Example of the Benders on LP2, Using "Best" Dual Cuts

In this example we'll try to use the "best" solution possible, i.e. the one that adds the most binding dual constraint to the modified master problem MP^t. As suggested in Section 7.5.1, this is often the seller-optimal one with respect to bundles in the partition σ (i.e. minimize $\sum_j \pi_j$) and the one with the minimal prices possible for $S \notin \sigma$.

We can implement this solution mathematically by first solving for $D(\sigma)$, then solving a modified version, the "second dual":

$$\min \sum_j \pi_j$$

$$\text{s.t.} \quad p(S) + \pi_j \geq v_j(S), \quad \forall j \in N, \forall S \subseteq M \tag{7.7}$$

$$\sum_j \pi_j + \sum_{S \in \sigma} p(S) = D(\sigma) \tag{7.8}$$

$$\pi_j, p(S) \geq 0, \quad \forall j, \forall S$$

Constraints (7.7) are identical to the original dual, constraint (7.8) guarantees that the solution will be an optimal dual solution, and the choice of objective guarantees that the optimal solution chosen will be that with the highest seller payoff.

Alternately it is plausible that some methods of solving the assignment problem will directly yield buyer-optimal prices.

In this example there are three agents, Agents 1, 2, and 3, and three goods, A, B, and C. The valuations are:

S	$\varnothing$	$\{A\}$	$\{B\}$	$\{C\}$	$\{A,B\}$	$\{A,C\}$	$\{B,C\}$	$\{A,B,C)\}$
$v_1(S)$	0	2	3	4	9	7	6	11
$v_2(S)$	0	4	5	6	7	9	7	10
$v_3(S)$	0	5	2	3	6	5	8	8

There are five possible partitions in this example. Call them:

$\sigma^V = \{\{A\},\{B\},\{C\}\}$, $\sigma^W = \{\{A,B,C\},\varnothing,\varnothing\}$, $\sigma^X = \{\{A,B\},\{C\},\varnothing\}$ $\sigma^Y = \{\{A,C\},\{B\},\varnothing\}$, $\sigma^Z = \{\{B,C\},\{A\},\varnothing\}$

For Round 1, let's fix $\sigma^1 = \sigma^W = \{\{A,B,C\},\varnothing,\varnothing\}$.

Round 1, Dual: Having fixed σ^1, solving the assignment problem under that partition would assign $\{A, B, C\}$ to Agent 1, $\varnothing$ to Agents 2 and 3, and give $D(\sigma^1) = 11$.

The "best" solution as we've defined it gives $\pi_1 = \pi_2 = \pi_3 = 0$ and:

S	$\varnothing$	$\{A\}$	$\{B\}$	$\{C\}$	$\{A, B\}$	$\{A, C\}$	$\{B, C\}$	$\{A, B, C)\}$
$p(S)$	0	5	5	6	9	9	8	11

Remark. *As a quick aside, consider briefly the solution if it had been Agent 2 instead of Agent 1 with the highest value for the bundle $S' = \{A, B, C\}$. Say for example $v_2(\{A, B, C\}) = 12$, so the seller-optimal solution (call it $(\hat{\pi}, \hat{p})$) would have $\pi_2 = 0$ and $p(\{A, B, C\}) = 12$. Now the same situation as described in the counterexample of Section 7.5.3, where in fact this cut is not Pareto optimal. It is dominated by the cut which is identical except for $\bar{\pi}_2 = 1$, $\bar{p}(\{A, B, C\}) = 11$, $\bar{p}(\{A, C\}) = 8$, and $\bar{p}(\{B\}) = 4$.*

It can be quickly ascertained that the new cut $(\bar{\pi}, \bar{p})$ is valid, and that for the partition $\sigma^Y = \{\{A, C\}, \{B\}, \varnothing\}$,

$$V(\bar{\pi}, \bar{p}, \sigma^Y) = 4 + 8 + 1 = 13$$
$$V(\hat{\pi}, \hat{p}, \sigma^Y) = 5 + 9 + 0 = 14$$

and therefore $V(\bar{\pi}, \bar{p}, \sigma^Y) < V(\hat{\pi}, \hat{p}, \sigma^Y)$ and $(\bar{\pi}, \bar{p})$ dominates $(\hat{\pi}, \hat{p})$, so $(\hat{\pi}, \hat{p})$ could not be Pareto optimal if this were the case.

In this case, however, the seller-optimal solution is indeed Pareto optimal. We'll use the $p(S)$ as above to add a constraint to the modified master.

Round 1, Modified Master: With our dual solution, we'll add to create MP^1 the following constraint, given first in the general form and then in this instance:

$$\theta \leq \sum_{\sigma \in \Pi} z_\sigma * \left(\sum_{S \in \sigma} p(S) \right) + \sum_{j \in N} \pi_j$$

$$\theta^1 \leq z_{\sigma^V} * (5 + 5 + 6) + z_{\sigma^W} * (11 + 0 + 0) + z_{\sigma^X} * (9 + 6 + 0)$$
$$+ z_{\sigma^Y} * (9 + 5 + 0) + z_{\sigma^Z} * (8 + 5 + 0) + 0 + 0 + 0$$

Solving MP^1 gives us the values $\theta^1 = 16$, $z_{\sigma^V} = 1$, all other $z_\sigma = 0$.

Round 2, Dual: Solving the assignment problem under σ^V, the center gives $\{C\}$ to Agent 1, $\{B\}$ to Agent 2, and $\{C\}$ to Agent 3. $D(\sigma^2) = 14$, $\pi_2 = 2$, $\pi_1 = \pi_3 = 0$.

$\theta^1 > D(\sigma^2)$, so we iterate another round. The "best" prices are:

S	$\varnothing$	$\{A\}$	$\{B\}$	$\{C\}$	$\{A,B\}$	$\{A,C\}$	$\{B,C\}$	$\{A,B,C)\}$
$p(S)$	0	5	3	4	9	7	8	11

Round 2, Modified Master: With the new constraint from (π^2, p^2), MP^2 is:

$$\begin{aligned} \text{MP}^2: \quad & \max \ \theta^2 \\ \text{s.t.} \quad & \theta^2 \leq 16\sigma^V + 11\sigma^W + 15\sigma^X + 14\sigma^Y + 13\sigma^Z \\ & \theta^2 \leq 12\sigma^V + 11\sigma^W + 13\sigma^X + 10\sigma^Y + 13\sigma^Z + 2 \\ \sum_{\sigma \in \Pi} & z_\sigma \leq 1 \\ & \theta^2 \geq 0, \quad z_\sigma \in \{0,1\}, \quad \forall \sigma \in \Pi \end{aligned}$$

with solution $\theta^2 = 15$, $z_{\sigma^X} = 1$, all other $z_\sigma = 0$. .

Round 3, Dual: Solving the assignment problem under σ^X, the center gives $\{A,B\}$ to Agent 1, and $\{C\}$ to Agent 2. $D(\sigma^2) = 15$, $\pi_2 = 2$, $\pi_1 = \pi_3 = 0$.

Now $\theta^2 = D(\sigma^3)$, so we have converged and σ^X is an optimal partition. For prices we use the prices from the final dual round, which are $p(\{A,B\}) = 9$ and $p(\{C\}) = 4$.

Summary: Using the "best" solution as described here - seller optimal in the case of the partition under consideration, and minimal supporting prices on all other bundles - speeds convergence in this example and may across most cases. However, it is important to recognize that this is considerably more difficult than just solving the assignment problem and generating the other prices in a simplistic manner. The information passed from agents to center is of the same size as a complete valuation over all "interesting" bundles in each round.

7.8 Computing Vickrey Payments

In order for bidding agents' compliance with the Benders auction to be in *ex post* Nash Equilibrium, we must ensure that the ultimate outcome corresponds with that of a Generalized Vickrey Auction. The Benders auction as we have already described it will allocate the goods in the optimal manner (assuming agents are truthful), but we also must compute Vickrey payments - the set of prices at which the auction terminates do not necessarily follow the second-price principle.

As in the Dantzig-Wolfe auction, two approaches to computing Vickrey prices immediately come to mind. The first is to solve each marginal economy E_{N-j} and compute the Vickrey payments accordingly. The second is to try to extrapolate the correct prices from the core of pricing information that the center has accumulated over the course of the auction. These approaches are discussed with respect to Dantzig-Wolfe in Sections 4.3.5-4.3.7.

The first approach, in more detail, is to use a Benders auction to solve each marginal economy E_{N-j} (the allocation problem excluding agent j), determine each value $V(N/j)$, and compute the Vickrey payments accordingly. This entails solving n problems of similar complexity to the main problem. With the Dantzig-Wolfe auction, the allocation for E_N provides an advanced starting point for the master problem from which to solve E_{N-j}. Starting with an allocation that is close to the optimum is less helpful to the Benders algorithm, however, since the center must obtain sufficient pricing information to rule out all other partitions, even if the optimal partition has already been considered. Thus, in re-solving for the marginal economies E_{N-j}, Benders must start almost entirely from scratch. This makes this approach comparatively less attractive than it is in the Dantzig-Wolfe case, but it is still a useful approach - it will yield the correct answer, just at high computational cost.

The other approach, that of extrapolating correct Vickrey prices from the information already accumulated, is more elegant than solving each marginal economy directly, but more difficult realize concretely. The idea would be to sift through the mountain of prices contained in the central constraints and find prices that are feasible solutions to the dual of each E_{N-j}, and thus support the optimal allocation for that marginal economy. Unfortunately, as in Dantzig-Wolfe, we have no theoretical guarantees at this point that all of the information necessary to produce Vickrey prices gets submitted to the center over the course of the auction. It is possible that the majority of such prices are, and the rest could be obtained with quick, direct queries rather than indirectly through another auction, but this may also be wishful thinking. A deeper understanding of the behavior of prices is needed to make stronger claims about the efficacy of the price extrapolation method; however, attention is deserved because it could provide a much more elegant way to obtain Vickrey prices, avoiding the heavy computational burden of solving each of the n marginal economies directly.

7.9 Summary of Benders on LP2

To summarize this chapter, applying Benders decomposition to Bikhchandani and Ostroy's LP2 formulation of the CAP has yielded a promising framework for implementing a combinatorial auction. LP2 is not useful for Dantzig-Wolfe because it does not guarantee integrality by itself, but Benders decomposition enables us to transform it into a mixed integer program which has provably integral optimal solution. The problem is decomposed into integral partition variables and continuous bundle-agent assignment variables. The former are fixed by the master problem, the Benders subproblem (the dual) is solved to generate an appropriate dual cut, and this cut is added to the modified master problem. This process is repeated until the modified master problem is sufficiently constrained to yield an optimal solution to the complete master problem. (The complete master problem, however, is never enumerated).

The auction interpretation of this is that the center fixes and announces a partition, and the agents respond with pricing information relevant to that partition. This information is incorporated by the center and a new partition is announced. This process is repeated until the center has enough pricing information to guarantee the optimality of an allocation.

The most challenging part of implementing this algorithm is decentralizing the Benders subproblem. Unlike Dantzig-Wolfe, Benders is not readily adaptable to taking advantage of additional block structure within a coefficient matrix after the initial decomposition. The most promising approach at this point seems to be solving the problem of allocating the partition as an assignment problem, then constructing a complete dual cut from the prices generated by the optimal assignment. Within this approach there are many possible avenues to explore. It seems at this point that aiming for Pareto optimal dual cuts is too ambitious a goal, given computational constraints. We can predict that the tension between the quality of dual cuts and the amount of computational effort we are willing to expend on them will likely characterize future efforts to implement this auction algorithm. The possibility of an extremely efficient method of determining Pareto optimal cuts seems slim, yet plausible; the potential rewards make it well worth investigating.

Like the Dantzig-Wolfe auction, the central agent in the Benders auction over LP2 continually improves its information over time, until it is finally able to announce an optimal allocation. Unlike Dantzig-Wolfe, the termination of the Benders algorithm is not marked by the bidding agents no longer responding with best-response bundles. Instead, the center determines when the auction is over on the basis of the information it has aggregated in the master problem, combined with the latest subproblem (dual) solutions.

Solving the modified master problem at each round is not trivial - it involves an exponential number of 0-1 variables z_σ. This could be mitigated somewhat by converting the formulation to bundle-specific decision variables $x(S)$ as mentioned in Section 7.3.

A potential disadvantage of the Benders auction is that structurally, every partition must be actively ruled out, as compared to Dantzig-Wolfe where they are implicitly ruled out as prices rise and agents choose not to request them. In many small examples with non-optimal dual cuts, the partitions were nearly or completely enumerated before the modified master problem was sufficiently constrained to guarantee an optimal overall allocation. This a a large concern, as the number of partitions is exponential - it would be very interesting examine the behavior of the algorithm experimentally on larger test cases to see if this is indeed an issue.

This chapter is just a beginning - there are myriad theoretical and practical issues left to address with regard to the Benders auction. Nevertheless, it seems to be a promising auction framework and deserves further consideration to explore its advantages and disadvantages and better understand what it has to offer.

Chapter 8

Benders on LP3 and LP4

Having gone into great depth on the Benders auction on LP2, this Chapter will briefly examine the problem structures resulting when the Benders decomposition is applied to the stronger LP3 and LP4 formulations. Since these linear program formulations are already guaranteed to be integral, there is no need to take advantage of the Benders decomposition's ability to mesh integral and continuous variables. One could hypothesize that the computational cost of the additional density in the problem formulations would make these options less attractive than LP2, which is not strong enough to guarantee integrality as a linear program but can as a Benders mixed integer program.

8.1 Mapping the Benders Decomposition to LP3

Recall that the LP3 formulation is stronger than LP2, guaranteeing an integral solution even if all variables are allowed to be continuous. LP3 differs from LP2 in that instead of anonymous partitions $\sigma \in \Pi$ and associated variables z_σ, LP3 uses non-anonymous partitions $\mu \in \Gamma$ and associated variables δ_μ. A non-anonymous partition not only specifies which bundles are available, but also which agents they are assigned to. The LP3 formulation is given once again below:

$$
\begin{aligned}
V(N) = \max \sum_{j \in N} \sum_{S \subseteq M} v_j(S) y(S,j) \\
\text{s.t.} \sum_{S \subseteq M} y(S,j) \le 1, \quad \forall j \in N \\
y(S,j) \le \sum_{\mu \ni S^j} \delta_\mu, \quad \forall j \in N, \forall S \subseteq M \\
\sum_{\mu \in \Gamma} \delta_\mu \le 1 \\
y(S,j) \ge 0 \quad \forall S \subseteq M, \forall j \in N
\end{aligned}
$$

Unfortunately, this formulation does not work well with Benders - if we fix the δ_μ in the first step, analogous to fixing z_σ for LP2, we have implicitly fixed all of the $y(S,j)$ variables as well, leaving basically no work for the second half (the agent half) of the decomposition procedure. A tight, minimal-priced dual solution will lead to fast convergence, but this is basically indistinguishable from the center soliciting a complete valuation from each bidding agent and solving the entire allocation problem itself in one go.

8.2 Mapping the Benders Decomposition to LP4

Unlike LP3, the LP4 formulation of Bikhchandani et al. [2001] guarantees integrality without the use of non-anonymous partitions, thus making it a more attractive candidate for the Benders approach. The LP4 formulation is:

$$
\begin{aligned}
V(N) = \max \sum_{\sigma \in \Pi} \sum_{j \in N} \sum_{S \subseteq M} v_j(S) y^\sigma(S,j) \\
\text{s.t.} \quad \sum_{S \in \sigma} y^\sigma(S,j) \le z_\sigma, \quad \forall j \in N, \forall \sigma \in \Pi \\
\sum_{j \in N} y^\sigma(S,j) \le z_\sigma, \quad \forall S \in \sigma, \forall \sigma \in \Pi \\
\sum_{\sigma \in \Pi} z_\sigma \le 1 \\
\sum_{\sigma \in \Pi} \sum_{S \in \sigma} y^\sigma(S,j) \le 1, \quad \forall j \in N \\
y(S,j) \ge 0 \quad \forall S \subseteq M, \forall j \in N
\end{aligned}
$$

N is the set of agents, M is the set of goods. $v_j(S)$ is agent j's value for bundle S, and $y^\sigma(S,j)$ equals 1 or 0 depending on whether agent j is allocated bundle S in partition σ.

We also have a variable z_σ for each possible partition $\sigma \in \Pi$. Π is the set of all possible partitions.

8.2.1 Logical Benders Decomposition of LP4

As in LP2, the logical decomposition is to fix the z_σ and leave the $y^\sigma(S,j)$ to vary. We then solve the dual with the z_σ fixed and use the solution to generate constraints in our modified master problem MP2 (a.k.a. MP^t in Round t). This process is iterated until MP2 is sufficiently constrained to yield a feasible optimum. We'll know this when the objective values of the dual and of MP2 coincide ($D(\sigma^t) = \theta^t$).

Since LP4 is proven to be integral even with all variables continuous [Bikhchandani et al., 2001], there is no need to prove its integrality after we constrain $z_\sigma \in \{0,1\}$.

8.2.2 Primal and Dual of LP4 with z_σ fixed

Recall that we have fixed the z_σ and chosen a partition σ. Similarly to LP2, we can write the following programs for dual, $D(\sigma)$.

$$\begin{aligned} D(\sigma) = \min & \sum_{j\in N} \pi_j + \sum_{S\in\sigma} w_S^\sigma + \sum_{j\in N} \mu_j^\sigma \\ \text{s.t.} \quad & \pi_j + \mu_j^\sigma + w_S^\sigma \geq v_j(S) \quad \forall j \in S, \forall S \in \sigma \\ & \mu_j^\sigma, w_S^\sigma, \pi_j, \pi^s \geq 0 \quad \forall j, S, \sigma \end{aligned}$$

The seller's surplus π_s is equivalent to $\sum_{S\in\sigma} w_S^\sigma + \sum_{j\in N} \mu_j^\sigma$. LP4 provides payments in the form of a non-anonymous component μ_j^σ and a non-linear component w_S^σ.

We can obtain bundle- and agent-specific prices by defining:

$$p_j(S) = \mu_j^\sigma + w_S^\sigma$$

and perhaps anonymous prices by saying

$$p(S) = \max_{j\in N}(p_j(S))$$

There are a tremendous number of variables in this dual problem, and we also need additional computation to generate bundle prices from the variable settings in the solution.

Hopefully we can just take the prices from the fixed σ, and generate the others analogous to the method used in our decomposition of LP2.

8.2.3 The Modified Master Problem, MP^t

The modified master problem MP^t is initiated as:

$$\begin{aligned} \text{MP}^t: \quad & \max \ \theta \\ & \sum_{\sigma \in \Pi} z_\sigma \leq 1 \end{aligned}$$

Having solved the dual and received back a set of prices that supports the given partition σ over all bundles $S \subseteq M$, we formulate a new constraint of the form

$$\theta \leq \sum_{\sigma \in \Pi} z_\sigma * (\sum_{S \in \sigma} w_S^\sigma + \sum_{j \in N} \mu_j^\sigma) + \sum_{j \in N} \pi_j$$

Solving this version of MP2 will give us a new setting for the z_σ, which will dictate a new partition σ to fix, and we'll repeat the Bender's process.

The structure we see here is very similar to that of the Benders auction on LP2, but with more redundancy in both variables and constraints that are artifacts from the original LP4 formulation, which required them to guarantee integrality as a pure linear program. Since this formulation is integral even as a pure LP, we could solve the master problem MP^t as a LP rather than a MIP, which would be faster in practice. However, it is unclear that the benefit derived from this would outweigh the computational cost of maintaining so much redundancy over the leaner LP2 formulation. Further exploration of the idea of designing a Benders auction by applying the decomposition to LP4 may be warranted, but at the present we conclude that the auction based on LP2 appears to be considerably more promising.

Chapter 9

Conclusion

This thesis proposes several new and specific frameworks to implement a Generalized Vickrey Auction in a distributed fashion. Each framework was generated by casting the CAP as a linear program and solving the problem using a decomposition method, which moves much of the computation from the central agent to the bidding agents. To ensure that agents do not attempt to manipulate the system, each framework implements a GVA outcome, which puts the entire mechanism in *ex post* Nash Equilibrium. Each framework thus succeeds in implementing a Vickrey-Clarke-Groves mechanism that incorporates self-interested agents into the computation, *without* sacrificing the quality of the solution. The agents cooperate and each contribute toward the socially optimal solution because it is in each of their own best interests.

We began by outlining the main ideas of distributed mechanism design and establishing the goal of designing an incentive-compatible, distributed auction based on applying decomposition methods to linear programming formulations of the CAP. The main body of the paper was divided into two major parts, the first devoted to the Dantzig-Wolfe decomposition and the second devoted to the Benders decomposition. Each part began by introducing the decomposition method in a general setting, then moved on to applying the decomposition to linear program formulations of the CAP and attempting to interpret the results in a meaningful way as an iterative auction.

The Dantzig-Wolfe auction was first introduced on the simple domain of the assignment problem, a.k.a. the special case of unit demand. This auction is based on linear relaxations of the integer program representation of the problem, which is then solved via the Dantzig-Wolfe decomposition algorithm. This process has a very natural auction interpretation, with the shadow prices from the dual in each round representing bundle prices. In theory, prices will rise over time on bundles containing over-demanded goods and fall on bundles containing under-demanded goods, until an equilibrium between prices and agent demand is reached. It was hypothesized that prices may be purely ascending - this is a very interesting possibility and we would eagerly anticipate the results of experimentation to further understand the behavior of these prices.

The Dantzig-Wolfe auction was then further described for the full domain of the combinatorial allocation problem, although this involves more sophisticated linear relaxations, LP3 and LP4, that do not conform as easily to the Dantzig-Wolfe structure. This formulation is theoretically valid - the biggest concern is the exponential number of variables, which may make the process very difficult computationally and perhaps unworkable for large examples, which would be unfortunate. We speculated on some steps that might help take the Dantzig-Wolfe auction from theory to efficient practice, among them generating partition variables in the master problem only as needed and exploiting the similar structure of consecutive master problems.

The Benders auction was first introduced on LP2, an intermediate linear programming formulation of the CAP which is not useful to Dantzig-Wolfe, but the structure of the problem matches well with the ability of the Benders decomposition to mesh integral and continuous variables. Benders was also applied to LP3 and LP4, but LP2 remained the most promising.

Benders decomposition does not have as natural an auction interpretation as does Dantzig-Wolfe, but nevertheless a meaningful interpretation was extracted and several examples of the working process of the auction were presented on a small scale. The key challenge arising in the Benders auction is the manner in which the dual subproblem is decentralized and solved. This subproblem bears some resemblance to the assignment problem, a similarity which can hopefully be exploited. We observed tension between the desire to generate the most binding (Pareto optimal) dual cuts and the desire to solve the subproblem without solving the entire dual. Generating Pareto optimal cuts seems quite impractical; on the other hand, without very strong cuts, we are concerned that all, or at least a majority, of the exponential number of possible partitions would have to be directly explored, which is unsatisfactory. While convergence of the Benders auction is guaranteed in theory, in practice we are concerned that it may take an unreasonable amount of time under some circumstances.

9.1 Future Work: Towards a Distributed VCG Implementation

Unfortunately, due to time considerations it was not feasible to implement either of the auctions proposed in this thesis. A strong next step would be to code up one or both of the auctions, assemble an appropriate suite of test cases, run the auction on the test cases and analyze the results. The optimality and eventual convergence of both auction algorithms is guaranteed theoretically; however, it would be very interesting to see how they behave in practice. In particular, the movement of the shadow prices across rounds in the Dantzig-Wolfe auction has already been pointed out as an area that merits experimentation and deeper understanding. Also, the convergence behavior of both algorithms is an issue of concern, particularly with the Benders, which could prove impracticable if the number of iterations needed to sufficiently constrain the master problem is too high. Finally, empirical work would us to better grasp the magnitude of the computational challenge presented by these algorithms, how much it restricts us, and to what extent improvements to improve efficiency are necessary to use the Dantzig-Wolfe and/or Benders auctions in large-scale real-world applications.

The biggest area for further exploration is how to translate the Dantzig-Wolfe and Benders auctions into versions that are manageable from a computational standpoint. Other interesting and related issues include

- The best way to solve the marginal economies E_{N-j} and compute Vickrey payments in both Dantzig-Wolfe and Benders
- The similar structure of the iterated master program in Dantzig-Wolfe
- The multiplicity of dual solutions in Dantzig-Wolfe
- The behavior of Dantzig-Wolfe prices - will they always be ascending?
- The structure of the Benders modified master problem, and potential alternate representations ("one-constraint," bundle decision variables, price-updating scheme)
- Strong and Pareto optimal dual cuts in Benders, and the tradeoff between ease of obtaining the cuts and their quality
- Convergence behavior of both algorithms, particularly Benders
- The possibility of decentralizing the master problem itself, in order to obtain a completely decentralized mechanism

Both the Dantzig-Wolfe auction and the Benders auction have exciting potential to develop into a formidable distributed implementation of the VCG mechanism. We hope that this thesis inspires interest and future research in this area.

Bibliography

J. Abrache, T.G. Crainic, and M. Gendreau. Decomposition methods and iterative combinatorial auctions. 2004.

D. Bertsekas. The auction algorithm: A distributed relaxation method for the assignment problem. 1988.

S. Bikhchandani and J. Ostroy. The package assignment model. 2001.

S. Bikhchandani, S. de Vries, J. Schummer, and R. Vohra. Linear programming and vickrey auctions. 2001.

S. Bradley, A. Hax, and T. Magnanti. *Applied Mathematical Programming*. Addison Wesley, 1977.

V.P. Crawford and E. Knoer. Job matching with heterogeneous firms and workers. 1981.

G.B. Dantzig. *Linear Programming and Extensions*. Princeton University Press, 1963.

G. Demange, D. Gale, and M. Sotomayor. Multi-item auctions. 1986.

R. Garg and D. Mishra. Descending price multi-item auctions. 2004.

S. Kameshwaran. Algorithms for piecewise linear knapsack problems with applications in electronic commerce. 2004.

E. Kwerel and J. Williams. A proposal for a rapid transition to market allocation of spectrum. 2000.

L. Lasdon. *Optimization Theory for Large Systems*. Dover, 2002.

V. Lesser. Cooperative multiagent systems: A personal view of the state of the art. 1999.

T. Magnanti and R. Wong. Accelerating benders decomposition: Algorithmic enhancements and model selection criteria. 1979.

D. Parkes. Iterative combinatorial auctions. 2001.

D. Parkes and J. Shneidman. Distributed implementations of vickrey-clarke-groves mechanisms. 2004.