Designing Mechanisms for Sustainable Carrier Alliances in Transportation Networks

Özlem Ergun, Richa Agarwal
School of Industrial and Systems Engineering
Georgia Institute of Technology, Atlanta, GA 30332-0205, USA

1 Introduction

Traditionally, companies and business processes in transportation and supply chain logistics have focused on their own resources and ability to improve performance. However recently, when faced with pressures to operate more efficiently, companies are realizing that suppliers, consumers and even competitors can be potential collaborative logistics partners. The system wide collaboration perspective provides opportunities for increased profitability that are impossible to achieve with an internal focus only. Various technological advancements, such as the Internet providing the connectivity and infrastructure necessary to easily share large volumes of data, are fuelling the trend towards collaboration among different companies. In sea cargo transportation, since 1990 when Sea-Land and Maersk introduced the alliance system and began sharing capacity on ships in the Atlantic and Pacific oceans, mergers have become increasingly common. Since owning a ship involves large capital investment (usually millions of US dollars) and the cost of idling a ship runs in tens of thousands of dollars per day, carriers collaborate and form alliances to share infrastructural setup and capital costs.

In this paper, we study large scale transportation networks that operate as an alliance among carriers. Our study is motivated by the *service network design* problem in the liner shipping industry. Liner shipping mainly involves carrying cargo stored in containers on regularly scheduled service routes. Liner services involve high fixed costs and administrative overhead because they promise to depart on a predetermined schedule regardless of whether the ship is full. The number of ships required for a given liner service route is determined principally by the frequency required on the service route. For example, a weekly liner service between New York and Hamburg may require four ships to maintain the necessary frequency. In the liner shipping network design problem we are given a set of carriers and a set of ports. Each carrier has a fleet of ships and a set of cargo to be delivered. Delivering a unit of the cargo earns a given revenue. To form an alliance, carriers bring their fleets into a pool and operate them together. The service network is designed by creating the ship routes, i.e. the sequence of port visits by the given fleet. Ships move in cycles, referred to as *service routes* from one port to another following the same port rotation for the entire planning horizon. Further, carriers decide which cargo to accept or reject for servicing and which *path(s)* to use to deliver the selected cargo. The cargo is allowed to travel on ships on multiple service routes before reaching its final destination. A port where cargo is transferred from one ship to another, for further transportation, is referred to as a *transshipment port*. Once a set of service routes is decided, members of the alliance assign their ships for operating the chosen routes and allocate each ship's capacity among the alliance members.

Alliance formation among carriers poses various challenges. In an alliance, carriers work in collaboration with each other, however each carrier's individual goal remains to be the maximization of his own benefit. Hence, for forming sustainable alliances, the task is not only to design an efficient service network but also to provide algorithms to share the benefits and costs of an alliance in such a way that all carriers are motivated to collaborate. In this setting, sharing benefits and costs generally translates into exchanging capacity on ships among the carriers. One way to regulate capacity exchanges among the carriers is to assign suitable *capacity exchange costs* so that the carrier who owns the capacity on a ship is motivated to sell the capacity to a carrier who can utilize it to deliver cargo.

Problems where a number of players interact, with varying degree of collaboration and self motives, appear not only in transportation networks but in a wider variety of seemingly unrelated fields such as internet routing, auctions, telecommunications, etc. The mathematical tools and insights most appropriate to understand these problems are obtained by uniting concepts of mathematical economics and game theory with that of algorithm design. The notion in cooperative game theory that is most relevant to us is that of "mechanism design". The traditional mechanism design problem concerns a set of players $\mathcal{N} = \{1, 2, \ldots, n\}$ who collaboratively choose an outcome $\tilde{o}$ from a set $\mathcal{O}$ of possible outcomes. Each player k has a preference relation for different outcomes $\tilde{o}$ in the set $\mathcal{O}$, denoted by $v_k(\tilde{o})$, to quantify his valuation of outcome $\tilde{o}$. The goal is to design an algorithm that chooses an outcome, $\tilde{o} \in \mathcal{O}$ and an n-tuple of side payments $\{s_1, \cdots, s_n\}$ such that the total payment, x_k, to player k is $x_k = v_k(\tilde{o}) + s_k$. The total payment is what each individual player aims to optimize. Intuitively, a mechanism solves a given problem by providing side payments to players to assure that the required output occurs, when players choose their strategies so as to maximize their own selfish profits. Mechanism design has been used successfully to develop algorithms for inter-connected collection of computers such as on the internet. For a detailed discussion of mechanism design we refer the reader to [2].

Network design problems, with minimum requirements for service frequency, are common in the transportation industry. In this paper we address various challenges offered by a generic carrier alliance and provide mechanisms to make it sustainable. To keep the discussion general, we refer to the fleet of ships as *assets* and the set of carriers as *players.* As players form alliances by pooling their assets and integrating their networks, a large scale optimization problem can be solved to obtain the best centralized service network. We develop a mathematical model that can be used to solve large instances of the network design problem. Our model successfully incorporates many emerging trends for example the minimum frequency constraint and transshipments.

Individual players working in an alliance cannot be assumed to accept the network designed by the centralized optimization algorithm but follow their own self-interests. The split of benefits and costs, to satisfy all the members of an alliance, is an intriguing research topic and very little is available in literature on the systematic study of alliances in transportation networks. We design allocation mechanism to distribute the benefits and costs among the members of an alliance. To help the overall alliance achieve its maximum potential revenue, we provide incentives to the players to pursue the solution suggested by the centralized optimization model. The benefits obtained from the collaborative routes directly are often not enough to motivate individual players to behave in the best interest of the alliance. We provide side payments, via the capacity exchange cost, to the players so that they are motivated to "play along". Next, we present a formal definition of our problem.

2 Problem Definition

Let V denote the set of nodes in the network and $\mathcal{N} = \{1, 2 \dots n\}$ denote the set of players. We assume that all assets are identical with T units of capacity and for a player $k \in \mathcal{N}$, N_k represents the number of assets in his fleet. For a player k, each demand is characterized by its origin (o)-destination (d) pair, the maximum demand that can arise, $D^{(o,d,k)}$, and the revenue obtained by satisfying one unit of demand, $R^{(o,d,k)}$. (o, d, k) is used to identify a demand from o to d for player k and the demand set of player k is identified by Θ_k. In an alliance formed by pooling assets $N = \sum\limits_k N_k$ and consolidating demand $\Theta = \cup_k \Theta_k$, the players face following problems:

1. Together they need to design their service network. For this, they need to design a set of service routes (say $\overline{C} = \{C_1, \cdots, C_r\}$) to operate utilizing their assets. Also, they need to decide a set of cargo (say $\overline{\Theta} \subset \Theta$) to deliver and the paths to use to deliver the selected cargo.

2. The members of an alliance need to decide how to realize the service routes in $\overline{C}$. For example, they need to decide the number of assets that each player should assign to the service routes in $\overline{C}$.

3. Each player k needs to compute the valuation, v_k, of the solution, given by $(\overline{C}, \overline{\Theta})$, depending on the cost incurred by him and the revenue generated by delivering his demands.

4. For a given $(\overline{C}, \overline{\Theta})$, since the valuation v_k alone is not enough to guarantee the satisfaction of player k, the alliance needs to decide an n-tuple of side payments, $\{s_1, \cdots, s_n\}$, for its members such that the total payment, $x_k = v_k + s_k$, to player k is optimal for him.

3 Solution Strategy

Before providing the details, we first present an outline of our solution strategy. The goal of an individual player is to design a service network which maximizes his profit. However, since he is working in collaboration with other players, a network that generates maximum overall revenue for all players is selected. Clearly, such a network can be obtained by replacing the individual players with one large player, with a fleet equal to the combined fleet of individual players and a demand structure equal to the combined demand of all players. We refer to the collaborative solution by $opt(\mathcal{N}) = (\overline{C}, \overline{\Theta})$. Section 3.1 describes in detail the optimization problem to obtain $opt(\mathcal{N})$. In Section 3.2, the valuation of solution $opt(\mathcal{N})$ is calculated for each player by calculating the revenue generated by him and the costs incurred by him. The valuation obtained from solution $opt(\mathcal{N})$ is not guaranteed to provide enough motivation for a player to act according to the schedule $opt(\mathcal{N})$. To provide this guarantee side payments are provided to the players. Side payments are made by a player for utilizing capacity on an asset to the owner of the asset via capacity exchange costs. Computation of side payments using inverse optimization techniques is presented in Section 3.3. Next, we provide the details of our solution strategy.

3.1 Network Design

The set of service routes determine which paths can be designed to deliver the cargo. The cargo delivered, and the paths chosen to deliver the cargo, determines the revenue that can be generated and hence determines the profitability of the service network. Thus, these two problems are highly inter-dependent and it is important that they be studied in an integrated framework. For a set of players and a network $G = (V, E)$, we now present a mixed integer programming model, CP, to determine an optimal set of service routes to operate, set of cargo to deliver and paths to deliver the selected cargo, simultaneously. Let $\mathcal{C}$ denote the set of all feasible service routes. For $C \in \mathcal{C}$, L_C denotes the number of assets required to maintain the minimum required frequency on C and $Cost_C$ denotes the cost of operating route C. CP has two sets of variables: binary variables x_C for $C \in \mathcal{C}$ to denote if route C is operated and non-negative continuous variables $f_e^{(o,d,k)}$ to represent the flow of cargo from origin o to destination d due to player k on edge e.

$$(CP): \quad r(opt(\mathcal{N})) = \max \sum_{(o,d,k)\in\Theta} f_{(d,o)}^{(o,d,k)} R^{(o,d,k)} - \sum_{C\in\mathcal{C}} Cost_C x_C \tag{1}$$

$$\text{subject to} \quad \sum_{e\in InEdges(v)} f_e^{(o,d,k)} - \sum_{e\in OutEdges(v)} f_e^{(o,d,k)} \leq 0 \forall v \in V, \forall (o,d,k) \in \Theta \tag{2}$$

$$\sum_{(o,d,k)\in\Theta} f_e^{(o,d,k)} - \sum_{\{C\in\mathcal{C}: e\in C\}} T x_C \leq 0 \ \forall e \in E \tag{3}$$

$$f_{(d,o)}^{(o,d,k)} \leq D^{(o,d,k)} \quad \forall (o,d,k) \in \Theta \tag{4}$$

$$\sum_{C\in\mathcal{C}} L_C x_C \leq N \tag{5}$$

$$x_C \in \{0,1\} \ \forall C \in \mathcal{C} \ \text{ and } \ f_e^{(o,d,k)} \geq 0 \qquad \forall e \in E, \forall (o,d,k) \in \Theta. \tag{6}$$

In the above formulation, the objective function (1) maximizes the net profit by subtracting the sum of operating costs from the revenue generated. Constraint (2) is a flow balance constraint at every vertex of the network. Constraints (3) and (4) are capacity constraints on the edges. Constraint (5) requires that we do not use more assets than we have available.

In earlier research [1], we studied CP in detail and showed that the problem is NP-hard. Further we developed and computationally tested various heuristic and LP based algorithms to solve the problem. Let $opt(\mathcal{N}) = (\overline{C}, \overline{\Theta})$ denote an optimal solution to the above problem, where $\overline{C}$ is the set of optimal service routes to be operated by the combined fleet and $\overline{\Theta}$ is the subset of demand that is delivered. Let $\overline{f}$ denote the optimal flow on edges.

3.2 Valuation of the Schedule

For a player, the valuation of solution $opt(\mathcal{N})$ is determined by calculating the revenue generated by him and the costs incurred by him. The revenue generated by a player k is calculated by summing over the revenue generated by satisfying demand (o,d,k) such that $(o,d,k) \in \overline{\Theta} \cap \Theta_k$. Similarly, each player pays for maintaining and operating his assets on the collaborative routes. To compute the costs incurred, a player first needs to know the assignment of his assets to the selected routes. Let, $y_{C_j}^k$ represents the number of assets player k assigns to route C_j and $u_{C_j}^k$ represents the utility he obtains by assigning a unit of asset on route C_j. The problem of optimally assigning assets, for all the players, to the set of selected service routes reduces to the following generalized assignment problem:

$$(AAP): \quad \max \sum_{k,C_j} u_{C_j}^k y_{C_j}^k \tag{7}$$

$$\sum_k y_{C_j}^k = L_{C_j} \qquad \forall C_j \in \overline{C}; \qquad \sum_{C_j} y_{C_j}^k \leq N^k \qquad \forall k \in N; \qquad y_{C_j}^k \quad int. \tag{8}$$

The utility function, u, can be determined heuristically in many different ways. We solve the asset assignment problem, AAP, exactly and heuristically and report the effect of different asset assignment algorithms and different utility functions on the overall mechanism.

Once the assignment problem is solved, we can calculate the valuation of solution $opt(N)$ for player k, that is:

$$v_k(opt(\mathcal{N})) = \sum_{(o,d,k)\in\Theta_k} R^{(o,d,k)}\overline{f} - Cost\ of\ operating\ routes. \tag{9}$$

3.3 Computation of Side Payments

We model the selfish behavior of players by assuming that given the collaborative network the players solve their cargo routing problems individually. Given an assignment of assets, it is in the best interest of the collaboration that the players make their cargo routing decisions as in $\bar{f}$. Note that $\bar{f}$ requires players to share capacity on the assets. We facilitate this by allowing a player to charge other players for using his assets on an edge e whenever he has an asset assigned to edge e. The rest of the times he will need to pay other players for using capacity on edge e. We refer to this payment as the capacity exchange cost on edge e and denote it by $cost_e$. Note that the capacity exchange costs provide the side payments to the players, in addition to the valuation (9) obtained by them. Given an assignment of routes in $\overline{C}$ among agents, let γ_e^k be the fraction of times agent k has an asset on edge e. We calculate γ_e^k by a heuristic algorithm. Then we propose the following mathematical formulation for modelling an individual player k's behavior in the alliance:

$$(SCP^k):$$

$$\max \sum_{(o,d,k)\in\Theta_k} f_{(d,o)}^{(o,d,k)} R^{(o,d,k)} + \sum_{e\in E_v}\Big(\sum_{(o,d,k)\notin\Theta_k} \gamma_e^k f_e^{(o,d,k)} - \sum_{(o,d)\in\Theta_k} (1 \quad -\gamma_e^k) f_e^{(o,d,k)}\Big) cost_e \tag{10}$$

$$\text{subject to} \quad \sum_{e\in InEdges(v)} f_e^{(o,d,k)} - \sum_{e\in OutEdges(v)} f_e^{(o,d,k)} \leq 0 \qquad \forall v\in V, \forall (o,d,k)\in\Theta \tag{11}$$

$$\sum_{(o,d,k)\in\Theta} f_e^{(o,d,k)} \leq \overline{Cap_e} \qquad \forall e\in E \tag{12}$$

$$f_{(d,o)}^{(o,d,k)} \leq D^{(o,d,k)} \qquad \forall (o,d,k)\in\Theta \tag{13}$$

$$f_e^{(o,d,k)} \geq 0 \qquad \forall e\in E, \forall (o,d,k)\in\Theta \tag{14}$$

Note that this is a conservative model since we allow an individual player to modify other player's flow also. In practice, an individual can only make decisions regarding his own flow. Thus the maximum revenue that player k can obtain will always be less than the optimal value of SCP^k.
Assignment of Prices to Network Legs For the single player problem SCP^k we wish to identify $cost$ such that the collaborative solution, $\overline{f}$ is an optimal decision for the player also. Next, we make use of inverse optimization techniques to compute the cost structure that achieves this goal. Let $\pi^k = \{\pi_v^{(o,d,i),k} : \pi_v^{(o,d,i),k} \geq 0 \quad \forall v\in V,\ \forall (o,d,i)\in\Theta\}$, $\lambda^k = \{\lambda_e^k : \lambda_e^k \geq 0 \quad \forall e\in E\}$ and $\omega^k = \{\omega^{(o,d,i),k} : \omega^{(o,d,i),k} \geq 0 \ \forall (o,d,i)\in\Theta\}$ denote the dual variables associated with constraints (11), (12) and (13) respectively. Note the use of super-script k to denote that the dual is considered for player k. Below we present the dual of SCP^k for player k in a compact form:

$$(DSCP^k): \quad \min \sum_{e\in E} \overline{Cap_e}\lambda_e^k + \sum_{(o,d,i)\in\Theta} \omega^{(o,d,i),k} D^{(o,d,i)} \tag{15}$$

$$\text{subject to} \ \ \alpha\Delta^k - \gamma^k cost \geq 0. \tag{16}$$

One form of the linear programming optimality conditions states that primal solution f and dual solution $(\pi^k, \lambda^k, \omega^k)$ are optimal for their respective problems if f is feasible for SCP^k, $(\pi^k, \lambda^k, \omega^k)$ is feasible for $DSCP^k$, and together they satisfy the complementary slackness conditions.

Our aim is to determine a cost vector *cost* such that the flow $\overline{f}$ as given by $opt(\mathcal{N})$ is optimal for all individual agent problems i.e. SCP^k, $\forall k \in \mathcal{N}$. From above, $\overline{f}$ is an optimal solution for every SCP^k if for every $DSCP^k$ there exists a dual feasible solution $(\overline{\pi}^k, \overline{\lambda}^k, \overline{\omega}^k)$ that satisfies the primal-dual complementary slackness conditions. This gives us the following characterization of the inverse optimization problem to determine the cost vector such that the dual variables and the dual constraints satisfy the complementary slackness conditions, with respect to the primal solution $\overline{f}$:

$$(INVP): \quad \alpha\Delta^k - \gamma^k cost \geq 0 \quad \text{for } 1 \leq k \leq n. \tag{17}$$

To sum up, the inverse problem is a feasibility problem to identify *cost*. If $\overline{cost}$ is a feasible solution to $INVP$, the overall payoff to an agent k can be written as:

$$x_k \quad = \quad v_k(opt(\mathcal{N})) + s_k \tag{18}$$

where the vector of side payments $\{s_1, s_2, \cdots, s_n\}$ is calculated by,

$$s_k = \sum_{e \in E} \Big(\sum_{(o,d,i) \notin \Theta_k} \gamma_e^k f_e^{(o,d,i)} - \sum_{(o,d,i) \in \Theta_k} (1 - \gamma_e^k) f_e^{(o,d,i)} \Big) \overline{cost_e}. \tag{19}$$

Note that the vector of payoffs to agents $\{x_1, x_2, \cdots, x_n\}$ is such that $\sum\limits_{k \in \mathcal{N}} x_k = r(opt(\mathcal{N}))$. This is easy to see since once a feasible solution is found for $INVP$, the flow in the network is the same as the flow of optimal solution $opt(\mathcal{N})$. Also note that the vector of side payments $\{s_1, s_2, \cdots, s_n\}$ is such that $\sum\limits_{k \in \mathcal{N}} s_k = 0$. Theorem 1 guarantees that such a cost vector can always be found.

Theorem 1. *The inverse problem $INVP$ is feasible.*

To prohibit players from colluding to form sub-coalitions, we in fact want to have a cost vector such that for any subset $S \subset \mathcal{N}$, $\overline{f}$ is an optimal solution for the corresponding problem SCP^S. However, there are exponential number of such subsets and including an inverse problem corresponding to each of them in $INVP$ will cause $INVP$ to become exponential in size. Theorem 2 shows that it is sufficient to consider only single player problems in $INVP$ to determine a suitable cost vector.

Theorem 2. *Given an allocation of routes in $\overline{C}$ among players in $\mathcal{N}$, the inverse problem in ($INVP$) identifies a cost vector such that $\overline{f}$ is optimal for any subset $S \subset \mathcal{N}$ of players.*

It is reasonable to assume that an individual player would seek higher payoff in the alliance as compared to the revenue that he can generate on his own. To this end, we have enhanced our model by adding the following set of limited rationality constraints in $INVP$:

1. $opt(\{k\}) \geq x_k$ for each $k \in \mathcal{N}$.

2. $opt(\{k, i\}) \geq x_k + x_i$ for $k, i \in \mathcal{N}$.

where, $opt(S)$ for $S \subset \mathcal{N}$ is the maximum revenue that the players in set S can obtain, when working on their own.

4 Computational Experiments

Next, we present results of our computational experiments. The focus of our computations is to study the performance of the mechanism designed in this paper in the context of liner shipping. We test the performance of our mechanism by comparing the payoff allocation made by our mechanism to the standard game theoretic concepts such as the core. The notion of core is one of the most prominent and widely accepted notion of "fair" allocation of costs and benefits in cooperative game theory and it is similar to the idea of Nash equilibrium in non-cooperative game theory. An allocation of benefits is said to be in the core if the sum of the payoffs over all players is their maximum attainable profit (budget balance property) and no subset of players can collude and obtain a better payoff for its members (stability property). Mathematically, a payoff vector x is said to be in the core if:

$$\sum_{i \in N} x^i = opt(N) \tag{20}$$

$$\sum_{i \in S} x^i \geq opt(S) \quad \forall S \subset N. \tag{21}$$

For an allocation in the core, the grand coalition is perceived as fair and is not threatened by its sub-coalitions. A payoff allocation in the core represents a very strong type of stability and provides a fair allocation.

We performed our computations on instances involving up to 10 ports with up to 27 demand triplets and 50 ships. The data generation is explained in the next section. All of our algorithms are implemented in C++ in an Unix environment. We also made extensive use of the callable libraries in CPLEX 9.0. All computational experiments were performed on a Sun280R system with UltraSparc-III processor. Results are reported on 50 randomly generated instances in each test class.

4.1 Data Generation

We performed our experiments on the data simulating real life data from the liner shipping industry. We assume that all ships are identical with 4000 TEU (twenty foot equivalent unit) of capacity. Sailing distance between ports are chosen randomly from [2, 30] or [14, 42] to simulate intra-region and inter-region distance, in days, between ports in Asia and North America. Origin-destination pairs are chosen randomly from the pairs of ports. Demands are randomly generated from [0.1, 1.0] fraction of the capacity of the ship. Finally, revenue generated by satisfying a unit demand is chosen to be in direct proportion of the distance between the origin port and the destination port of the demand. Different classes of random instances are generated to test the robustness of the algorithm. Classes are characterized by specifying the number of ports (P), the number of ships (S) and the number of demand triplets (D). For example an instance with 6 ports, 30 ships and 18 demand triplets is represented as $P6S30D18$. We consider networks with up to 10 ports and fleet sizes with upto 80 ships. This data generation scheme is same as used by [1]. Please refer to [1] for further details and justification of various parameters regarding data generation.

We consider alliances with two, three and four carriers. Each demand and ship is assigned to one of the carriers with equal probability, unless specified otherwise.

4.2 Effect of Ship Assignment and Rationality Constraints

Recall that the problem of assigning ships of different carriers on the service routes is formulated as (AAP) in Section 3.2. This problem is NP-hard. However, in our case since the number of selected service routes are between 3 and 10 (depending on the problem size), an explicit enumeration scheme can also be used to determine the exact assignment of ships to the service routes. Next, we study how the mechanism is effected by different assignment of ships on the service routes. We obtain different assignment by considering different algorithms to solve (AAP). In particular, we consider an exact assignment, a greedy assignment and a random assignment of the ships. We also consider two different utility functions (u in the objective function of (AAP)) - 1. the utility of assigning a ship to a service route for a carrier is taken proportional to the sum of his flow on the edges of the service route (denoted by f) and 2. the utility of assigning a ship to a service route for a carrier is taken to be proportional to the sum of his profit generated from his flow on the edges of the service route (denoted by f.R).

We also study the effect of enhancing the inverse problem $(INVP)$ by adding rationality constraints. As mentioned at the end of Section 3.3, rationality constraint for a subset S of carriers states that carriers in S seek higher payoff in an alliance as compared to the payoff they can generate on their own. We divide the rationality constraints into different sets, depending on the number of carriers considered. For example, two carrier rationality constraints is the set of rationality constraints for all subsets with two carriers and is denoted by $\{2\}$. To study the effect of rationality constraints we introduce one set of rationality constraints at a time to the inverse problem $INVP$. Inverse program together with all the single carrier rationality constraints is denoted as $INVP + \{1\}$ and inverse program together with all the single carrier and two carrier rationality constraints is denoted as $INVP + \{1\} + \{2\}$. Recall that $INVP$ is a feasibility problem and the payoff allocation made by our algorithms is always budget balance. Thus for a three carrier alliance if $INVP + \{1\} + \{2\}$ is feasible than it means that a cost structure that yields payoff allocation in the core can be identified.

Table 1 reports the effect of different assignment of ships on the service routes and the effect of rationality constraints on the solution quality for 3 carrier alliances and different test classes. In this table, we use $\{0\}$ to denote that $INVP$ is solved, $\{1\}$ to denote that $INVP + \{1\}$ is solved and $\{1\} + \{2\}$ to denote that $INVP + \{1\} + \{2\}$ is solved. The first column denotes different problem classes. It indicates the total number of ports, ships and demand pairs considered. Each demand and ship is assigned to one of the three carriers with equal probability. The next three columns report the number of instances (out of 50) for which an allocation in the core is found when the AAP is solved exactly and the utility function is taken to be proportional to the flow times the revenue. The second column reports this number when the inverse problem $INVP$ is solved. The third, and fourth column report these numbers when the inverse problem $(INVP)$ is solved together with all single carrier rationality constraints and $(INVP)$ is solved with all single and two carrier rationality constraints, respectively. The next three triplet of columns report the corresponding numbers when AAP is solved exactly, greedily and randomly, respectively. In these cases the utility function is taken to be proportional to the flow.

Different assignment algorithms and utility functions result in different number of ships being assigned by a carrier to each of the selected service routes. This in turn influences a carrier's valuation (9) of the optimal solution and the way the optimal solution is realized by the alliance. Note that the inverse problem computes the cost structure for a given assignment of ships to the service routes. Table 1 suggests that the number of cases for which the mechanism successfully finds a cost structure does not depend significantly on the assignment of ships to the service routes.

Table 1: Effect of asset assignment and rationality constraints.

Test Class	Exact						Greedy			Random		
	f.R			f			f			f		
	{0}	{1}	{1}+{2}	{0}	{1}	{1}+{2}	{0}	{1}	{1}+ {2}	{0}	{1}	{1}+{2}
P6S18D6	9	15	46	8	16	46	7	16	46	6	17	47
P6S18D9	12	17	47	9	16	46	10	19	48	8	17	47
P6S30D6	4	12	35	4	13	35	4	13	35	4	14	36
P6S30D9	10	22	46	10	18	45	7	17	45	9	20	46
P10S30D18	15	18	50	14	17	49	16	17	50	14	20	50
P10S30D27	11	9	47	11	7	47	10	8	47	10	8	48
P10S50D18	15	13	48	15	12	48	11	18	48	7	18	49
P10S50D27	17	12	50	20	18	50	17	20	50	20	20	50

If we consider all the rationality constraints, even for a random assignment of ships to the service routes, in most of the instances the mechanism finds a cost structure that yields an allocation in the core.

From Theorem 1, the inverse program $INVP$ is feasible. We found in our computational study that inverse problem together with single carrier rationality constraints is also feasible in all the instances. However, in some cases a feasible solution for $INVP + \{1\} + \{2\}$ could not be found. Note that for three carrier alliances a solution to $INVP+\{1\}+\{2\}$ means that a cost structure that yields payoffs in the core can be identified. For $INVP$ and $INVP + \{1\}$ we report if the feasible solution provided by CPLEX is in the core. For these cases there might be alternate solutions and some might provide an allocation in the core (for example, instances in which we find an allocation in the core by considering $INVP + \{1\} + \{2\}$ but not when we consider $INVP$ or $INVP + \{1\}$).

Table (1) suggests that a feasible solution to ($INVP$) in 10-25% of the instances directly yields an allocation in the core. As the inverse problem is constrained by adding single carrier rationality constraints in 25-45% of the instances the feasible solution yields an allocation in the core. Further $INVP + \{1\} + \{2\}$ is feasible in 70-95% of the cases, depending on the test class. Thus in 70-95% of the cases our mechanism provides an allocation in the core. Recall that it is not necessary that an instance will have a non-empty core.

4.3 Analysis of Different Test Classes

We analyze different test classes in Table 2. We consider alliances with three carriers. Ships and demand pairs are distributed uniformly among the carriers. We solve the ship assignment problem exactly and the utility of assigning a ship to a service route for a carrier is taken proportional to the sum of his flow times the revenue on the edges of the service route. First column in Table 2 denotes different problem classes. For each test class, the second column reports the average CPU time taken (averaged over 50 instances) in minutes to solve a problem instance. This includes the time to solve the service design problem for all the subsets of carriers and the time taken to solve the inverse problem. The third column represents the number of instances, out of a total of 50 random instances generated for each test class, for which an allocation in the core exists. To test if the core of a problem is non-empty, a linear program consisting of all the core inequalities is constructed and its feasibility is tested. The next three columns report the average percentage of unsatisfied demand to the total demand, the average number of un-utilized ships and the average utilization of capacity on the edges of the network respectively, for the instances with a non-empty core. The next three columns report same statistics for the cases with empty core.

The second column in Table 2 suggests that as the problem size (number of ports, ships or demand pairs) increases the time taken to solve the problem increases. Also, more than 95% of the time reported here is taken in solving the network design problem for various subset of carriers. The increase in time taken to solve the network design problem with the increase in problem size is similar to the trend reported in [1].

Note that among the test classes with 6 ports, P6S30D6 has the highest number of instances with an empty core. A closer look at Table 2 reveals that this test class has the highest number of un-used ships and the lowest percentage of unsatisfied demand. Also instances in this class have lower utilization of capacity on the edges of the network. More specifically, these networks have over capacity. We constructed an additional class of instances namely P10S50D10. These instances also have over-capacity and show similar behavior as that of instances in the P6S30D6 class. Specifically, many instances in test class P10S50D10 also have empty core. This leads us to the conclusion that instances with over-capacity are more likely to have an empty core. The primary motivation for

Table 2: Analysis of test classes.

Test Class	Time	# Non -empty core	Non-empty core			Empty core		
			%Unmet demand	Unused ships	%Utilizi -ation	%Unmet demand	Unused ships	%Utiliz -ation
P6S18D6	1.92	48	21.9	0.5	0.70	41.6	1	0.59
P6S18D9	3.08	49	36.93	0.25	0.82	45.07	0	0.96
P6S30D6	5.75	36	3.61	2.22	0.64	0.25	3.29	0.56
P6S30D9	8.60	46	11.60	0.76	0.77	5.30	0.75	0.72
P10S30D18	98.01	50	43.83	0.04	0.83	N/A	N/A	N/A
P10S30D27	181.46	50	58.91	0	0.86	N/A	N/A	N/A
P10S50D18	306.12	50	16.90	0.45	0.88	N/A	N/A	N/A
P10S50D27	514.61	50	28.19	0	0.91	N/A	N/A	N/A
P10S50D10	48.70	35	0.52	4.39	0.60	0.63	6.43	0.65

carriers to collaborate in liner shipping is that they do not have enough ships to maintain weekly frequency on the routes. For instances other than in P6S30D6 and P10S50D10 test class, since carriers and subset of carriers have few ships (as compared to the available demand), in most of the cases the grand alliance offers the best possibility for maintaining the required frequency on the service routes and thus most of the instances have non-empty core. Instances in P6S30D6 and P10S50D10 test classes, are however more likely to have profitable sub-coalitions. Our experiments yield that subsets of carriers that have good synergy in the origin- destination port of their demand triplets are more likely to form sub-coalitions. In general, if sub-coalitions have higher synergies (as compared to the grand alliance) then it is less likely that the grand alliance will be formed. Also there are fewer incentives for carriers to get into the organizational and managerial complexities of big alliances.

For 6 port instances with 18 ships, the percentage of unsatisfied demand is quite high. Further the average un-satisfied demand for the instances with empty core is even higher than the average un-satisfied demand for the instances with non-empty core. Thus instances with small fleet size in which carriers find synergies among themselves to satisfy higher demand are more likely to have a stable grand alliance. Also we note from Table 2 that as the size of the network increases from 6 to 10 ports all the instances in all the test classes (except P10S50D10) have non-empty core. Instances with 10 ports that have very high demand as compared to the available fleet (un-satisfied demand is 40-60% of the total demand) are very likely to form stable grand alliance. In these instances, as there is a shortage of ships, only the grand alliance provides a global optimal schedule for the available fleet. Table 2 reflects that in fact the grand alliance schedules almost all the ships in the fleet and provides very high utilization of capacity (85-95%) on the operated routes.

Note that from Table 1 and Table2, if the core of a problem instance is non-empty, our mechanism succeeds in finding an allocation in the core in almost all (95-100%) of the instances, when the inverse problem is considered with all the subset rationality constraints.

4.4 Size and Number of Carriers

Next, we study the effect of number of carriers in an alliance. Table 3 reports results for alliances with two, three and four carriers. The first column represents the test class. To generate instances with i-carriers the number of demand pairs and ships are distributed uniformly randomly among i carriers. Thus, Table 3 reports results for alliances among carriers with similar characteristics.

Table 3: Analysis of the size of an alliance.

Test	2-Carriers		3-Carriers		4-Carriers	
Class	core	% improvement	core	% improvement	core	% improvement
P6S18D6	50	63.04	48	275.45	46	717.40
P6S18D9	50	40.73	36	167.60	37	475.61
P6S30D6	50	17.11	36	50.58	29	117.87
P6S30D9	50	20.12	46	66.17	43	115.64
P10S30D18	50	26.91	50	61.05	50	99.65
P10S30D27	50	19.68	50	53.42	48	88.13
P10S50D18	50	14.24	50	31.38	48	72.11
P10S50D27	50	15.94	50	32.36	49	69.45

The second and the third column report results for alliances with two carriers. The second column reports the number of instances (out of 50) for which an allocation in the core exists. The third column reports the average (average taken over 50 instances) percentage improvement in the total revenue generated by the alliance as compared to sum of the revenue generated by individual carriers working independently. The next two twins of columns report similar statistics for alliances with three and four carriers, respectively. For a particular instance, the percentage improvement in the revenue generated as a result of the alliance is computed as:

$$\frac{opt(N) - \sum\limits_{i \in N} opt(\{i\})}{\sum\limits_{i \in N} opt(\{i\})}.$$

Table 3 suggests that as the number of carriers increases in an alliance, the alliance tends to become more un-stable in the sense that the number of instances with a non-empty core decreases. Note that as the number of carriers increases the number of constraints that need to be satisfied to obtain an allocation in the core increases exponentially. Higher the number of constraints higher the chances that the constraints will conflict with each other and thus higher the chances that the core will be empty. With higher number of carriers in an alliance it is more likely that some subset of carriers will have better synergies for an alliance than the grand alliance. Qualitatively, as the number of carriers increase in an alliance the organizational complexity of the alliance increases and the decision making process becomes time consuming. One of the most successful alliances among liner carriers have been the Maersk-Sealand alliance which consists of only two carriers [3]. Some of the bigger alliances have organized and re-organized themselves a number of times within a short span of time. For example, the Global Alliance which was formed in 1995 among four carriers (APL, OOCL, MOL and Nedlloyd) reorganized in 1998 to form the New World Alliance (NOL/APL, MOL, HMM) after the merger of APL and NOL in 1997.

The third, fifth and the seventh column of Table 3 clearly shows that alliances can generate higher revenues as compared to the carriers operating on their own. For a particular instance, i.e. for a given set of ports, fleet size and demand pairs, as we increase the number of carriers (that is distribute the fleet and demand pairs among a larger number of carriers), the revenue that an individual carrier can generate reduces. However, the optimal solution of the grand alliance remains the same, independent of the number of carriers in the alliance. Thus, as reflected by Table 3 the the percentage increase in the revenue generated as a result of the alliance increases as the number of carriers increases.

4.5 Role and Contribution of Carriers in an Alliance

We study how participants with complementary and similar roles influence an alliance. Specifically, we study the alliance between a ship owner and a group of shippers. That is one player has all the ships and the other players have all the demand. First, we study instances (drawn from different test classes P6S18D6 - P10S50D27) with one ship owner and one shipper. This is a perfect situation for collaboration and all these instances have a non-empty core. Further, in all such instances our mechanism provides a cost structure such that the resulting payoffs to both the participants is in the core, when the inverse problem is solved with all the subset rationality constraints. Thus a stable alliance can be formed in all these instances. Next, we study problem instances with three shippers and a single ship owner. Depending on the problem instance ($P6S30D9$ etc) we found that the core is non-empty in 90%- 100% of the cases. Among the instances with non-empty core, in 95%-100% of the instances our mechanism provides a cost structure such that the resulting payoff to the participants is in the core. Comparing one ship owner and one shipper case with the one ship owner and three shippers case we conclude that in the latter case, shippers give rise to competition and instability in the grand alliance.

An interesting observation is that for $P6S30D6$ problem instances for one ship owner and three shippers case, 98% of the instances have non-empty core. Whereas, for this test class when the ships and demand pairs are distributed uniformly among four carriers only 58% of the instances have non-empty core (Table 3). For the $P6S30D6$ class the number of ships are enough to satisfy most of the demand, thus even if there are three competing shippers the alliance is stable. As the ship owner has sufficient number of ships, he has an incentive to collaborate with as many shippers as possible to increase his revenue. Similarly, though the shippers compete for capacity on the ships, in the case when the system has over-capacity they can all form a sustainable alliance with the ship owner. However this is not the case when ships and demand pairs are uniformly distributed among four carriers as many subset of carriers find synergies to form sub-coalitions. In general, for other test classes also, instances with ships and demand pairs distributed among one ship owner and three shippers are more likely to have a non-empty core as compared to four equi-sized carriers. This is simply because in the former scenario the players have higher degree of complementarity in their role. Carriers have used conferences and alliances to fix price and moderate the buying power of shippers. As a result of our experiments we conclude that it is a good strategy for shippers also to form alliances and consolidate cargo before negotiating with the carriers and ship operators. This practice is observed in the industry as giant shippers and freight-forwarders consolidate the cargo of small shippers.

5 Conclusions

In this paper we addressed various problems posed by alliance formation among carriers in transportation networks. We designed allocation mechanism for an alliance to share the benefits of the alliance in such a way that all the members are motivated to act in the best interest of the alliance. Since the revenue generated by the collaborative solution alone is not enough to guarantee the satisfaction of an individual carrier, the mechanism provides side payments to the carriers to motivate them to "play along" in the alliance. Considering our preliminary results, we believe that the suggested solution approach has the potential to help carriers form sustainable alliances.

In particular for liner shipping alliances, our experiments suggest that across all test classes, in most (more than 95%) of the instances that have non-empty core our mechanism successfully finds a cost structure such that the resulting payoff to the carriers is in the core, when the inverse problem is solved with all the rationality constraints. Assignment of ships to the service routes influences the cost incurred by a carrier (thus his payoff) and the ownership of capacity on the edges by the carriers. However our results indicate that independent of the assignment of ships to the service routes in most of the cases our mechanism successfully finds a cost structure such that the overall payoff to the carriers is in the core. Analysis of different test classes suggests that the core is empty for a very high number of instances (more than 72%) drawn from the classes in which carriers have sufficient number of ships to satisfy the available demand. Further our experiments yield that, as the number of carriers increase in an alliance, the percentage improvement in the total revenue generated by the alliance as compared to the sum of the revenue generated by individual carrier independently increases. However, it becomes harder to find a solution in the core as the number of constraints to obtain a core allocation increase exponentially with the number of carriers. Further we conclude that carriers who have complementarity in their roles, for example ship owners and freight forwarders, are more likely to form stable alliances. Note that many other factors (such as compatibility in mission, strategy, governance, culture, organization and management etc of different partners of an alliance) also contribute significantly to the success of an alliance.

In this paper we considered alliance formation among 3-4 carriers. For these alliances we considered all subset rationality constraints to find an allocation in the core. In many transportation and other logistics problems it is necessary to consider alliances with higher number of participants. Also in liner shipping, smaller alliances are collaborating to from even bigger alliances, for example Grand Alliance and The New World Alliance laid down foundations for cooperation in 2006. However considering all rationality constraints becomes prohibitively expensive as the number of carriers increase in an alliance. To extend the mechanism developed in this paper for alliances with higher number of participants, subset rationality constraints need to be added in the inverse problem in a constraint generation setting.

This paper provides a basic framework for the research that can be used for designing allocation mechanism for various network design problems. The research integrates tools from optimization, economics and mathematics to study the selfish behavior of individual players in an alliance thus providing advances in interdisciplinary work.

References

[1] R. Agarwal and O. Ergun. Ship scheduling and network design for cargo routing in liner shipping. *to appear in Transportation Science*, 2007.

[2] N. Nisan and A. Ronen. Algorithmic mechanism design. *Games and Economic Behavior*, 35:166–196, 2001.

[3] D. W. Song and P. M. Panayides. A conceptual application of cooperative game theory to liner shipping strategic alliances. *Maritime Policy and Management*, 29(3):285–301, 2002.