A Cooperative Game for Multi-Agent Collaborative Planning

Sumit Chakraborty and Asim Kumar Pal

***Abstract*—Cooperative game is an approach for a group of decision-making agents (DMAs) to reach mutually beneficial agreements where the players make agreements in order to optimize their common cost or payment. A coordination mechanism is required to achieve a rational and structured plan for a decision jointly made by independent DMAs. This paper presents a cooperative game for the problem of collaborative planning where each planning domain is controlled by a DMA. The main purpose here is to demonstrate how the cooperative game theory can be applied to the problem of multi-agent collaborative planning. It deals with cost allocation methods which is basically an optimization problem. We present several coordination mechanisms for the game based on domain planning, data exchange and compensation negotiation. These belong to two categories, namely, multistage local planning domain based coordination mechanism (LPDCM) and single stage global planning domain based coordination mechanism (GPDCM). Here DMAs wants to maintain the privacy of their strategic information. This asymmetry of information may cause increase in cost and generate inefficient solutions. The mechanisms preserve the privacy of information using secure multi-party computation concepts. This improves the quality of a plan and subsequent decision-making process of the cooperative game significantly. This work introduces a secure multi-party linear programming extended protocol (SMLPEP) by extending an existing two-party secure linear programming protocol. This work extends the Chakraborty's work on collaborative planning for supply chain [17] and [3], which in turn was based on Dudek's piece [6] of 2004. The present work looks at the collaborative planning problem of [6] from the perspective of cooperative game, focuses on the development of mechanism design for efficient cost sharing purposes for several scenarios and thus enhances the generalization capability of the problem.**

***Index Terms*—coordination game, coordination mechanism, secure multiparty linear programming protocol, supply chain.**

Sumit Chakraborty (email: surya20046@yahoo.co.in) was with Cognizant Technology Solutions India, in the rank of Assistant Manager, Business Development. He obtained his doctorate (Fellowship) in Management Information Systems in 2007. He has worked on problems in supply chain, negotiation, pricing and auction, and privacy preserving computation.

Asim Kumar Pal (email: asim@iimcal.ac.in), the corresponding author, is Professor, Management Information Systems Group, Indian Institute of Management Calcutta, Joka, Diamond Harbour Road, Kolkata, India – 700104, www.iimcal.ac.in, Tel: +919830637252 / +919748533525 (cell), +913324678300 / +913324662945, Fax: +913324678062, +913324678307.

I. INTRODUCTION

A *cooperative game* (***copg***) (also often referred as *coordination game*) is a common approach for a group of *participants* or *decision-making agents* (***DMA***s) or simply *agents* to reach mutually beneficial agreements. This is an important conflict management and group decision-making approach for making a joint decision by a set of agents. The agents exchange information in the form of offers, counter-offers and arguments and search for a fair consensus [7], [8]. In *distributed artificial intelligence*, a coordination game is the study of preference aggregation that leads to better outcomes for all players in spite of selfish strategic behavior of the agents [11]. Mechanism design is the study of preference aggregation protocols that work well in the face of self-interested agents [1]. An agent may gain incentives if it reports misleading preferences to the preference aggregator for selecting a joint outcome. The basic objective here is to study how the individual preferences of the agents can be aggregated towards a *rational choice* while maintaining *privacy* of the participants [15]. The *mechanism designer* should ensure that individual interests of the agents are best served by their rational behavior. An efficient mechanism allocates payments to the agents fairly. The payments should be carefully selected to motivate all the agents to act rationally. *Coordination mechanisms* (***codm***) have been used profitably in various applications such as combinatorial auction, reverse auction, task allocation, strategic sourcing, compensation negotiation for human resources management and recommender systems [1], [2], [6], [12], [14]. [9] introduced the concept of *algorithmic mechanism design* and [10] delved into different issues and applications of this concept.

In the present work, a copg has been studied for the problem of *collaborative planning* (***colp***). Efficient codm's have been attempted to achieve a rational and structured plan. In this game the agents do not want to share private information with a fear that such type of information may be exposed to their competitors. This asymmetry of information may result in expensive and inefficient solutions. The proposed codm's preserve the privacy of strategic information of the agents with the help of *secure multiparty computation* (***smc***), a cryptographic technique. This improves the quality of a plan and subsequent decision-making process in the coordination game significantly [3]. The proposed copg starts with the planning phase. The

ISBN: 978-988-18210-5-8
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

agents initiate several joint activities by specifying their objectives, preferences, aspiration and reservation levels and the communication mode. They set various agenda such as codm's, the timing of exchanges, deadlines, priorities and constraints. They exchange offers and arguments; learn about the limitations of other agents; identify the areas of agreement and disagreement and modify negotiation strategies. Next, they develop joint proposals by relaxing their individual limitations and reach an agreement. Finally, the agents analyze the compromise proposals at the conclusion phase and may explore the scope of possible improvements. The game is analyzed here from three perspectives mainly, rationality of the agents, optimization of the process and preservation of privacy of information among the agents.

The present work had its root in [6] which looked at various scenarios of colp based supply chains. The latter had developed *linear programming* (lp) models for the same and also considered some issues of privacy. [17] had extended some of these models, improved upon it particularly from the point of view of security concerns and made some of these algorithms more efficient by minimizing the requirement of negotiations. It also suggested local domain based and global domain based plans for achieving the global minimum in total cost and introduced the idea of mechanism designs. [3] investigated some of these cases further. The present work develops a cooperative game to broaden the perspective of the colp and looks closely at the issue of cooperative mechanism design for several scenarios, particularly multiple stage *LPDCM* (*local planning domain based coordination mechanism*) and single stage *GPDCM* (*global planning domain based coordination mechanism*). This work also gives a *secure multi-party extended linear programming protocol SMPELP* by extending and also expanding upon Du's work [5]. It applies SMPELP to GPDPM.

[16] deals with the algorithmic cooperative game theory and develops a variety of games, their cores and nucleolus (see below) considering efficiency and stability of these games. This however has not looked at the colp problem which is the main topic of the present work. Further, the former does not particularly give any mechanism design and also does not treat the issue of privacy and security which the current work emphasizes.

The paper is organized as follows. Section II presents an overview of cooperative game; III defines the cooperative game for collaborative planning; IV describes local planning domain based coordination mechanisms: SBSS, SBMS and MBSS; V presents global planning domain based coordination mechanism, while section VI discusses secure lp protocol SMPELP and smc issues in general. Finally, section VII concludes the paper highlighting some applications in operations research.

II. COOPERATIVE GAME

Game theory is concerned with a complex decision making process in which two or more players who may have conflicting interests interact. Each of these players tries to optimize its own objective function. A game can be classified as *cooperative game* (copg) or a *non-cooperative game*. In a copg the players make agreements in order to minimize their common cost or to maximize their common pay-off. This is not possible in a non-cooperative game. This section defines the solution concepts of the cooperative game theory, core, shapely value, bargaining set, nucleolus and kernel [4],[10], [16].

Cooperative Game: A *cooperative game* is a game where a group of players enforce a *cooperative behavior*. A copg denoted (N, u) is defined by N = {1,2,…,N} denoting a set of players and u a real valued characteristic function. A subset $S \subseteq N$ is called a *coalition*, where N is called the *grand coalition*.

Imputation: The basic objective of a copg is to find an acceptable distribution of cost among the agents. Imputations are efficient and individually rational distribution. An *imputation* $\boldsymbol{y} = (y_1, \ldots, y_N)$ is a vector in R^N such that cost y_i is allocated to player i and $y(N)=\sum_{i=1}^{N} y_i$. An imputation is a *pre-imputation* that satisfies the requirement $y_i \leq c(i)$. Let $c(N)=\sum_{i=1}^{N} c(i)$, where c is the *characteristic function* in the game (N, c). The excess of a coalition S w.r.t. cost allocation vector $\boldsymbol{y}$ is $e(S,\boldsymbol{y}) = c(S) - y(S)$. (N,c) is a *monotone* game if c is monotone, i.e. $c(S) \leq c(T)$ for $S \subset T \subset N$. (N, c) is *proper* if the characteristic function is sub-additive, i.e. $c(S)+C(T) \geq C(S \cup T)$ for all $S, T \subset N$, $S \cap T = \phi$. In a proper copg it is always cost effective to form large coalitions.

Solution Concept: A *solution concept* of a copg must satisfy a number of properties. The total cost allocated to the players must be equal to the total cost of the game, i.e. y(N) = c(N). This is also known as *pareto-optimality*. The cost allocated to a player should not be higher than the cost the player would have to incur if he acts individually without joining others. This property is known as *individual rationality*. The allocation of costs should be symmetric. The solution should satisfy the property of monotonocity. If the overall cost increases, the allocation of the player should increase accordingly.

Core: If the players of a game work jointly then the critical issue is the allocation of profit among those players. If one or more players consider that the proposed allocation is disadvantageous to them they can decide to leave. The *core* is the most significant fair solution concept of a copg. In a core solution there is no incentive for any player to leave the grand coalition, the core solutions are stable. In a game (N, c), the *core* is defined as those imputations y that satisfy $y(S) \leq c(S)$, $S \subseteq N$ and y(N) = c(N). The total cost allocated to the players in a copg should not exceed the cost of a system dedicated to the coalition and should satisfy group and individual rationality constraints. The efficiency constraint implies that the total cost of the game is to be equitably distributed among the players.

Shapely Value: The *shapley value* is the unique payoff vector that is symmetric, additive and efficient. This also

ISBN: 978-988-18210-5-8
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

satisfies anonymity and assigns zero payoffs to dummy players. The order of the players does not affect the costs allocated to the players.

Bargaining Set: A set of objections and counter objections. An imputation y belongs to a *bargaining set* M(c) of the game if for any objection of a player against another with respect to y there exists a counter objection.

Nucleolus: The *nucleolus* indicates those imputations that minimize the maximum discontent of any player of a copg.

Kernel: The *kernel* of a game indicates the imputations for which no player outweighs another player.

There are many illustrations of cooperative combinatorial games such as assignment, permutation, sequencing, travelling salesman, delivery, packing and covering, matching, routing games, minimal cost spanning tree, facility location and network flows games [4].

The main purpose of this paper is to demonstrate how the cooperative game theory can be applied to the problem of *multi-agent collaborative planning*. It deals with cost allocation methods involving combinatorial optimization. Several codm's have been proposed based on domain planning, data exchange and compensation negotiation. These mechanisms are categorized as: a) *Multi-stage Local Planning Domain based Coordination Mechanism* (*LPDCM*) : SBSS (Single B Single S), SBMS (Single B and Multiple S) and MBSS (Multiple B Single S), and b) *Single stage Global Planning Domain Based Coordination Mechanism* (*GPDCM*).

III. COLLABORATIVE PLANNING GAME

Two Classes – B and S Agents: This section proposes a copg for the problem of colp where each participant has a well-defined objective function and a set of constraints that represent its preferences over the possible outputs of the game. These participants act rationally to optimize their objective functions and follow the codm's correctly. For the purpose of explaining the present copg, the codm for the single 'B' and single 'S' (SBSS) is being used here. There are two different classes of agents involved in the coordination game: *class B* and *class S*. As such B and S do not signify anything or any particular behaviours. Except that the B class of agents will have similar interests. The same holds for the S class. But the B class and the S class agents will have opposite (or complementary) types of behaviour or interests. For example, B could represent the Buyer class and S the Supplier class in two successive tiers of a *supply chain* (***SC***). The most important point is that the B agents can collaborate among themselves (i.e. form coalitions), so also the S agents. B agents can either singly or as a coalition interact with the individual S agents or a subset of or all of the S agents in a coalition. For different purposes (operations or computations) the agents will act singly or jointly in a coalition of agents belonging to the same class. They can even act in a grand coalition of all the agents of both classes for yet another computation (e.g. GPDCM). The specific codm will specify the sequence of individual and joint computations among the partners involved.

In case of the bi-party negotiation (SBSS) only two agents are involved in the mechanism – one B agent and one S agent. Similarly, for other negotiation situations varying number of B and S agents are involved, such as single B agent and multiple S agents in SBMS, multiple B and multiple S agents in MBSS and multiple B agents and multiple S agents in MBMS mechanisms.

Planning domains: Local and Global: The *local planning domain* (***lpd***) of B is defined through the constrained optimization problem:

$$\min\, (o^B)^T \mathbf{x}^B, \text{ s.t. } M^B \mathbf{x}^B \le b^B$$

where $\mathbf{x}^B$, o^B, b^B and M^B are the decision variable vector, the cost vector, the constraint lower bound vector and the constraint matrix respectively of B (T: matrix transpose operation). Similarly, the lpd of S is:

$$\min\, (o^S)^T \mathbf{x}^S, \text{ s.t. } M^S \mathbf{x}^S \le b^S.$$

Combining these two one obtains the joint optimization problem:

$$\min\, o^T \mathbf{x}, \text{ s.t. } M\mathbf{x} \le b$$

$$\text{where } \mathbf{x}=\mathbf{x}^B \oplus \mathbf{x}^S,\ o=o^B \oplus o^S,\ M=M^B \oplus M^S \text{ and } b=b^B \oplus b^S$$

for the entire system referred as the *global planning domain* (***gpd***). Here, **x**, o, M and b represent the decision variable set, the cost vector (i.e. objective function), the constraint matrix and the constraint upper bound vector for the global plan. The *combination operator* $\oplus$ has to be defined appropriately. The basic problem is to develop a codm which ensures that consistent plans are generated by reducing planning cost, total negotiation time, information disclosure to any agent and asymmetry of information disclosure to the parties.

Plan: In the case of bi-party negotiation, the B agent gives its plan to the S agent. The latter executes the plan for the former. The negotiation starts with B bidding a plan P to S. S evaluates P and counter bids an alternative plan P'. B in turn evaluates P' and counter proposes yet another P", and so on. Finally, if the negotiation ends successfully, S executes the commonly accepted agreed plan. The negotiation for a plan consists of successive bidding cycles. In each bidding round, a plan P is bid by either B or S. A successful negotiation process consists of a starting plan (initial plan) followed by a series of compromise plans which culminates in a finally accepted plan (see below):

Initial Plan: B proposes P_0 which is optimal for B.

Compromise Plan, Plan Evaluation and Compensation Negotiation: Subsequently compromise plans P_i, $i \geq 1$, are (alternatively) bid by B and S. P_i^B (P_i^S) denotes a plan bid by B (S). A *compromise plan* P_i is generated based on the evaluation of the previous plan P_{i-1} bid by the opponent. This *plan evaluation* is done both in terms of costs and fulfillment of the plan objective. Further, each compromise plan also involves a compensation amount to be given to B by S, which is settled through one or more internal rounds of compensation negotiations. Further, it is also quite possible for an agent to bid a plan P_i against an earlier plan P_j, $j < i-1$,

ISBN: 978-988-18210-5-8
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

of the opponent based on further information revealed during the subsequent negotiation process (i.e. after P_j was bid). But this fact can optionally be revealed by the bidder to make it possibly easier for the opponent to make a decision.

Final Plan: P_f is the *final plan* that is agreed by both the parties irrespective of who offered it.

Plan Cost: For any plan P the cost components of B and S are denoted by $C^B(P)$ and $C^S(P)$ respectively. These are private to the agents and will not be disclosed to the opponent, i.e. what is revealed in the negotiation process is the proposal for B and the proposal for S without any cost implications. The total cost for a plan P, $C(P)= C^B(P)+C^S(P)$, is also not revealed to either agent.

Cost Effects – Local and Global: Since P_0 is optimal for B, $C^B(P_0) < C^B(P_i)$ for all $i \geq 1$, i.e. the *cost effect* for B (S) for P_i, $\Delta C^B(P_i) = C^B(P_i) - C^B(P_0)$ ($\Delta C^S(P_i) = C^S(P_0) - C^S(P_i)$). Note that both the cost effects are positive. Cost effect of a B agent or an S agent is also referred as *local cost effect*, whereas the *global cost effect* or total cost effect for P_i is sum of the local cost effects of all the agents. This is because the objective of the coordination process is to decrease the total cost, not the individual costs. However, B is entitled to ask for suitable compensation from S to compensate for the additional cost it has to incur in P_i. Individual cost effects are treated as private information.

Compensation Amount and Cost Sharing: B will always ask for a compensation amount, which is at least the cost effect. The compensation negotiation has basically two purposes: i) to determine whether the current plan P_i is a feasible one, i.e. whether total cost of P_i has increased over the previous plan P_{i-1} (or any other past plan P_j, $j < i-1$); and ii) to determine the amount of savings in costs to be shared between B and S. This is known as *savings sharing or cost sharing*.

Cost Implication (CI): CI for B for a plan P is the cost component of P ($CI^B(P)$) minus the compensation settled (Comp(P)). Similarly, for S. Note, the total of cost implications for B and S is same as the total cost for the plan, C(P). Thus,

$CI^B(P) = C^B(P) - Comp(P)$,
$CI^S(P) = C^S(P) + Comp(P)$, and
$C(P) = C^B(P) + C^S(P) = CI^B(P) + CI^S(P)$.

Compensation Negotiation, Rational Behaviors of the agents and Privacy Preservation: Compensation negotiations are realistic. The agents behave rationally. If the total cost reduces, compensation will always be settled such that no agent loses compared to the previous round. In other words the cost implications for both parties improve. Further, if the compensation negotiation fails, it only means that the total cost for the current bid is more than that for the previous bid. When the negotiation ends successfully in the final plan P_f, the total cost achieved is nothing but $C(P_f)$. The total savings through the negotiation will be $C(P_0) - C(P_f) > 0$, which is apportioned as $Comp(P_f)$ for B and $C(P_0) - C(P_f) - Comp(P_f)$ for S. Both B and S are assumed to be rational in exchange of truthful communication and are interested in reducing total plan cost. If none of the parties respond then there will be a deadlock. That means that neither B nor S is interested in cost reduction, which violates our assumption. Note, privacy preservation of individual agents is an important concern for this copg. For this purpose the cost effects are compared privately. Because the cost effect amounts are kept secret from the respective opponents the compensation negotiation becomes relevant and the parties feel encouraged to participate in this negotiation.

Stopping Criteria: Stopping the game is possible on various counts: total time taken, total number of plan bidding rounds, number of successive failed biddings, satisfaction of both parties, etc. If any agent withdraws prematurely the game ends unsuccessfully.

IV. COORDINATION MECHANISM: LPDCM

An *algorithmic coordination mechanism* is composed of various types of elements – a *finite set of agents*; a *finite set of inputs* as possessed by each agent; a *finite set of outcomes* as defined by *output function*; an *utility function* what each agent aims to optimize; *payments*, i.e. compensation and bonus; *strategies*: an agent selects a strategy from a family of strategies defined by the mechanism; a *dominant strategy* that maximizes the utility of an agent for all possible strategies of other agents involved in the mechanism; *revelation principle* by which the agents report their strategies and truthful implementation [9], [10]. A mechanism is *truthful* if all the agents report their strategy types. *Truth telling* may be a *dominant strategy*. A mechanism is *strongly truthful* if truth telling is the only dominant strategy.

This section presents LPDCM for the proposed coordination game. Optimization of a *collaborative planning process* can happen in two principal ways. A gpd is obtained by combining the lpd's which includes optimizing functions and constraints of all the agents involved and then optimize the global plan in this domain based on all the domain variables and constraints. A lpd is obtained for each agent and local plans optimized in these domains given static or dynamic set of constraints. The constraints become dynamic if it is not necessary that all the constraints are previously known to each agent a priori. Actually, often, constraints of one agent become dependent on the constraints of other agents. Finally, global plans which are intended to be optimal for all the parties together are obtained usually through an iterative process of optimizing the plans locally.

A. SBSS Coordination Mechanism

1. B bids optimal plan P_0 to S. Set i=0. Reference plan = P_0.
2. Repeat until the stopping criteria is satisfied:
   a. Set i = i + 1;
   b. B counter bids P_i^B to S, or S bids P_i^S to B.
   c. B and S compute local cost effects $\Delta C^B(P_i)$ and $\Delta C^S(P_i)$.
   d. B and S privately compare $\Delta C^B(P_i)$ and $\Delta C^S(P_i)$ and sets the reference plan to P_i if $\Delta C^S(P_i) > \Delta C^B(P_i)$.
3. If both parties agree, output plan $P_f = P_i$. B and S jointly settle the compensation for B through negotiation, based on relative cost effects for the final plan P_f.

ISBN: 978-988-18210-5-8
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

B. SBMS Coordination Mechanism

1. B bids its optimal plan P_0 by splitting into subplans P_{01}, …, P_{0m} (s.t. $P_0=P_{01}\oplus \dots \oplus P_{0m}$) for m S agents. Set i=0. Set, Reference plan = P_0. ($\oplus$ is the plan combination operator.)
2. Repeat until the stopping criteria is satisfied:
   a. Set i = i + 1;
   b. B agent's round: For each j, j=1,…,m, B counter bids P_{ij}^{B} to S_j in parallel.
   S agents' round: For each j, j=1,…,m, S_j counter bids P_{ij}^{Sj} to B. Thus, the combined delivery plan P_i received from m S-agents is $P_i = P_{i1} \oplus \dots \oplus P_{im}$.
   c. B and S agents (S_j,j=1,…,m) compute local cost effects $\Delta C^B(P_i)$ and $\Delta C^S(P_i)$.
   d. The leader of the S agents (say, S_l) and B privately compare $\Delta C^S(P_i) = \sum_{j=1}^{m} \Delta C^{Sj}(P_i)$ and $\Delta C^B(P_i)$ and sets the reference plan to P_i if $\Delta C^S(P_i) > \Delta C^B(P_i)$ (The sum and comparison will be computed privately.).
3. If all the parties agree, the output is the final plan $P_f = P_i$. B and S agents jointly settle the compensation to be given to the losing parties (The agreement will be reached and negotiations conducted privately.).

C. MBSS Coordination Mechanism

Let an S agent be involved in negotiation with m B agents. One of the B agents plays the role of the leader and interacts with the S agent. There can be two types of MBSS mechanisms. If each B agent communicates its plan to S individually, MBSS mechanism will be similar to SBMS. For a particular bidding round S may treat the plans of B agents simultaneously or according to different priorities. The second scenario will be a multi-stage mechanism. If the leader of B agents conducts a negotiation among the B agents and generates a combined plan and interacts with S on the basis of this combined plan, MBSS mechanism will converge into SBSS mechanism.

Theorem I: SBSS, SBMS and MBSS codm's preserve the privacy of cost and cost effects of B and S under the assumptions of relevant smc protocols.

(For brevity we are putting the arguments in a combined fashion.) The sole B agent or any of the multiple B agents (singly or collectively) does not disclose its cost or cost effect to S (i.e. either to the single S agent or to any of the multiple S agents singly or collectively), as the case may be. The converse is also true for S. Then B and S agent(s), singly or jointly, as the case may be, evaluate the relative cost effectiveness of any two successive plans – one proposed by the sole B agent (or one of the B agents, or one set of plans by multiple B agents combined into one plan – as the case may be), and the other by S agent(s) (locally single, or locally combined into a single one as required). Note, here the aggregation was done only for computing the total cost or cost effect, when required. The smc is applied for secure summation, privacy preserving comparison and consensus building. Each of them finds out local cost effects for a plan and compares that with that of the other agent privately. Here, only the relative cost effectiveness (i.e. positive or negative sign of the cost effect) of two successive plans is disclosed to both parties but none of B and S agents can get any idea of individual cost effects. Only the finally agreed plan which is targeted to achieve the maximum cost saving is disclosed. The privacy holds under the assumptions that the relevant secure and privacy preserving cryptographic smc protocols are in place.

For the issue of convergence, for example, in step 2(d) of SBSS, if $\Delta C^S(P_i) > \Delta C^B(P_i)$, the plan P_i becomes the reference plan for next iteration (i.e. P_i has improved in the total cost over P_{i-1}). Thus to show, $C(P_i)<C(P_{i-1})$, i.e. $C^B(P_i)+C^S(P_i)<C^B(P_{i-1})+C^S(P_{i-1})$ i.e. $\Delta C^S(P_i)>\Delta C^B(P_i)$, which is given. Thus, the algorithm accepts a counter proposal only when the corresponding plan is better than the previous plan. Thus, the algorithm basically ensures a sequence of plans having monotonically reducing cost (by ignoring the expensive plans which are generated and then not considered further). The finally accepted plan has the minimum cost among the plans generated by both B and S.

There is however no guarantee of achieving the globally minimum cost plan, as the negotiation process does not necessarily converge to the global optimum solution. The negotiation, which is being done based on the local information of an agent or a subset of B or S agents, can not guarantee the global optimum which will require total information. Only repeated application based on progressive computation can achieve a mutually acceptable solution which hopefully is one of the better local optima. Finding the global optimum would as such be a combinatorially explosive problem, even in the presence of total information. It will be a lp problem for the current case, where we have assumed a linear model. Thus, the following result.

Theorem II: SBSS, SBMS and MBSS codm's converge to a plan P_f which is only locally optimum w.r.t. the total cost.

In these codm's, the agents select their dominant strategies to optimize individual utilities. The agents act rationally and reveal their strategies truthfully. These mechanisms are strongly truthful since truth telling is the dominant strategy. B and S agents go through a series of plans. If the stopping criteria is satisfied, the B agent(s) start negotiation with the S agent(s) on the basis of the plan resulting in possibly maximum cost saving and settles the claim for compensation. The proposed codm requires only one round of negotiation for settling the compensation claim and this corresponds to the plan of minimum cost. The trading agents try to follow the solution concepts as discussed in section II for fair allocation of the cost savings among themselves. The stopping criteria is used to decide whether or not to continue the iterative negotiation process based on the current and previous best outcome detected so far. The improvement in the total cost is an important criterion in this connection. Another stopping criterion can be time passed, i.e. deadline of planning. Both the B and S agents get the information

ISBN: 978-988-18210-5-8
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

regarding relative cost-effectiveness of various plans. This leads to the following result.

Theorem III: SBSS, SBMS and MBSS are efficient mechanisms in terms of fair cost allocation.

The computation and communication costs depend on the number of plans generated. Each plan generation requires computation for estimation of local cost effects of B and S and private comparison of the cost effects. Local cost effect is estimated by solving an optimization problem. For each plan generated, B and S solve Yao's millionaire problem [13] in order to compare local cost effects. This is an expensive smc protocol requiring a lot of communication between B and S. There is however no need of a third-party mediator.

V. COORDINATION MECHANISM: GPDCM

Global planning domain (gpd) is basically a combination of lpd's which are distributed among the agents of both B and S classes and are private to them. The coordination game comprises two or more lpd's, one domain for each agent. For the linear case, it is assumed that linear models are useful to generate plans within individual planning domains. Each individual planning domain has its own objective function and constraints involving its own set of variables. The objective function for a planning domain mainly refers to the cost components for the respective domain. The total cost is the sum of costs for all the domains. Thus the objective functions and constraints are distributed across the agents. This is basically a lp problem of finding the global optimum plan related to the minimum cost subject to linear constraints on the decision variables.

The traditional lp solution methodology such as the simplex method assumes centralized computation wherein the centralized solver knows all objective functions and constraints. But, this situation is not valid for distributed computing and particularly for smc scenario. The objective functions and constraints are distributed among the respective agent(s) as private information. Therefore the usual solution methodology such as simplex method cannot be used directly to solve the problem of colp. In this case the agents require jointly construct a combined matrix of constraints and the combined objective function, which refers to the gpd. Here we have described a gpd based codm which is uniformly applicable for different multi-agent scenarios (SBSS, SBMS, MBSS). The scheme generates global optimum plan related to the minimum total cost and that solution should be acceptable to all the parties involved in collaborative planning.

1. The agents jointly run SMPLPEP (which constructs the global plan as an lp and solves it to finds the globally optimum solution in a secure manner) (See VI.). The agents then accept the solution without any claim for any compensation.
2. Each agent finds only its own contribution to the total cost.

The mechanism preserves the privacy of objective function and constraints of each agent involved in the collaborative planning. There is no disclosure of any agent (of B or S) to any other agent because of the application of secure sum protocol. The question of cost effects do not arise because the optimization is now a one step process (single stage). And the privacy of individual components in the constraint matrix and cost vector is taken care of by the secure lp protocol. The mechanism generates a gpd by combining the lpd's of all the agents involved and then optimize the global plan in this domain based on all the domain variables and constraints. But, the scheme is currently suitable only for the linear model. The cost of computation depends on the number of decision-making agents and the complexity of optimization problem.

A. LPDCM vs. GPDCM

GPDCM is naturally more intensive computationally as well as communication wise compared to LPDCM. This is mainly because much bigger matrices and vectors are required to be processed in GPDCM. Further, these matrices and vectors involved in matrix-vector product protocols require a lot of communication to maintain privacy [5]. Whereas in LPDCM most of the computations are held locally, only the plans are to be exchanged securely (mainly). On the other hand GPDCM achieves the global minimum cost without requiring any negotiations, but LPDCM achieves only a locally optimum cost yet requiring negotiations which may also be time consuming. Further, the process of giving counter plans in LPDCM during the main iterations need not be easily automated, in which case LPDCM could become quite inefficient in terms of both speed and quality. However, if the model is non-linear GPDCM in its current form will fail, and LPDCM can still work provided the players have adequate domain knowledge to come out with good feasible plans.

Nucleolus: The basic objective of the coordination mechanisms is to explore the imputation that minimizes the maximum discontent of the players of the game. This is only possible if the players of the cooperative game act rationally and share the correct information cooperatively maintaining privacy at desired level. LPDCM tries to reduce the discontent of the players through a bargaining set – a set of objections and counter-objections. This is a multi-stage coordination mechanism. GPDCM does not consider any bargaining set; this is a single stage mechanism. The major challenge of GPDCM is to define combinatorial optimization problem correctly by combining the rational business objectives and constraints of each decision making agent.

Core: The players of a game work jointly, then the critical issue is the allocation of profit among those players. If one or more players consider that the proposed allocation of profit is disadvantageous to them they can decide to leave. The *core* is the most significant fair solution concept of copg. In a core solution there is no incentive for any player to leave the grand coalition, the core solutions are stable. LPDCM tries to explore the core solution. It though can not ensure the global optimal solution. Therefore, the chance of getting the core solution is less. LPDCM is basically an approximation algorithm. On the other side, the chance of getting stable core solution, i.e. global optimal solution, is more in case of GPDCM.

VI. SECURE MULTI-PARTY LINEAR PROGRAMMING EXTENDED PROTOCOL (SMPLPEP)

In the proposed copg two or more agents want to conduct a computation based on their private inputs but none of them wants to share its proprietary data set to others. The objective of smc is to compute with each agent's private input such that in the end only the outputs (and implied knowledge thereof) are known to the respective target agents (as per design). The agents are semi-honest in that they are free to use intermediate results. Here we are not discussing standard smc protocols such as secure sum and private comparison. We are also skipping protocols like Matrix Vector product which can be obtained by extending Du's work [5]. We are proposing here a secure multi-party extended lp protocol which has been used in GPDCM.

Du [5] solved secure two party cooperative lp where the two parties keep their own constraint matrix (M_i) and constraint vector (v_i) private to themselves [5]. Du did not consider the privacy of the objective function and the solutions of the decision variables to the respective parties (agents). We have extended Du's method in a few ways: (i) m parties, $m \geq 2$ (ii) keep the respective part of optimization function o_i private to each agent i and (iii) keep the solutions of decision variables private to the respective agents (excluding common decision variables). But, the main integrity in Du's method comes from the following result:

$y^* = \text{argmax}_y \; o^T y$ subject to $My \leq v$ --- (1)

$z^* = \text{argmax}_z \; o^T Hz$ subject to $GMHz \leq Gv$ --- (2)

where G and H are random matrices having all positive elements and H is nonsingular. Then $y^* = Hz^*$ holds.
We give below our m-agent privacy-preserving lp.

a) There are m decision makers or agents 1, …, m who wish to solve an lp jointly, yet maintaining privacy regarding their own part of the constraints and the objective function. The agent i has the local constraint matrix M_i, constraint vector v_i and its part of the optimization vector o_i. The global constraint matrix, constraint vector and the optimization vector are M, v and o are obtained as follows (* is the combining operator):

$M = M_1 * \ldots * M_m$, $v = v_1 * \ldots * v_m$, and $o = o_1 * \ldots * o_m$.

Note, the solution vector of the global optimization problem $\max_y o^T y$ s.t. $My \leq v$ will also be appropriately split, i.e. $x = x_1 * \ldots * x_m$, s..t. x_i is the relevant local solution vector for i. Mathematically it will be easier to handle if there is a simple transformation of the local constants M_i, v_i and o_i to M'_i, v'_i and o'_i, s.t. $M = M'_1 + \ldots + M'_m$, $v = v'_1 + \ldots + v'_m$ and $o = o'_1 + \ldots + o'_m$.

b) Alice and Bob exchange relevant components of their shares and sums up to get the relevant component of the final solution x. Alice and Bob send to each agent i the components of the solution of its decision variables. Each i combines these components and gets the solution of its decision variables.

---

1. Let two out of m parties be selected at random be called Alice and Bob. Renumber the agents: Bob 1, Alice 2, others 3,…, m. The agent $1 \leq i \leq m$ has inputs M'_i, v'_i and o'_i.
2. Bob splits his constants into (m-1) random parts one for each of the remaining agents (including Alice) as follows: $M'_1 = M'_{12} + \ldots + M'_{1m}$, $v'_1 = v'_{12} + \ldots + v'_{1m}$, $o'_1 = o'_{12} + \ldots + o'_{1m}$.
3. For each agent i = 2,…,m (call it Sally)
   a) Bob generates G and H.
   b) Sally and Bob share $o'^T_i H$ using the matrix vector product protocol. (Note, Bob already has $o'^T_i H$).
   c) Sally computes ($o'^T_{1i} H + o'^T_i H$) and sends it to Alice. (N.B.: Bob does not learn $o'^T_i H$ and Sally does not learn $o'^T_{1i} H$; this is not strictly required for m>2).
   d) Sally computes $G(M'_{1i} + M'_i)H$ and $G(v'_{1i} + v'_i)$ and sends these to Alice. (N.B.: Sally does not learn G, H, M'_{1i}, v'_{1i}, $GM'_{1i}H$, Gv'_{1i}; Bob does not learn GM'_iH, Gv'_i).
4. a) Alice constructs the global optimization problem (with the noisy matrices G & H): $O' = \sum_{i=2}^{m} (o'_{1i} + o'_i)$

   $GM'H = G \sum_{i=2}^{m} (M'_{1i} + M'_i)H$, $Gv' = G \sum_{i=2}^{m} (v'_{1i} + v'_i)H$,

   (N.B: Alice does not learn G, H and any of the individual constants M'_1, v'_1 or o'_1.)
   b) Alice solves the optimization problem (2) and finds z*.
5. a) Bob shares $x = Hz^*$ with Alice using matrix-vector product protocol which is the solution of the global optimization problem (1).
   b) Alice and Bob exchange relevant components of their shares and sums up to get the relevant component of the final solution x. Alice and Bob send to each agent i the components of the solution of its decision variables. Each i combines these components and gets the solution of its decision variables.

Theorem IV: SMPLPEP preserves the privacy of objective functions and constraints of the decision-making agents.

The secure lp treats the objective function and constraints of each agent as private inputs. Sally computes ($o'^T_{1i} H + o'^T_i H$) privately. Bob does not learn $o'^T_i H$ and Sally does not learn $o'^T_{1i} H$. Thus, the protocol preserves the privacy of objective function of each agent. Sally also computes $G(M'_{1i} + M'_i)H$ and $G(v'_{1i} + v'_i)$ privately such that Sally does not learn G, H, M'_{1i}, v'_{1i}, $GM'_{1i}H$, Gv'_{1i}; Bob does not learn GM'_iH, Gv'_i. In 4th step of the protocol, Alice learns GM'H, Gv' and o' but does not learn G, H and any of the individual

ISBN: 978-988-18210-5-8
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

constants M'_1, v'1 or o'_1. In step 5, Alice holds a vector z* and Bob holds a matrix H. The objective is to compute Hz* using privacy preserving matrix-vector product protocol. Here, Alice cannot get any information of H and Bob cannot know the content of z*. Alice and Bob exchange each other's components of final solutions so that any agent cannot know the solution of the decision variables of other agent except the solution of common decision variables.

VI. CONCLUSION and APPLICATION

Multi-agent negotiation based colp is important for efficient SC management in e-market. The proposed game and the codm's can be applied to *collaborative planning, forecasting and replenishments* (*CPFR*) in retail and manufacturing business. Dudek worked on collaborative SC planning by setting up an lp framework [6]. This work was extended by exploring various scenarios of multi-party negotiation among the trading agents [3]. The sharing of correct strategic information is important for efficient coordination of operational processes across a SC. But the buying and selling agents of a SC are often reluctant to disclose sensitive strategic information since the information can either be used by the SC agents or can be revealed to their competitors. Lack of information exchange gives rise to information asymmetry and causes problems related to capacity utilization, inventory control, transportation, distribution and customer service. CPFR is a strategic tool for comprehensive value chain management of an organization. This is an initiative among all the stakeholders of the SC in order to improve their relationship through jointly managed planning, process and shared information. The ultimate goal is to improve a firm's position in the competitive market and the optimization of its own value chain in terms of optimal inventory, improved sales, higher precision of forecast, reduced cost and improved reaction time to customer demands.

Future vision of the present works: *A supply chain can become a revenue chain.* Different types of algorithmic codm's can be developed through different types of *business intelligence moves* which can enhance profit and revenue in SC coordination. Moreover, an efficient negotiation support system can automate the proposed copg of colp.

REFERENCE

[1] V. Conitzer, T. Sandholm and C. Boutilier, 2007. Automated design of multistage mechanisms. In Proceedings of IJCAI.

[2] V. Conitzer, and T. Sandholm, 2003. Applications of automated mechanism design. In Proceedings of UAI Bayesian Applications Workshops.

[3] S. Chakraborty and A. K. Pal, 2007. Privacy-preserving collaborative business process management, CBP Workshop, International Conference on Business Process Management, Brisbane, LNCS 4928, Springer Verlag, pp. 306-315,.

[4] A. Chinchuluun, P. M. Pardalos, A. Migdalas, and L. Pitsoulis (Ed.), 2008. Pareto optimality Game Theory and Equilibria. Springer.

[5] W. Du, 2001, A study of several specific secure two-party computation problems, Ph.D. Thesis, Purdue University.

[6] G. Dudek, 2004,2009. Collaborative planning in supply chain negotiation based approach, Editions 1 and 2, Springer-Verlag.

[7] S. S. Fatima, M. Wooldridge and N. R. Jennings, 2004, An agenda based framework for multi-issue negotiation, Artificial Intelligence, 152, pp. 1-45.

[8] S. Kraus, 2001, Strategic negotiation in multiagent environments, MIT Press, Cambridge, MA.

[9] N. Nisan and A. Ronen, 1999. Algorithmic mechanism design, In Proceedings of 31st Annual ACM Symposium on Theory of Computing, pp. 129-140.

[10] N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani, 2007, Algorithmic game theory, Cambridge University Press.

[11] S. Russel, and P. Norvig, 2008. Artificial intelligence: a modern approach, 2nd Edition, Prentice Hall.

[12] M. Silaghi, 2002. An algorithm applicable to clearing combinatorial exchanges. (Technical report no. CS-2002-14) Florida Institute of Technology, Melbourne, Florida.

[13] A. C. Yao, Protocols for secure computations, In Proceedings of IEEE Annual Symposium on Foundations of Computer Science, pp. 160-164.

[14] M. Yokoo, K. Suzuki, and K. Hirayama, 2005, Secure distributed constraint satisfaction: reaching agreement without revealing private information, Artificial Intelligence, 161, pp. 229-245.

[15] G. Weiss, 1999, Multiagent systems, MIT Press, Cambridge, Massachusetts.

[16] X. Deng and Q. Fang. 2008, Algorithmic Cooperative Game Theory.

[17] S. Chakraborty, "A study of several privacy-preserving multi-party negotiation problems with applications to supply chain management" Doctoral dissertation (unpublished), Indian Institute of Management Calcutta, 2007.

ISBN: 978-988-18210-5-8
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)