Social Value Propagation for Supply Chain Formation

Toni Penya-Alba[1], Meritxell Vinyals[2], Jesus Cerquides[1], and Juan A. Rodriguez-Aguilar[1]

[1] IIIA-CSIC, Campus de la UAB, E-08193 Bellaterra, Spain
{tonipenya,cerquide,jar}@iiia.csic.es
[2] ECS, Faculty of Physical and Applied Sciences, University of Southampton, Southampton, United Kingdom. SO17 1BJ
mv2y11@ecs.soton.ac.uk

Abstract. Supply Chain Formation is the process of determining the participants in a supply chain, who will exchange what with whom, and the terms of the exchanges. Decentralized supply chain formation appears as a highly intricate task because agents only possess local information, have limited knowledge about the capabilities of other agents, and prefer to preserve privacy. State-of-the-art decentralized supply chain formation approaches can either: (i) find supply chains of high value at the expense of high resources usage; or (ii) find supply chains of low value with low resources usage. This work presents CHAINME, a novel decentralized supply chain formation algorithm. Our results show that CHAINME finds supply chains with higher value than state-of-the-art decentralized algorithms whilst decreasing the amount of resources required from one up to four orders of magnitude.

Keywords: Supply chain, belief propagation, scalability

1 Introduction

Supply Chain Formation (SCF) is the process of determining the participants in a supply chain (SC), who will exchange what with whom, and the terms of the exchanges [13]. Today's companies are required to dynamically form and dissolve trading relationships at a speed and scale that can be unmanageable by humans, giving rise to the need for automated SCF.

Automating SCF poses an intricate coordination problem to firms that must simultaneously negotiate production relationships at multiple levels of the SC. This problem has been already tackled by the AI literature. Initial contributions [14,5,4] addressed the problem by means of combinatorial auctions (CAs) that compute the optimal SC allocation in a centralized manner. Since even finding a feasible SC allocation is NP-Hard [12,7], sufficiently large SCF problems will be intractable, hence hindering the scalability of the global optimization performed by centralized, auction-based approaches. Furthermore, as argued in [13], even when the computation is tractable, no single entity may have global allocative authority to compute allocations over the entire SC. Thus, there is a need for *approximate distributed* solutions to the SCF problem.

In [13], Walsh et al. proposed to solve the SCF problem in a fully decentralized manner. Each good in the SC is auctioned separately and all auctions run simultaneously without direct coordination. Therefore, each auction allocates a single resource considering the offers to buy or sell submitted by agents. Nevertheless, the approach proposed by Walsh et al. suffers from high communication requirements, as discussed in [11].

Later on, Winsper et al. [16] cast the decentralized SCF problem as an optimization problem that can be approximated using (max-sum) loopy belief propagation [6]. Nonetheless, as shown in [9], the problem representation employed by Winsper et al. leads to exponential memory and communication requirements that largely hinder its scalability. Thus, Penya-Alba et al. provide in [9] a scalable approach to the decentralized SCF problem through a new encoding of the SCF problem into a binary factor graph. However, as we show in this paper, as the number of agents at trade increases, the algorithm in [9] is unable to find SCs whose value are close to the optimal.

To summarize, state-of-the-art decentralized SCF algorithms can either: (i) find supply chains of high value at the expense of high resources usage; or (ii) find supply chains of low value with low resources usage. Against this background, in this paper we present CHAINME, a novel decentralized supply chain formation algorithm that assesses supply chains with higher value than state-of-the-art decentralized algorithms. Furthermore, the resources required by CHAINME are from one up to four orders of magniude less than those used by other algorithms.

The paper is organized as follows. Section 2 reviews previous approaches to distributed SCF. Section 3 describes CHAINME, our main contribution. Section 4 benchmarks CHAINME against previous algorithms for distributed SCF, and section 5 draws conclusions and sets paths for future research.

2 Background and related work

First, section 2.1 reviews periodic double auctions as a basic market mechanism to allocate independent commodities. Then, section 2.2 reviews the state-of-the-art approaches to decentralised SCF.

2.1 Periodic double auctions

Most of the classic auctions examined in the literature are one-sided, in that a single seller or buyer accepts bids from multiple buyers or sellers. Two-sided or double auctions, in contrast, permit multiple buyers and sellers to bid to exchange a designated commodity.

The periodic version of the double auction [17], sometimes termed a call market [8], collects bids over a specified interval of time, then clears the market at the expiration of the bidding interval.

Imagine you are hired as mediator in a vintage computer market. Some of the participants (buyers) will be interested in buying an MSX while other participants (sellers) are interested in selling them. Table 1 shows an example with 4 sell offers (Alice offers to sell an MSX for € 2, Bob for € 3, and so on) and 4 buy offers (Eve offers to pay € 6 for an MSX, Frank offers to pay € 5 and so on). In this setting, the most profitable option for you (the mediator) is to buy from Alice and Bob and to sell to Eve and Frank for a benefit of € 6.

In general, we note the largest possible benefit as π^*. We refer to a π^*-configuration as the set of buyers and sellers that achieve benefit π^*. Alice, Bob, Eve, and Frank are *active* in the π^*-configuration and Carol, Dave, Gene, and Hank are not. The number of consumers at trade in the π^*-configuration (2 in this case) is noted as η.

Seller	**Sell Offer**	**Buy Offer**	**Buyer**
Alice	€ *-2*	€ *6*	Eve
Bob	€ *-3*	€ *5*	Frank
Carol	€ -4	€ 2	Gene
Dave	€ -5	€ 1	Hank

Table 1. Mediation example

In the general case, consider a mediator (m_g) that aims to trade good g. Let $\mathcal{S}_g$ be the set of sellers that are willing to sell g and $\mathcal{B}_g$ the set of buyers that are willing to buy g. Then, determining the π^*-configuration in a periodic double auction amounts to:

1. sorting sell bids descendingly,
2. sorting buy bids descendingly, and
3. matching buyers and sellers in order until it is no longer profitable to do so, that is, until the buy offer is not able to cover the sell offer or there are no more sell offers.

Sellers		**Buyers**	**Fact**
s^1	$\cdots$	b^1	$O^g_{b^1} + O^g_{s^1} > 0$
$\vdots$		$\vdots$	
s^η	$\cdots$	b^η	$O^g_{b^\eta} + O^g_{s^\eta} \geq 0$
$s^{\eta+1}$	$\cdots$	$b^{\eta+1}$	$O^g_{b^{\eta+1}} + O^g_{s^{\eta+1}} < 0$
$\vdots$		$\vdots$	

Table 2. General mediation scenario

Table 2 describes a general periodic double auction, with $s^1, ..., s^\eta, ..., s^{|\mathcal{S}_g|}$ being the sellers ordered descendingly by offer, $b^1, ..., b^\eta, ..., b^{|\mathcal{B}_g|}$ being the buyers ordered descendingly by offer, $O^g_{s^i}$ is the offer of seller s^i and $O^g_{b^j}$ is the offer of buyer b^j. The buyers and sellers over the dashed line are in the π^*-configuration whilst the ones below are not.

Price rules A price rule in a periodic double auction establishes the price τ paid by buyers and received by sellers. There are some constraints that the price has to fulfill to ensure individual rationality for the active agents and fairness for the inactive agents.

To ensure individual rationality, no seller at trade should be paid less than her bid ($\tau \geq -O^g_{s^\eta}$) and no buyer at trade should pay more than her bid ($\tau \leq O^g_{b^\eta}$). To ensure fairness the price cannot be larger than the bid of any seller that is left out of trade ($\tau \leq -O^g_{s^{\eta+1}}$) and it cannot be smaller than the bid of any buyer that is left out of trade ($\tau \geq O^g_{b^{\eta+1}}$). Thus, the price can be any number τ such that $\tau^- \leq \tau \leq \tau^+$ where

$$\tau^- = \max(-O^g_{s^\eta}, O^g_{b^{\eta+1}})$$

and

$$\tau^+ = \min(-O^g_{s^{\eta+1}}, O^g_{b^\eta}).$$

The rule that sets the price at τ^- is known as the (M+1)st price rule, whilst the rule that sets the price at τ^+ is known as the M-th price rule [17].

For our example in table 1, the (M+1)st price rule will set the price at $\tau^- = \max(-(-3), 2) =$ €3 (corresponding to Bob's offer). The M-th price in this case is $\tau^+ = \min(-(-4), 5) =$ €4, established by Carol's offer.

The price interval (τ^-, τ^+) is usually known as the bid-ask interval. Algorithm 1 assesses the bid-ask interval.

Algorithm 1 Bid-ask assessment in a periodic double auction.

function AssessBidAsk($\mathcal{O}^g$)
    /* Determine the π^*-configuration */
    Sort sellers decreasingly by offer getting:
      $\mathcal{S} = \langle s^1, \ldots, s^{|\mathcal{S}_g|} \rangle$
    Sort buyers decreasingly by offer getting:
      $\mathcal{B} = \langle b^1, \ldots, b^{|\mathcal{B}_g|} \rangle$
    $\eta \leftarrow 0$ /*Assess the number of trading agents*/
    while $O^g_{s^{\eta+1}} + O^g_{b^{\eta+1}} \geq 0$ **do**
      $\eta \leftarrow \eta + 1$
    end while
    $\tau^- \leftarrow \max(-O^g_{s^\eta}, O^g_{b^{\eta+1}})$
    $\tau^+ \leftarrow \min(-O^g_{s^{\eta+1}}, O^g_{b^\eta})$
    return (τ^-, τ^+, η)
end function

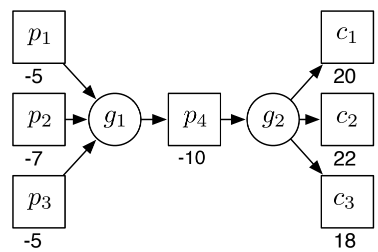


Fig. 1. Example of a TDN.

2.2 Related work

Next, we review the state-of-the-art on decentralized SCF separating the contributions in the literature in two groups: market-based approaches and message-passing approaches.

Market-based approaches In [2] Babaioff and Nisan provide a distributed mechanism providing ex-post individual rationality and incentive compatibility, budget balance and high global economic efficiency in linear supply chains. Later on, Babaioff and Walsh [3] extended this result to supply chains that satisfy the unique manufacturing technologies property, that is, markets where two producers that have the same output, should have exactly the same input goods in the same amounts. In this work we are interested in the more general SCF scenario introduced by Walsh and Wellman in [13].

In order to encode a SCF problem, Walsh and Wellman present the notion of Task Dependency Network (TDN). To efficiently solve a SCF problem over a TDN, they introduced the Simultaneous Ascending (M+1)st Price with Simple Bidding protocols (SAMP-SB and SAMP-SB-D, noted as SAMP-SB* henceforth) [13]. Each protocol is composed of an auction mechanism along with some bidding policies. In what follow we outline both the notion of TDN and the SAMP-SB* protocols.

Task dependency network A TDN is a graphical description of a SCF problem that takes the form of a bipartite directed acyclic graph. Figure 1 depicts an example of TDN. Producers (p_1, p_2, p_3, p_4) and consumers (c_1, c_2, c_3) are represented by rectangles, goods (g_1, g_2) are represented by circles, and links between rectangles and goods represent potential flows of goods. The number below each participant a (either producer or consumer) stands for her activation cost (C_a). The activation cost is positive for consumers, who pay to be part ot the SC, and negative for producers, who must be paid to be part of the SC. For each SC participant a, let $\mathcal{G}_a$ be the set of all goods she is related to, either as a buyer (input goods) or as a seller (output goods). Producers p_1, p_2, p_3 can produce each one a unit of output good g_1, which is an input good of p_4. Similarly, consumers c_1, c_2, c_3 can consume each one a unit of g_2 acting as buyers in such transaction without selling any other good. However, this is not the case of producer p_4 that requires one unit of g_1 to produce g_2. Thus, producer p_4 acts as a buyer for g_1 and as a seller for g_2.

The TDN in figure 1 allows several feasible SC configurations. For instance, configuration $SC_1 : p_1 \rightarrow p_4 \rightarrow c_2$ is feasible, whereas $SC_2 : p_4 \rightarrow c_2$ is not (nobody provides g_1 to p_4). The value of a configuration is assessed by adding the activation costs of participants that are active in the SC. The value of SC_1 is $-5 - 10 + 22 = 7$. The SCF problem is that of finding the feasible configuration with maximum value. In figure 1, that corresponds to configuration SC_1.

Auction mechanism and bidding policies A SAMP-SB* mechanism comprises a set of auctions, one per good. Each auction is run by a different agent, who plays the role of *mediator*. Given a good g, each participant interacts with the mediator of the good, m_g, by submitting her offers to buy or sell g. For instance, in Figure 1 producer p_1 would send an offer message to mediator m_{g_1} to sell g_1. Each auction runs independently of the other auctions in the supply chain. However, all auctions run simultaneously.

Each auction is an increasing periodic double auction with price quotes. When a mediator receives a new bid, she sends each of her bidders a price quote specifying the bid-ask interval that would result if the auction ended in the current bid state. The price quote also reports to each bidder the quantity she would buy or sell in the current state. The price quotes are not issued until all initial bids are received, but are subsequently issued immediately on receipt of new bids.

Each participant follows a simple, non-strategic bidding policy, in reply to mediators' notifications. Thus, participants' bidding behaviour is purely reactive. The SAMP-SB* bidding policies require that, for each auction, the prices of a participant's successive buy offers increase by no less than some generally small) positive number δ_b and the prices of successive sell offers increase by no less than δ_s. Inaction leaves previous bids standing in an auction. More concretely, SAMP-SB* distinguishes between bidding policies for consumers and producers as follows.

Consumer bidding policy. A consumer c not winning her input good g will bid by increasing the (M+1)th price (τ^-) by the minimum required increment δ_b (see equation 1). The bid is issued whenever the consumer's gain is non-negative ($C_c - \tau_g^- - \delta_b \geq 0$), otherwise she will stop bidding. [3]

$$O_c^g \leftarrow \tau_g^- + \delta_b \qquad (1)$$

Producer bidding policy. Every time a producer p receives a quote, if p is currently winning the auction for her output good and losing the auction for some input good g, she increases her last offer for g by the minimum required increment (δ_b). [3]

$$O_p^g \leftarrow O_p^g + \delta_b. \qquad (2)$$

[3] Initial offers submitted for inputs goods are set to 0.

Furthermore, if the quote is coming from an input good, p updates her offer for her output good g' ($O_p^{g'}$) as[4]:

$$O_p^{g'} \leftarrow \max(O_p^{g'} + \delta_s, C_p + \sum_{g \in \mathcal{G}_p \setminus g'} \hat{\tau}_g) \quad (3)$$

where $\hat{\tau}_g$ stands for the perceived cost of input good g. If p is currently winning g, $\hat{\tau}_g$ is τ^-, otherwise $\hat{\tau}_g = \max(\tau^+, \tau^- + \delta_b)$.

Bidding continues until all messages have been received, no participant chooses to revise her bids, and no auction changes its prices, ask prices, or allocation. At this point, the auctions clear; each bidder is notified of the final prices and how many units she transacts per good. Since SAMP-SB may converge to solutions in which some participants obtain a negative utility to participate in the SC, the SAMP-SB-D protocol includes a final phase that allows agents to decommit.

Message-passing approaches In [16] Winsper and Chli propose an alternative graphical representation for the encoding of a SCF problem as a factor graph. Under this representation, the SCF problem is cast as an optimization problem and subsequently approximated by the (max-sum) loopy belief propagation algorithm (LBP)[6]. Nonetheless, as shown in [9], the problem representation employed by Winsper et al. leads to exponential memory and communication requirements that largely hinder its scalability. To overcome this limitation, recently Penya-Alba et al. propose an alternative factor graph encoding for the SC problem [9], the so-called Reduced Binary Loopy Belief Propagation (RB-LBP), that dramatically lowers the max-sum requirements, from exponential to quadratic, while leading to higher valued SCs.

The approaches in [16] and [9] replace the process of bidding in auctions with message-passing between SC participants. Since goods are not explicitly represented in these encodings, message-passing directly takes place between SC participants without any mediator. For instance, during the execution of RB-LBP for the SCF problem represented in figure 1, producer p_1 will iteratively exchange messages with p_4 regarding their trading decisions for good g_1. This process continues until reaching convergence (either all participants agree on their trading decisions or a maximum number of steps is reached). Upon convergence, each message passing algorithm includes a post-processing phase to remove possible incompatibilities from the resulting SC configuration.

3 CHAINME

In this section we describe our approach for decentralized SCF, the so-called CHAINME (CHaining Agents IN Mediated Environments). As we will show in Section 4 state-of-the-art methods for SCF can either produce high-valued SCs

[4] The producer places her first output offer only after receiving the first notification for all her inputs

at the expense of high resource usage, or find low-valued SC configurations while requiring low resource usage. CHAINME aims at providing an algorithm for decentralized SCF that combines the best features of both approaches. That is, CHAINME aims at finding high-valued SCs using as little resources as possible.

The rest of this section is organized as follows. In section 3.1 we provide a detailed description of how CHAINME agents make decisions. Then, in section 3.2 we compare the operation of CHAINME with that of RB-LBP and SAMP-SB*.

3.1 Algorithm Description

Consider a SCF problem such as described by a TDN. In CHAINME, there is an agent for each of the participants in the SC (either producers or consumers). Furthermore, for each of the goods at trade there is an agent that will act as mediator for that good.

CHAINME is a message-passing algorithm involving participants and mediators. Each participant only communicates with the mediators of the goods she wants to acquire or produce. Likewise, each mediator only communicates with the participants willing to buy or sell the good she mediates.

Likewise RB-LBP, agents in CHAINME follow a established protocol that has two main phases. During the first phase, each participant finds out how valuable she is for the SC as a whole when she is active (i.e. producing or consuming goods). Based on that information, during the second phase, each participant decides whether to be active (part of the SC) or not. Section 3.1 details the first phase, whereas section 3.1 details the second phase.

Assessing how valuable participants are. During this phase, agents exchange messages iteratively in turns, from participants to mediators and from mediators to participants. First, each participant submits her offers, encoding her willingness to participate in the SC, to the mediators she is connected to. After that, each mediator m_g communicates to each of her neighboring agents a an approximation of their local social value (S_a^g) for that good. In general, the social value for a group of agents of an event is the difference between the aggregated benefit for those agents if the event happens and the aggregated benefit for those agents if the event does not happen. In our case S_a^g is the social value, for all agents connected to g but a, of a being active. That is, the benefit that the other agents connected to g would obtain if a is active minus the benefit that the other agents connected to g would obtain if a is inactive. We refer to S_a^g as the social value for g of a. After receiving mediators' messages, participants use the received social value estimates to update their offers, which are subsequently sent to mediators. This process continues until messages do not change from iteration to iteration or a maximum number of iterations is reached. At this point, each participant knows how valuable she is for the supply chain.

Next, we detail how participants compute their offers and mediators compute participants' local social values. Henceforth we consider that a good g is mediated by m_g, $\mathcal{S}_g$ stands for the set of participants willing to sell good g, and $\mathcal{B}_g$ is the set of participants willing to buy good g.

How a participant determines her offers Algorithm 2 shows the procedure that a participant (either consumer or producer) follows in CHAINME. At each iteration, each participant makes an offer to each good mediator she is connected to. Given a participant a and a good g, first the agent approximates her value for being active in the SC (V_a) by adding up the local social values reported from the mediators the agent is connected to (line 6). This value also takes into account the activation cost of agent a. Then,

$$V_a = C_a + \sum_{g \in \mathcal{G}_a} S_a^g \tag{4}$$

An agent sends to a mediator m_g her *marginal value*, namely her value without the value contributed by g, to signal her significance in the supply chain excluding g. This amounts to subtracting the local social value of a for g from the agent's value:

$$O_a^g \leftarrow V_a - S_a^g \tag{5}$$

Notice that the higher the contribution from a good to the value of a participant, the lower the offer; the lower the contribution, the higher the offer. Consider the following cases. On the one hand, if a participant knows that her value is large with respect to her local social value for some good g, she will send a high offer to m_g to signal her high value for the rest of the supply chain (excluding good g). On the other hand, if a participant knows that most of her value is contributed by g, she will send a low offer to m_g to signal her low value for the rest of the supply chain.

How a mediator determines social values Recall that the social value for a good g of an agent a is defined as the difference between the benefit for the remaining agents of having her at trade versus not having her at trade. Thus, going back to table 1, to assess the social value of Alice for that good (say g) we need to assess the largest possible benefit *for the other agents* when Alice is active (υ_A^*) and when she is not active (υ_{-A}^*), and assess the social value of as $S_A^g = \upsilon_A^* - \upsilon_{-A}^*$. Since Alice is active in the π^*-configuration, υ_A^* is $\pi^* - O_A^g =$ €6 - (€-2) = €8. To assess υ_{-A}^* we remove Alice bid . The best configuration will definitely have Bob and Eve (since Eve is paying more than Frank and Bob was happy with that what Frank was paying). Furthermore, since Frank's offer covers the price requested by Carol, the best configuration also includes Frank and Carol, and thus υ_{-A}^* is €4, setting the social value of Alice for good g to $S_A^g = \upsilon_A^* - \upsilon_{-A}^* =$€4. In another scenario, it could be the case that Carol requested more than Frank was willing to pay, and then we would not include them.

Next, we detail how a mediator computes the local social value for every agent she is linked to depending on whether the agent is active in the π^*-configuration or not. A general mediation scenario is described in table 2. Consider first the case of any seller s^k active on the π^*-configuration. The benefit when s^k is active, $\upsilon_{s^k}^*$, is simply $\pi^* - O_{s^k}$ (we subtract O_{s^k} because it should not be considered benefit to the other agents but to s^k itself). To assess $\upsilon_{-s^k}^*$ we need to remove bid O_{s^k} and recompute the best configuration. All pairs (s^i, b^i) with $i < k$ are

Algorithm 2 Algorithm run by a participant a.

for all goods $g \in \mathcal{G}_a$ /* Initialize social values */ **do**
    $S_a^g \leftarrow 0$
end for

while not convergence **and not** reached maximum number of iterations **do**
    $V_a \leftarrow C_a + \sum_{g \in \mathcal{G}_a} S_a^g$
    for all goods $g \in \mathcal{G}_a$ **do**
        Send offer $O_a^g \leftarrow V_a - S_a^g$ to the mediator m_g.
    end for
    for all goods $g \in \mathcal{G}_a$ **do**
        Receive social value S_a^g from m_g.
    end for
end while
Set agent a to be available if $V_a \geq 0$
Send the state of agent a to each of her good mediators
/* Determine whether the agent should be active */
while not convergence **and** agent a is available **do**
    Receive from each of her good mediators whether agent a should be active or not.
    Set agent a to be available if all of her good mediators want her to be active
    Send the state of agent a to each of her good mediators.
end while

profitable, (since they were profitable before removing seller s^k). Furthermore, all pairs (s^{i+1}, b^i) with $k \leq i < \eta$ are also profitable, since each pair (s^{i+1}, b^{i+1}) was profitable before removing s^k and the offer for b^i is at least as good as that of b^{i+1}. To assess the best attainable benefit after removing s^k, we must consider whether: (i) to add a new seller (namely $s^{\eta+1}$); or (ii) to remove one of the current buyers (namely b^η). Thus, we have that $v^*_{-s^k} = \pi^* - O_{s^k} - \min(-O^g_{s^{\eta+1}}, O^g_{b^\eta})$. In general, for any seller $\overline{s}$ active on the π^*-configuration, a mediator m_g can compute the seller's local social value for g as:

$$S^g_{\overline{s}} = \min(-O^g_{s^{\eta+1}}, O^g_{b^\eta}) = \tau_g^+. \quad (6)$$

Following the same line of reasoning we can assess the local social value for g of an inactive seller $\underline{s}$

$$S^g_{\underline{s}} = \max(-O^g_{s^\eta}, O^g_{b^{\eta+1}}) = \tau_g^-, \quad (7)$$

the local social value for g of an active buyer $\overline{b}$:

$$S^g_{\overline{b}} = -\max(-O^g_{s^\eta}, O^g_{b^{\eta+1}}) = -\tau_g^-, \quad (8)$$

and the local social value for g of an inactive buyer $\underline{b}$

$$S^g_{\underline{b}} = -\min(-O^g_{s^{\eta+1}}, O^g_{b^\eta}) = -\tau_g^+. \quad (9)$$

Note that the social value of sellers is always positive and that of buyers is always negative[5]. That is reasonable, since if we only take into account how

[5] Under the reasonable assumption that goods are positively priced.

much benefit the other agents make, a seller being active is a positive thing. She is providing a good which has a positive value and the remaining agents benefit from that. On the other side, a buyer being active forces the other agents to provide an additional good, thus incurring in a negative social value. Furthermore, the social value for a good of a participant does not depend on the particular value of her bid, but only on whether the participant is active or inactive. The local social values coincide (disregarding sign) with the bid (τ_g^-) and ask (τ_g^+) values in a periodic double auction that takes the participants' offers as bids. Thus, local social values can be assessed with the help of algorithm 1. This can be seen in the procedure that mediators follow in CHAINME to assess social values of the participants that is computationally described in detail in lines 1-10 of algorithm 3.

Algorithm 3 Algorithm run by a mediator m_g.

1: **while not** convergence **and not** reached maximum number of iterations **do**
2:     **for all** $a \in \mathcal{S}_g \cup \mathcal{B}_g$ /* Receive offers */ **do**
3:         Receive offer O_a^g from participant a.
4:     **end for**
5:     $(\tau_g^-, \tau_g^+, \eta) \leftarrow$ ASSESSBIDASK($\mathcal{O}^g$)
6:     Send τ_g^+ to active sellers.
7:     Send τ_g^- to inactive sellers.
8:     Send $-\tau_g^-$ to active buyers.
9:     Send $-\tau_g^+$ to inactive buyers.
10: **end while**
11: /* Determine which participants should be active */
12: Receive availability status from each neighboring participant.
13: **while not** convergence **do**
14:     Determine π^*-configuration which only involves available participants.
15:     Send each neighboring available participant whether she should be active or not.
16:     Receive availability update from each neighboring available participant.
17: **end while**

Assessing the supply chain configuration Participants and mediators exchange messages in turns following the algorithms described in section 3.1. After this process ends, participants need to decide whether they will be active in the resulting SC or not. This is achieved by iterating a two-step process. During the first step, participants determine whether they are available to be active. During the second step, mediators communicate to participants whether they are eligible to be active.

Each participant decides whether she is available to be active if that is profitable for the SC as a whole. This occurs whenever the addition of the local social values of each of the goods she is connected to together with her activation cost is positive, namely iff her value V_a is greater of equal than zero.

Once each participant has determined whether she is available to be active or not, she sends her availability status to all of her neighboring mediators. Once a participant decides that she is unavailable, she will no further change her status. Each mediator, after receiving the availability status from all her neighboring agents, determines the π^*-configuration by discarding participants that reported themselves as unavailable. After that, the mediator sends each available participant whether she is active or inactive in the π^*-configuration. When a participant receives the status request from each of her neighboring mediators, she decides to be available only if all of her neighboring mediators requested her to be active. She sends her availability status to all the mediators she is connected with. This process continues iteratively until no participant changes her availability status. Participants who are available when this occurs will be the active participants in the SC and will compose the SC configuration. The process to determine which participants are active is described in algorithm 2 (lines 14-21) and algorithm 3 (lines 12-17) for participants and mediators respectively.

3.2 Discussion

Next we analyse the differences and similarities between CHAINME and the state of the art. First, CHAINME, likewise SAMP-SB, uses an infrastructure of agents that mediates participants' interactions. The role of these mediators is to aggregate information that helps participants decide whether to be part of the supply chain or not. Unlike CHAINME and SAMP-SB RB-LBP does not employ mediators for goods. Thus, the producers and consumers of each good make their decisions regarding whether to trade or not through *direct* interactions, instead of *mediated* interactions.

Second, CHAINME and SAMP-SB mediators differ in the semantics of the information they convey to participants. On the one hand, recall that each mediator in SAMP-SB runs an increasing periodic double auction with price quotes. Given a set of bids and asks for a given good g, the mediator sends to each of the participants a price quote specifying: the bid-ask interval $[\tau_g^-, \tau_g^+]$ that would result if the auction ended; and whether the bidder currently wins the bid or not. The bid quote (τ_g^-) signals what a seller must offer to trade, whereas the ask quote (τ_g^+) signals what a buyer must offer to trade. On the other hand, analogously to SAMP-SB, each CHAINME mediator also computes the bid-ask interval for a given set of offers issued by participants. Nonethelesss, each CHAINME mediator distributes between participants information about their current local social values, depending on whether each agent is active or inactive (see equations 6 to 9), instead of what they should offer to become active. Furthermore, a CHAINME mediator does not tell participants whether they are part of the trade or not, and hence participants are unaware of whether they are so.

Third, the semantics of offers utterly differ from CHAINME and SAMP-SB. Given a particular good, the offer of a SAMP-SB participant expresses her interest in participating in the trade for the good. Differently, the offer of a CHAINME participant expresses her marginal value in the supply chain (her value disregarding the good).

Not surprisingly, the way participants use the information received from mediators to compose their offers totally differ between CHAINME and SAMP-SB. SAMP-SB bidding policies make participants compose their new offers by considering their last offers as well as the bid-ask interval (see equations 1 to 3). Successive offers always increase by some minimum required amount with the purpose of winning the auction (for buyers) or covering its costs (for sellers). A CHAINME participant considers her social value for the supply chain together with her local social values (see equation 4). Hence, the higher the contribution from a good to the value of a participant, the lower the offer; the lower the contribution, the higher the offer. Therefore, given a mediator m_g, the purpose of the offer of a CHAINME participant connected to g is to signal how important the participant is for the rest of the supply chain.

Finally, although the behaviour of CHAINME participants may resemble that of RB-LBP participants, there is a fundamental difference. A message from an RB-LBP participant to another participant (participants interact without mediators) indicates how important it is for the supply chain their trading for some good, completetely disregarding what other competitors for the good offer. An offer from a CHAINME participant to the mediator m_g of a good g indicates how important it is for the SC that the participant is included in the trade for the good. This offer is assessed as the aggregate of the social values received from all the mediators she is connected to but m_g. The offer encodes how valuable the participant is for the SC as a whole regardless of good g. Therefore, CHAINME participants benefit from the information aggregation performed by mediators to compose more accurate offers.

4 Experimental Evaluation

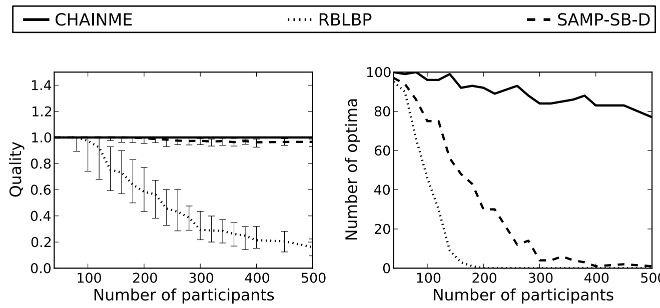
In this section we benchmark CHAINME against the decentralized state-of-the-art SCF algorithms: SAMP-SB, SAMP-SB-D, and RB-LBP. We perform our comparison in terms of solution quality (the SC value) and resource requirements (bandwidth, computation and memory usage).

Following [9], we use the test-suite described in [10] that is specifically designed to generate SCF problems. We generate TDNs with 50 goods spread over four SC levels. We analyze different scenarios by varying the number of participants from 40 to 500. For each scenario, the test suite generates 100 different TDNs.

The code for the algorithms, the generated problems, and the results obtained can be freely downloaded from [1].

We solve each SCF problem with SAMP-SB, SAMP-SB-D, RB-LBP, and CHAINME. We measure bandwidth as the number of messages sent and received by each agent times the size of the message. To measure computation we simply count the number of operations each agent performs.[6]

[6] A fair assessment of the amount of computation performed by mediators is difficult. The costliest operation for both SAMP-SB and CHAINME mediators is assesing the bid-



(a) Median quality.

(b) Number of optimal solutions.

Fig. 2. CHAINME, RB-LBP, and SAMP-SB-D solution quality.

Following [9], we impose a hard limit of 250 iterations after which the execution of RB-LBP and CHAINME is stopped and a solution is assessed. SAMP-SB is run until convergence since convergence is guaranteed in these protocols [13].

Since the distributions obtained for these measures are long-tailed and skewed, we use the median instead of the mean as a measure of central tendency following the recommendations in [15]. Where possible we do also show the 20th and 80th percentile as a measure of dispersion.

Section 4.1 analyzes the quality of the solution obtained (in terms of the value of the SC) by each algorithm. Then, section 4.2 analyzes the resource requirements of each algorithm.

4.1 Solution quality

We normalize solution quality to the 0-1 scale by dividing by the quality of the optimal solution.[7] Hence, a quality of one means that the solution found is optimal. Figure 2a shows the median and dispersion of the solution quality for CHAINME, RB-LBP, and SAMP-SB-D. We observe that CHAINME outperforms RB-LBP and performs slightly better than SAMP-SB-D. Moreover, the quality of the solution obtained by RB-LBP and SAMP-SB-D decreases as the number of participants increases. This effect is more noticeable for RB-LBP.

For problems with more than 100 agents, most of the times SAMP-SB converges to solutions with negative values due to the lack of decommitment phase.

ask interval. CHAINME uses algorithm 1 whose worst-case complexity is $\mathcal{O}(P \cdot \log P)$ (P stands for the number of participants a mediator is connected to). Thus, we record $P \cdot \log P$ operations each time the bid-ask interval is assessed in CHAINME. On the other hand, the assessment of the bid-ask interval in SAMP-SB can be done in $\mathcal{O}(\log P)$ following [17]. Hence, we record $\log P$ operations each time the bid-ask interval is assessed in SAMP-SB.

[7] The optimal solutions are found using a centralized mixed integer programming solver.

Since in terms of resources SAMP-SB is equivalent to SAMP-SB-D we have discarded SAMP-SB from further analysis.

Figure 2b plots the number of problems for which each method was able to find the optimal solution. The number of problems optimally solved by SAMP-SB-D and RB-LBP decreases very rapidly as the number of participants increases. By contrast, CHAINME is able to find the optimal solution in most of the problems, even in scenarios with a large number of participants In the 500 participants scenario,CHAINME converges to the optimal solution in more than 70% of the problems, whereas the other methods almost never find it.

In summary, the value of SC's assessed by CHAINME is higher than that obtained by the other algorithms. Moreover, it finds optimal solutions much more frequently.

4.2 Resource requirements

The memory requirements for an agent is proportional to the number of neighbors for the four algorithms compared. Thus, no current computational environment will be unable to run any of the algorithms due to memory constraints.

Recall from section 3.2 that CHAINME and SAMP-SB-D are mediated algorithms whilst RB-LBP is not. For that reason we benchmark bandwidth usage along four different dimensions: total bandwidth used by all agents, maximum bandwidth used by any participant, total bandwidth used by mediators and maximum bandwidth used by any mediator.

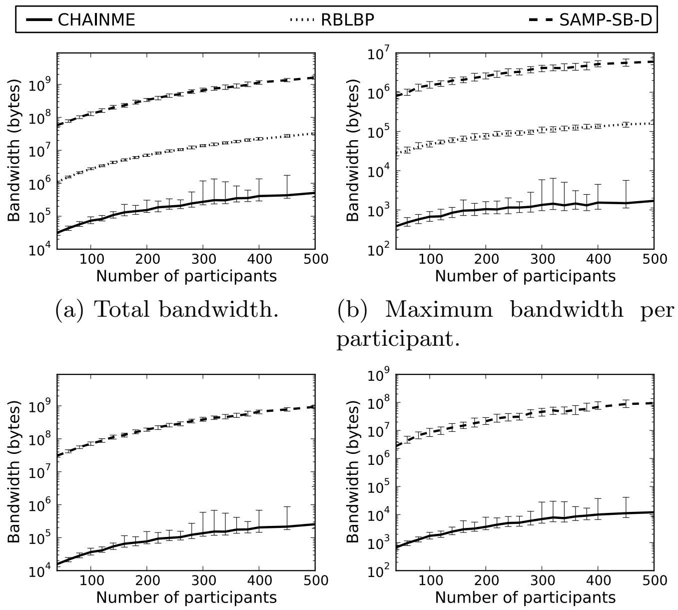
Figure 3 shows how the different algorithms performed in terms of bandwidth usage. Note that RB-LBP is left out of figures 3c and 3d due to its lack of mediator agents. In figures 3a and 3b we see that CHAINME consumes at most 1/60 of the bandwidth used by RB-LBP and at least 3 orders of magnitude less bandwidth than SAMP-SB-D. This difference is also confirmed when we only consider mediators in figures 3c and 3d.

Figure 4 shows how the different algorithms performed in terms of computation. Again, RB-LBP is left out of figures 4c and 4d due to its lack of mediator agents. In figures 4a and 4b we see that the number of operations performed by CHAINME is at least 2 orders of magnitude smaller than that of the runner-up. This difference is confirmed when we only consider mediators in figures 4c and 4d.

In summary, we have seen that CHAINME is able to provide better valued solutions than the state-of-the-art algorithms for decentralized SCF while requiring from one up to four orders of magnitude less resources.

5 Conclusions and Future Work

We have introduced CHAINME, a novel decentralized supply chain formation algorithm where agents use the concept of local social value to determine their worth to the supply chain as a whole. In our experiments, CHAINME outperforms state-of-the-art algorithms RB-LBP and SAMP-SB-D in terms of the value of the



(a) Total bandwidth.

(b) Maximum bandwidth per participant.

(c) Total mediators bandwidth.

(d) Maximum bandwidth per mediator.

Fig. 3. CHAINME, RB-LBP, and SAMP-SB-D bandwidth requirements. Plots use a log-scale for the y axis.

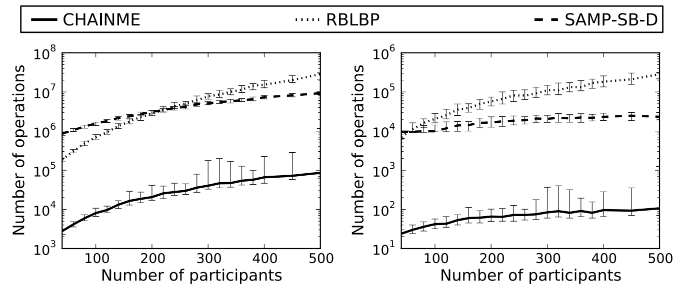
supply chains found. Furthermore, CHAINME consumes less than one tenth of the communication and one percent of the computational resources used by those algorithms.

Note that, since no payment function has been defined, CHAINME (as RB-LBP) should only be considered as a distributed approximate winner determination algorithms.
Therefore, the design of a mechanism would require the definition of a payment function. The design of such payment function and the analysis of the properties of the corresponding mechanisms should be pursued as future work.

The experiments show that the supply chain values obtained by CHAINME are optimal much more often than the state-of-the-art algorithms. Thus, we consider that it could be an approximate, low resources alternative to optimal centralized approaches (such as integer linear programming) in very large SCF scenarios. We plan to explore that path in the future.

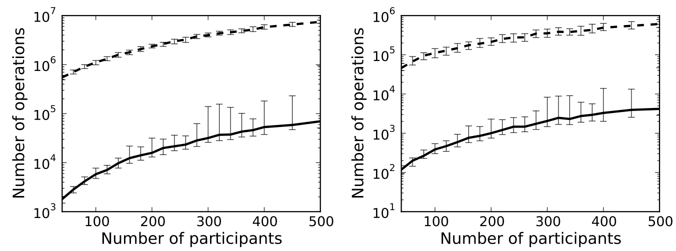
References

1. Removed to preserve anonymity.
2. M. Babaioff and N. Nisan. Concurrent Auctions Across The Supply Chain. *Journal of Artificial Intelligence Research*, 21:595–629, 2004.



(a) Total operations.

(b) Maximum operations per participant.



(c) Total mediators operations.

(d) Maximum operations per mediator.

Fig. 4. CHAINME, RB-LBP, and SAMP-SB-D computations. Plots use a log-scale for the y axis.

3. M. Babaioff and W. E. Walsh. Incentive-compatible, budget-balanced, yet highly efficient auctions for supply chain formation. *Decision Support Systems*, 39(1):123–149, Mar. 2005.
4. J. Cerquides, U. Endriss, A. Giovannucci, and J. A. Rodriguez-Aguilar. Bidding languages and winner determination for mixed multi-unit combinatorial auctions. In *IJCAI*, pages 1221–1226. Morgan Kaufmann Publishers Inc., 2007.
5. J. Collins, W. Ketter, and M. Gini. A multi-agent negotiation testbed for contracting tasks with temporal and precedence constraints. *Int. J. Electron. Commerce*, 7:35–57, October 2002.
6. A. Farinelli, A. Rogers, A. Petcu, and N. R. Jennings. Decentralised coordination of low-power embedded devices using the max-sum algorithm. In *AAMAS (2)*, pages 639–646, 2008.
7. V. Fionda and G. Greco. Charting the tractability frontier of mixed multi-unit combinatorial auctions. In C. Boutilier, editor, *IJCAI*, pages 134–139, 2009.
8. K. A. Mccabe, S. J. Rassenti, and V. L. Smith. Auction Institutional Design : Theory and Behavior of Simultaneous Multiple-Unit Generalizations of the Dutch and English Auctions. *The American Economic Review*, 80(5):1276–1283, 1990.
9. T. Penya-Alba, M. Vinyals, J. Cerquides, and J. A. Rodriguez-Aguilar. A scalable Message-Passing Algorithm for Supply Chain Formation. In *26th Conference on Artificial Intelligence (AAAI 2012)*, Toronto, 2012.
10. M. Vinyals, A. Giovannucci, J. Cerquides, P. Meseguer, and J. A. Rodriguez-Aguilar. A test suite for the evaluation of mixed multi-unit combinatorial auctions. *Journal of Algorithms*, 63(1-3):130–150, 2008.

11. W. Walsh. *Market protocols for decentralized supply chain formation.* PhD thesis, University of Michigan, 2001.
12. W. E. Walsh and M. P. Wellman. Marketsat: An extremely decentralized (but really slow) algorithm for propositional satisfiability. In H. A. Kautz and B. W. Porter, editors, *AAAI/IAAI*, pages 303–309. AAAI Press / The MIT Press, 2000.
13. W. E. Walsh and M. P. Wellman. Decentralized supply chain formation: A market protocol and competitive equilibrium analysis. *J. Artif. Intell. Res. (JAIR)*, 19:513–567, 2003.
14. W. E. Walsh, M. P. Wellman, and F. Ygge. Combinatorial auctions for supply chain formation. In *2nd ACM Conference on Electronic Commerce*, EC '00, pages 260–269, New York, NY, USA, 2000. ACM.
15. R. R. Wilcox and H. J. Keselman. Modern robust data analysis methods: measures of central tendency. *Psychological methods*, 8(3):254–74, Sept. 2003.
16. M. Winsper and M. Chli. Decentralised supply chain formation: A belief propagation-based approach. In *ECAI*, pages 1125–1126, Amsterdam, The Netherlands, The Netherlands, 2010. IOS Press.
17. P. R. Wurman, W. E. Walsh, and M. P. Wellman. Flexible double auctions for electronic commerce : theory and implementation. (July):17–27, 1998.