Department of Computer Science, University of Essex
Technical Report CSM-385, April 2003

Cooperation in Competitions – Constraint Propagation Strategies in Chain-bargaining

Edward Tsang
Department of Computer Science, University of Essex, Wivenhoe Park, Colchester CO4 3SQ, UK
Phone: +44 01206 872774
E-mail: edward@essex.ac.uk
URL: http://cswww.essex.ac.uk/CSP/

Abstract. Electronic business motivates automatic bargaining: computers may not be as good bargainers as expert human bargainers, but by talking to a large number of traders on the Internet, they stand a better chance of getting good deals. While an end-seller may be interested in getting the highest profit from each negotiation, traders have to balance between making large profits in few deals (thus missing opportunities in the failed negotiations) and making smaller profits in larger number of deals. When a deal is to be constructed through a chain of middlemen, these middlemen have to cooperate (in order to construct the deal) while trying to maximize their own profits. The middlemen can be seen as propagating constraints along the chain, with the aim to maximize its profit while satisfying all the bargainers' constraints (failing that, the chain will break down). In this paper, a simple chain-bargaining problem is defined. It is used to study constraint propagation strategies, which are essential components of automatic bargaining.

Keywords: automatic bargaining, constraint satisfaction

1 Introduction

With the growth of activities on the Internet, more machines have to communicate with each other Computer bargaining is quite different from human bargaining, where psychology, emotion, (in the case of audio communication) tone, (in the case of face to face bargaining), eye contact, appearance and facial expressions could all play important parts. There is a need to develop automatic (algorithmic) bargaining agents [11][15][20][21]. Although computer programs may not be as good as human bargainers, programs can talk to many more potential partners. The more potential partners in a free market a trader talks to, the more chance it has in finding good deals. This motivates the study of automatic bargaining programs.

An end-user's objective is obviously to minimize the cost, possibly through hard bargains. Utility may fall as the negotiation takes longer. The end-user (or supplier) may also have to purchase (sell) within a time limit. These factors motivate compromises. For a trader, the failure to strike a deal in situations where there is room for profit means loss of opportunity. In most cases opportunities are limited or involve costs. A trader may attempt to (a) making large profits in fewer deals or (b) making smaller profits in larger number of deals. How could it find a good balance between the two? What makes a good strategy?

In this paper, we focus on a fundamental chain-bargaining problem where a number of middlemen stand between the supplier and the end-customer. The supplier, the middlemen and the end-customer form a chain. For a deal to be completed, all members of the chain must agree on the prices with their corresponding suppliers/buyers. Therefore, members of the chain must balance between cooperation and competition. The middlemen can be seen as propagating constraints along the chain, with the aim to satisfy all the bargainers' constraints. The constraint of a supplier or middleman is to sell (within a limited amount of time) at a price above its cost. The constraint of an end-customer is to buy (within a limited amount of time) at a price below its utility.

Game theoreticians have long been studying bargaining strategies, e.g. see [2][3][4][14]. Optimal strategies have been found for many problems. The game theoretic approach is attractive because it is neat, and results often have provable properties. Realistic applications are far more complex than those studied by game theoreticians so far [9][11]. This motivates research by the agents community, e.g. see [10][11][20][21][23]. By using computer programs and simulations, complex strategies in game theory can be evaluated and verified. Agent technology overlaps with constraint satisfaction [6][18][22] in distributed constraint satisfaction, e.g. see [13][16][17][24][25].

In an earlier paper, we introduced the Simple Constrained Bargaining Game, CBG(S), where one buyer bargains with one seller (to be recapitulated below) [19]. In this paper, the game is extended to chain bargaining. New strategies are introduced. These strategies will be combined to form middlemen. We shall analyse the effectiveness of different middlemen in chain-negotiation and attempt to identify components that make good middlemen.

2 The Chain Constrained Bargaining Game

In an earlier paper, we introduced the Simple Constrained Bargaining Game (CBG(S)) [19]. It involves a buyer and a seller. Only one commodity is involved, for which the seller has a cost and the buyer a utility. Both sellers and buyers have time limits within which they have to sell/buy. They do not know each other's cost and time constraint.

The Simple Constrained Bargaining Game, CBG(S):

The game involves a buyer and a seller.

- (i) The seller is constrained by a *cost* and the number of days within which it has to sell (*DTS*);
- (ii) The buyer is constrained by a *utility* and the number of days within which it has to buy (*DTB*);
- (iii) The seller does not have any information about the buyer's utility and DTB;
- (iv) The buyer does not have any information about the seller's cost and DTS;
- (v) The players make alternative bids, with the seller to bid first;
- (vi) Each player bids exactly once per day;
- (vii) When both players bid for the same price, a sale is agreed;
- (viii) If a sale cannot be agreed before a player runs out of time, the negotiation terminates; no penalty is paid by either player;
- (ix) This is a one-off game: neither player has information about the others' past behaviour and performance.

For example, the seller may have a cost of 100 and it may have to sell within 12 days. The commodity may have a utility of 500 to the buyer, who has to buy within 16 days. The seller may start the bid at, say, 215. The buyer may counter-offer 53, which the seller may counter-offer 192, and the bargain continues until, say, the seller asks for 120 on or before day 12 which the buyer accepts. It is possible that no sale is agreed before one side (in the above example, the seller) runs out of time.

Success of a buyer/seller in one game depends to some extent on luck. However, if enough players play many games against each other, we may find some players better than others. If every pair of players was given the same set of problems, success of a player can be measured by comparing its profit against those by other players. Therefore, performance of a player depends on what other players are involved in the game. The question is whether optimal strategies exist, and if so, what they look like.

This is an interesting problem because it extends what has been done in game theory to something that is still manageable. Given enough effort, it may even be possible to find evolutionary stable solutions. The game forms the basis of more realistic situations which involve more constraints than just cost, utility, sell by dates and buy by dates.

In this paper, we introduce the Chain Constrained Bargaining Game (CBG(C)), which extends the CBG(S). The CBG(C) involves an end-seller, an end-buyer and an ordered list of middleman.

The Chain Constrained Bargaining Game, CBG(S):

The game involves an end-seller, an end-buyer and an ***ordered list*** of middlemen, which form a chain:

[End_Seller, Middleman_1, …, Middleman_N, End_Buyer]

(i) The end-seller is constrained by a *cost* and the number of days within which it has to sell (*DTS*);

(ii) The end-buyer is constrained by a *utility* and the number of days within which it has to buy (*DTB*);

(iii) Each middleman is made up of a buyer and a seller; a middleman may have its own time constraints for completing the negotiations;

(iv) A middleman can only buy from its preceding seller, which is either the end-seller or a predetermined middleman. It can only sell to its succeeding buyer, which is either the end-buyer or a predetermined middleman.

(v) The end-seller, end-buyer and middlemen all have no information about each others' constraints.

(vi) The game is started by the end-seller making its initial offer to the first middleman;

(vii) Control is passed from the end-seller to the first middleman, the second middleman, …, until it reaches the end-buyer when this process is reversed;

(viii) When a middleman receives control from its predecessor, it will make an offer to its successor unless sale has been agreed. If sale has already been agreed, it simply passes control to its successor;

(ix) When the end-buyer receives an offer, it will make a counter-offer to the last middleman in the chain;

(x) When a middleman receives control from its successor in the chain, it will make a bid to its predecessor unless sale has been agreed. If sale has already been agreed, it simply passes control to its predecessor;

(xi) Each player bids no more than once per day;

(xii) When two adjacent players bid for the same price, a sale is agreed between them;

(xiii) When every pair of adjacent players has agreed on the prices, chain negotiation is completed. Profit of each middleman is measured by the difference between its selling price (utility) and its buying price (cost);

(xiv) If a sale cannot be agreed before any player in the chain runs out of time, negotiation in the whole chain terminates. No player earns any profit, even if it has already agreed on prices with its predecessor and successor; no penalty is paid by any player;

(xv) Neither player has information about the others' past behaviour and performance.

The CBG(C) game provides a context for studying an elementary issue in chain negotiation: the issue of how to propagate constraints within a limited amount of time in an environment in which cooperation is required amongst parties with competing goals.

3 Strategies for CBG(C)

When the utility of the buyer is higher than the cost of the seller, there is opportunity for players in the chain to make profits. This is the incentive for all the players to complete a chain in CBG(C). For reference, we call the range between the cost and the utility the "profitable region".

Within the profitability region, agents would try to maximise their own profit. This is the incentive for players to drive hard bargains. However, when a player has a limited number of opportunities in chain negotiation, the cost of not completing a chain is the loss of potential profit. To succeed, a player has to balance between maximizing its own profit (possibly through hard bargaining) and giving its trading partners room for profit, a form of cooperation.

In Tsang and Gosling (2002), experiments on a number of buyers and sellers have been reported [19]. The top players in that paper are joined by top players in a tournament held at University of Essex to form the experiments in the current study. Each player contains a number of rules. Some of these players are complex and best understood by studying their source codes, which is available in [26]. Their general rules are briefly summarised below. In the summaries below, we sometimes omit the simple strategy that all the players bid the bottom line plus a predetermined (player-dependent) minimum profit on the last day.

The Jacob-Seller (dgjaco_s)

This seller accepts bids that are above the cost by a predefined margin, or when it judges (based on the bids history) that the buyer has reached its limit. The general rule is as follows: to start, it offers the cost plus a predefined premium. This offer is reduced linearly until 4th final day. It then offers cost plus a target profit (parameter to the program) for one move. The penultimate move makes an obvious drop in price to tempt the buyer. A minimum profit is demanded in the final offer.

The Keen-Seller-2 (keen_s2)

This simple seller (reported in [19]) is keen to make deals as soon as the bid is above its cost. However, when time is available, it attempts to get a better deal by delaying commitment by one round. The general strategy is to start by offering a very high price, and then reduce it by half towards the cost in each round.

The Stubbings-Seller (pmstub_s)

Unless special cases are detected, such as the buyer has bid below the cost of pmstub_s, the general rule is to offer $MC + \frac{MC}{r} \times \sqrt{r^2 - d^2}$, where MC is the minimum price, which is set at 60% above cost, 1+r is the given number of days to sell and d is the number of days gone. In other words, it drops its offering prices at an increasing rate, till the last day. Pmstub_s also attempts to judge whether the buyer has reached its limit. This is done by checking the ratio $\frac{1+(b_1 - b_2)}{1+(b_2 - b_3)}$ is below 10%, where b_1, b_2 and b_3 are the last, last but one and last but two bids.

The Stacey-Seller (rpstac_s)

This is a complex seller that uses 18 rules to handle different situations. It drives a hard bargain by various sensible means. For example, when the bid is above cost, the bid is accepted if (i) the last two bids are 50% above cost, (ii) the last three bids are 25% above cost or (iii) the last four bids are 15% above cost. The final two days' strategies are fine tuned with 7 rules covering various situations depending on its predetermined margin thresholds and the buyer's latest offer. When no specific rules are applicable, offer price is reduced by 7.5% of the cost per round as long as the offer is above cost.

The Smart-Seller-4 (smart_s4)

This seller was reported in [19]. A *Target* is worked out, principally based on an estimation of the pattern of the buyer's previous bids. Up to three bids are used to project the buyer's next bid. It continues to bargain until it runs out of time, or it believes the buyer has reached its bottom line and the bid is above its cost.

The Keen-Buyer (keen_b)

This simple buyer (modified from the version reported in [19]) is keen to make deals. It accepts any offer that is below its utility. The overall strategy is to bid a low price to start with, and then reduce it by half towards the utility in each round.

The Progressive-Buyer-2 (progress_b2)

This buyer was reported in [19]. The idea is to divide the utility value by the DTB, increasing the bid gradually. This gives the seller a chance to chart its progress and predict its bottom line. The underlying philosophy is that this gives the seller a chance to cooperate should the seller wants to. When the offer is below the utility, it is accepted if (a) there are less than 3 days left; or (b) the latest offer is within 95 and 100% of the previous offer (this is seen to be a sign of the seller reaching its limit).

The Tryhorn-Buyer (mjtryh_b)

This buyer is built upon two important modules: (a) a predictor that estimates the bottom line of the seller and (b) a purchase-adviser that decides whether an offer is acceptable. The predictor attempts to compute the arithmetic progress of the seller's offers. Complex rules were used to compute the next bid, but in general, drives a hard bargain by not raising its bids very much until late in the negotiation. An offer is acceptable if it is the buyer's last day to buy. Whether an offer is acceptable depends on (a) the offer/utility ratio (the lower the better) and (b) the length of the negotiation (the longer the negotiation, the keener it is to accept the offer).

The Sourtzinos-Tsang-Buyer (psourt_b)

This buyer uses a combination of bidding rules. In brief, it bids 1000th of the seller's first offer, and 100th of its second offer, as long as the bids are below its utility. Otherwise, the general rule is used, which is to bid the fraction of the utility that reflects the ratio between the utility and the seller's last offer, i.e. $\frac{Utility^2}{Last\_offer}$.

The Stacey-Buyer (rpstac_b)

This is a complex buyer that uses 20 rules to handle different situations. It drives a hard bargain by various sensible means. For example, even when the offer is below its utility, it delays acceptance. It refuses to raise its bid if the seller has not lowered its price for three rounds. The final two days' strategies are fine tuned with 6 rules covering various situations depending on its preferred margin (which is 35%) and the seller's latest offer. One general rule is to increase the offer by 7.5% per round as long as the bid is below utility.

4 Experiment 1: Bargaining With No Middlemen Involved

Buyers and sellers form the components of chains. To help us understand chain negotiation better, the individual buyers and sellers are played against each other to evaluate their relative strength. One of the questions that we would like to answer later is: will successful buyers / sellers make good components for middlemen?

A Chain Mediator (Version 4.2.1) is used for conducting large-scale experimentation among a set of sellers and buyers. Five sellers are five buyers are used in all the experiments reported in this paper. Sellers and buyers are paired up and played against each other. This means 25 games are played in each experiment. Each seller is played against each buyer in 1,000 randomly generated problems. In our earlier work, we learned that the relative performance between players varies depending on the size of the profitable region (end-buyer's utility minus end-seller's cost). Therefore, we ran experiments with different sizes of profitable regions. In all experiments, costs were randomly generated between 101 and 300. For low-profit experiments, utilities were randomly generated between 301 and 500. For high-profit experiments, utilities were generated between 1001 and 1300. Both buyers and sellers were given a random number of days, generated between 3 and 20, to buy and sell. The parameters used in the experiments throughout this paper are shown in Table 1 below.

Table 1: Parameters used in the experiments

Parameter set ID	p1	p2	p4
Min Cost	101	101	101
Max Cost	300	300	300
Min Days to Sell	3	3	3
Max Days to Sell	20	20	20
Min Utility	1001	301	5101
Max Utility	1300	500	5300
Min Days to Buy	3	3	3
Max Days to Buy	20	20	20

The performance of the players are summarised in Table 2. The profits are normalized for comparability between the two different runs. The normalised results are presented in Figure 3. It is important to note that the performance of a player depends on the population profile.

Table 2: Average Profits in Pairwised Negotiations

Player	Low profit	High profit	Profit over all games
dgjaco_s	101	169	270
keen_s2	68	295	363
pmstub_s	99	398	497
rpstac_s	76	430	506
smart_s4	67	329	396
keen_b	49	452	501
mjtryh_b	96	681	777
progress_b2	57	542	599
psourt_b	95	522	617
rpstac_b	122	690	812
Mean	83.00	450.80	533.80
Standard Deviation	21.85	156.63	163.91

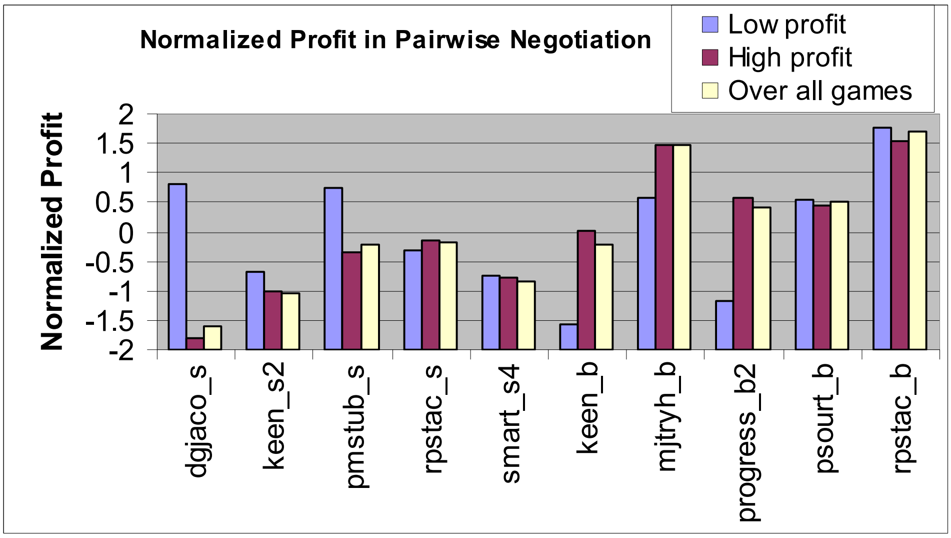

Figure 3: Normalized Profit in pairwise negotiations

The results shown here agree with our early results. Performance of the players varies depending on the size of the profitable region. As we found out in the past, the most dramatic example was the seller dgjaco_s, which was the champion seller when the profitable region was narrow, but poorest seller when the profitable region was wide. There is probably room for improving its game in highly profitable situations. The results are also consistent with earlier results in that the buyers generally scored higher than sellers. One plausible explanation is that, if the seller and buyer have the same time limit, the buyers have the last chance to bid.

Results in Figure 3 show that when the profitable region is wide, mjtryh_b and rpstac_b (both hard-bargainers) are the most successful buyers and pmstub_s and rpstac_s are the most successful sellers. When the profitable region is small, rpstac_b is by far the most successful buyer and dgjaco and pmstub are the most successful sellers. Overall, one can say that rpstac_b is the most successful and reliable buyer, followed by mjtryh_b. pmstub_s and rpstac_s are the most successful sellers.

To gain better understand of the aggregate scores, the individual pairwise results are recorded, as shown in Table 4. We also compute the percentage of potential profits that each pair of players realized – that is the total scores between two players divided by the summation of utility minus cost in all the games (bear in mind that the same set of games are played by every pair of players). For example, the realization rate between dgjaco_s and progress_b2 is (140.5 + 703.4) / 949.9 ≈ 89%. The results are shown in Table 5.

Table 4: Average profit per game between each pair of players
(Parameter set p1, 1,000 games between each pair of players, average potential profit per game is 949.9)

		keen_b		progress_b2		mjtryh_b		psourt_b		rpstac_b	Average
dgjaco_s	200.0	749.9	140.5	703.4	143.3	806.6	200.0	749.9	161.5	788.4	**169.1**
keen_s2	553.2	396.7	54.1	703.6	162.8	787.1	443.2	506.7	263.4	686.5	**295.3**
pmstub-s	434.3	515.6	400.2	481.9	372.9	577.0	429.8	520.1	352.1	597.8	**397.9**
rpstac_s	661.6	288.3	201.3	343.0	430.1	510.1	551.8	398.1	305.7	644.2	**430.1**
smart_s	605.9	311.7	254.8	476.0	185.0	723.4	379.6	434.4	218.0	731.9	**328.7**
Average		**452.4**		**541.6**		**680.9**		**521.8**		**689.7**	

Table 5: Percentage of potential profit realized between each pair of players

	keen_b	progress_b2	mjtryh_b	psourt_b	rpstac_b	Average realization
dgjaco_s	100%	89%	100%	100%	100%	**98%**
keen_s2	100%	80%	100%	100%	100%	**96%**
pmstub-s	100%	93%	100%	100%	100%	**99%**
rpstac_s	100%	57%	99%	100%	100%	**91%**
smart_s	97%	77%	96%	86%	100%	**91%**
Avg gain per game	**99%**	**79%**	**99%**	**97%**	**100%**	**95%**

If we pay attention to rpstac_s, the best performed seller of all, we can see that keen_b contributed heavily to its high scores (rpstac_s scored 661.6 when keen_b scored 288.3). In fact, rpstac_s scored less than progress_b2 (201.3 vs 343.0), mjtryh_b (430.1 vs 510.1) and rpstac_b (305.7 vs 644.2). The fact is, all other sellers scored less than these buyers. It is also worth paying attention to the fact that rpstac_s realized 99% to 100% of the potential profits with most buyers, but did it badly with progress_b2 – together they only realized 57% of the profits.

Amongst the buyers, progress_b2 managed to beat all its opponents in pairwise competition. It scored heavily against keen_s (703.6 vs 54.1). However, on average, progress_b only realized 79% of the potential profits with its opponents. Progress_b2 is quite willing to raise its bids. The low realization rate suggests that progress_b2's strategy for recognizing deadlines may be problematic.

The best performed buyers, mjtryh_b and rpstac_b were successful not just because they manage to gain more profit than their opponents, but also because they did not miss much opportunities. Despite the fact that they drive hard bargains, they realized 99% (mjtryh_b) and 100% (rpstac_b) of all the profits with its opponents. This suggests that between these buyers and their opponents, deadlines were recognized most of the time.

5 Experiment 2: Chains with Mixed Middlemen

A middleman is made up of a buyer and a seller. While it is possible to implement dedicated middlemen we would like to find out whether the buyers and sellers that we have studied so far could be used to make up good middleman. Furthermore, we would like to find out what components (buyers and sellers) make up successful middlemen.

We generated chains with one, five and ten random middlemen between an end-seller and an end-buyer. For chains with one middleman, we generated 1,000 chains. Each chain is run on 100 problems generated from the low and high profit problems as described in the previous section. This gives each player (end-seller, end-buyer or middleman) over 8000 occurrences in the runs for statistical reliability. For chains with five middlemen, problems were generated from parameter p1 in Table 1 (for low profit – as profits between 701 and 1299 is to be shared by seven players in each chain) and p4 (for high profit). Only 20 games were played by each chain, as there would be enough scorings to be collected in each chain to make the results statistically reliable (on average 4000 data points per player per parameter set). The same applied to chains with ten middlemen. Scores are shown in Table 6.

Table 6: Scores in Mixed Chains – Average Profits Per Game

Length of chain	1	1	5	5	10	10
Parameter Set	p2	p1	p1	p4	p1	p4
dgjaco_s-	40	115	6	18	0	2
keen_s2-	13	90	1	3	1	0
pmstub_s-	37	242	14	52	1	6
rpstac_s-	28	186	5	29	1	2
smart_s4-	19	172	5	30	1	3
keen_b-dgjaco_s	30	141	9	43	2	17
keen_b-keen_s2	23	218	14	107	1	20
keen_b-pmstub_s	41	350	18	203	2	26
keen_b-rpstac_s	33	193	11	149	1	16
keen_b-smart_s4	22	165	14	119	2	21
mjtryh_b-dgjaco_s	38	221	9	114	2	7
mjtryh_b-keen_s2	31	260	12	138	3	16
mjtryh_b-pmstub_s	37	463	16	205	2	17
mjtryh_b-rpstac_s	33	284	9	165	1	13
mjtryh_b-smart_s4	32	277	12	157	1	16
progress_b2-dgjaco_s	31	187	9	106	1	19
progress_b2-keen_s2	26	224	15	144	1	24
progress_b2-pmstub_s	32	315	19	212	1	30
progress_b2-rpstac_s	25	240	15	176	2	22
progress_b2-smart_s4	20	207	16	204	2	33
psourt_b-dgjaco_s	49	170	10	82	2	11
psourt_b-keen_s2	37	219	12	139	2	15
psourt_b-pmstub_s	35	362	19	216	5	20
psourt_b-rpstac_s	30	275	11	112	1	11
psourt_b-smart_s4	27	191	13	158	2	18
rpstac_b-dgjaco_s	43	282	15	143	2	21
rpstac_b-keen_s2	30	270	18	164	2	26
rpstac_b-pmstub_s	35	375	20	208	1	27
rpstac_b-rpstac_s	31	271	15	147	3	26
rpstac_b-smart_s4	39	278	17	155	2	29
keen_b-	17	182	4	59	1	8
mjtryh_b-	36	335	13	256	1	14
progress_b2-	18	253	24	325	2	65
psourt_b-	41	312	17	223	4	27
rpstac_b-	47	431	25	303	0	62
Mean	31.6	250.17	13.20	144.69	1.66	19.71
Standard Deviation	8.4	82.32	5.32	75.79	0.98	13.68

It is interesting to observe that results on problems generated using parameter p1 produced very low average profits for chains with 10 middlemen. The profitable region in problems generated with parameter p1 is between 701 and 1299. Shared between 12 players in each chain, there should be, on average, profit regions of 58 to 108 in each negotiation. Experimental results show that the average profit per game per player was less than 2. One plausible explanation is that in long chains, it only takes one uncompromising player to prevent the chain completing (leading to zero profit to all players in the chain). This will be further examined later.

The scores were normalized (by column in Table 6) so that results from different runs can be compared and combined. Results with chain length 10, parameter set p1 were discarded, as the scores were too low to be significant. Normalized results are shown in Figure 7.

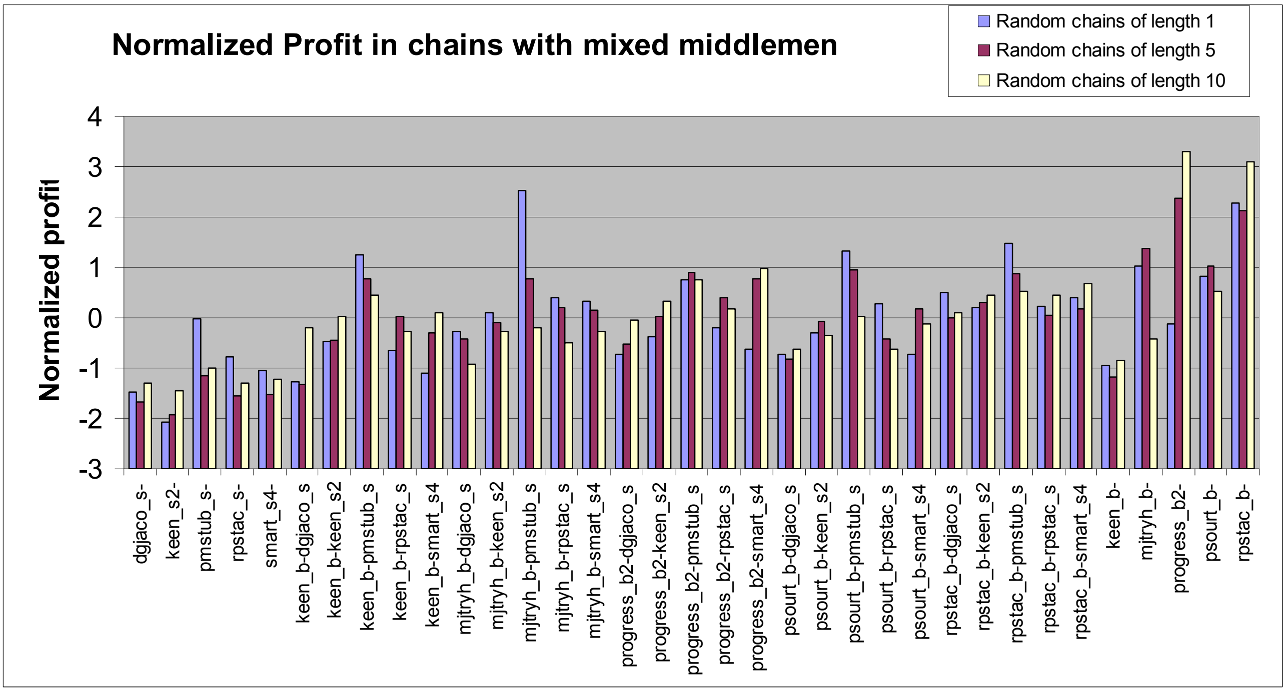

Figure 7: Normalized profits for players in chains with mixed middlemen

Figure 7 shows that results are sensitive to length of the chains. For chains with single middleman, (mjtryh_b, pmstub_s) was by far the best performer. In fact, all middlemen that used pmstub_s as their buyers performed rather well. As the length of the chains grow, profits of end-buyers progress_b2 and rpstac_b increase. No middlemen appeared to dominate the game in chains of length 5 and 10, though some performed than average; e.g. (progress_b2, smart_s4), (progress_b2, pmstub_s), (keen_b, pmstub_s) and (rpstac_b, pmstub_s). Overall, 7 middlemen (keen_b, pmstub_s), (progress_b2, pmstub_s), (psout_b, pmstub_s), (rpstac_b, dgjaco_s), (rpstac_b, pmstub_s), (rpstac_b, rpstac_s), (rpstac_b, smart_s4) all scored above average in chains of all length in the experiment.

The performance of (progress_b2, pmstub_s) is worth highlighting. It performed consistently in chains of length 1, 5 and 10. Recall from Table 5, progress_b2 has a relatively lower rate of realizing profits (79% on average). On its own, progress_b2 was only an average buyer. But combined with pmstub_s, it formed part of a consistently successful middleman. This perhaps highlights the difficulties in finding good combinations of buyers and sellers to form middlemen.

To understand the results better, a number of runs were studied in details. Following is the results of one particular run with four middlemen in a chain that contains all the players described in this paper. Both end-sellers and end-buyers have 8 days to sell/buy. The cost is 200 and the utility is 1200, which means there is a profitable region of 1000. Negotiations are presented in lists, in reverse chronological order. For readability, all sellers' offers are preceded by "+" and all buyers' bids are preceded by "-".

End-seller: dgjaco_s, cost 200

[-240, +240, -0, +290, -0, +290, -0, +317, -0, +345, -0, +372, -0, +400]

Middleman 1: (mjtryh_b, keen_s2)

[-290, +290, -0, +340, -0, +390, -0, +491, -0, +666, -0, +988, -0, +1604]

Middleman 2: (progress_b2, pmstub_s)

[+632, -632, +703, -2, +925, -2, +1136, -2, +1496, -2, +2087, -2, +3146, -2, +5132]

Middleman 3: (rpstac_b, rpstac_s)

[+973, -973, +17374, -891, +18783, -788, +20307, -642, +21954, -410, +23735, -6, +25660]

Middleman 4: (keen_b, smart_s4)

[+1055, -1055, +36150, -995, +45539, -935, +54458, -875, +62996, -815, +71205, -103, +102640]

End-buyer: psourt_b, utility 1200

Profits: dgjaco_s: 40, (mjtryh_b, keen_s2): 50, (progress_b2, pmstub_s): 342, (rpstac_b, rpstac_s): 341, (keen_b, smart_s4): 82, psourt_b: 145

With a cost of 200, end-seller dgiaco_s started the chain negotiation by offering 400 to the middleman (mjtryh_b, keen_s2), who saw it as its potential cost. Middleman (mjtryh_b, keen_s2) in turn offered 1,604 to the middleman (progress_b2, pmstub_s), who marked it up to 5,132 and offered to (rpstac_b, rpstac_s). This carries on until the offer of 102,640 reached the end-buyer psourt_b, who counter-offered 103 (as explained before, -103 should be read as "bid of 103" by seller psourt_b rather than "negative 103") to middleman (keen_b, smart_s4). Bids of 6, 2 and 0 were passed from one middleman to another until 0 was bid to the end-seller.

The first sale was agreed on day 7 (one day before its own deadline), when middleman (rpstac_b, rpstac_s), which had an offer of 703 for its approval from (progress_b2, pmstub_s), accepted the bid of 973 by (keen_b, smart_s4). This was followed by (keen_b, smart_s4)'s acceptance of the end-buyer's bid of 1055, confirming its profit of (1055-973=) 82. Later but still on day 7, middleman (rpstac_b, rpstac_s) refused to accept the offer of 703 by (progress_b2, pmstub_s), which would have given it a profit of (973-703=) 270. Instead, (rpstac_b, rpstac_s) counter-offered 632 to (progress_b2, pmstub_s). With this offer by its buyer, (progress_b2, pmstub_s) accepted the offer of 290 by (mjtryh_b, keen_s2)., who in turn accepted the offer of 240 by end-seller dgjaco_s, taking a profit of (290-240=) 50. On day 8, the final day, (progress_b2, pmstub_s) accepted the bid of 632 by (rpstac_b, rpstac_s), taking the profit (973-632=) 341.

What we have learned from analysing the above and other scenarios is that, in general, whenever a hard-bargaining seller (buyer) is present in a chain, high cost (low utility) is imposed to the next middleman. In the above example, when middleman (rpstac_s and rpstac_b) was offered 5,132 by its supplier, it offered 25,660 to the next middleman (targeting a staggering profit of 20,532). Subsequent bids and offers by (rpstac_b and rpstac_s) all left it with potentially very high profits. For example, despite increases of bids by (keen_b, smart_s4) from 6, 410, 642, 788 to 891, (rpstac_s and rpstac_b) insisted to bid 2 to (progress_b2, pmstub_s).

Close observation reveals that completion of the vast majority of chains, if they do complete at all, happened on the final couple of day limits. Effective propagation of constraints is essential for the completion of a chain. We observed from Table 5 that rpstac_s was rather accurate in recognizing deadlines. In the above example, it chose the right time to commit itself. But the fact that it was in a position to accept the price 973 was because it was offered the price 703 by its supplier. From the negotiation history above, it can be seen that the completion of the chain was brought about by two streams of progress: the first stream is the steady raise of bids by psourt_b to (keen_b, smart_s4), and keen_b's steady raise of bids to (rpstac_b, rpstac_s). The second stream is the steady drop of offering prices by dgjaco_s to (mjtryh_b, keen_s2), and keen_s2's steady drop of offering prices to (rpstac_b, rpstac_s). These two streams of concession gave (rpstac_b, rpstac_s) a chance to initiate completion of the chain. Most chains needs two streams of price concessions (not necessarily steady prices changes) to complete.

The hard-bargainers attempt to gain high returns at the risk of not completing chains. Many chains of length 10 (which have high probabilities of containing hard-bargainers) did not complete. Results show that all the middlemen that used rpstac_b as their buyers obtained above-average performance in chains of all length tested. Note that when a chain does not complete, the hard-bargainers are not the only players that lose out – all the players in the chain score zero. Easy-going bargainers' strategy is to pick up profits in chains that complete. Middlemen who allow others to estimate their bottom-line, e.g. (progress_b2, pmstub_s) and (keen_b, pmstub_s), performed reasonably well in the experiments.

6 Experiment 3: Chains with Uniform Middlemen

Aggregate results can hide a lot of facts. To analyse the performance of individual middlemen and their components, we conducted experiments on chains with uniform middlemen. This means we only used one buyer and one seller to make up each chain. An example of a uniform chain is:

[pmstub_s, (mjtryh_b, pmstub_s), (mjtryh_b, pmstub_s), …, mjtryh_b]

All 25 combinations of buyers and sellers were tested. The aim was to find out the combinations that make successful middlemen – middlemen that perform well in chains that contain only middlemen of their kind. By removing interactions between middlemen, our secondary aim is to identify middlemen that failed to complete chain negotiations.

As before, chains with one, five and ten middlemen were generated. They were run on problems with low as well as high profits, which were generated with parameters as descried in the previous section. Each uniform chain of length 1 was given 1000 problems to run. Each uniform chain of length 5 and 10 was given 500 problems to run. Importantly, every chain of identical length is given the identical set of problems. This allows scores between different chains to be compared directly. Results are shown in Table 8. The normalized results are shown in Figure 9.

Table 8: Scores in Uniform Chains – Average Profits Per Game

Chain Length:	1	1	5	5	10	10
Parameter Set:	p2	p1	p1	p4	p1	p4
dgjaco_s-	59	149	86	116	46	102
keen_s2-	28	125	19	47	7	15
pmstub_s-	43	193	75	101	40	78
rpstac_s-	26	144	32	60	14	35
smart_s4-	27	134	68	44	31	60
keen_b-dgjaco_s	62	200	129	200	53	200
keen_b-keen_s2	58	359	129	719	61	335
keen_b-pmstub_s	30	389	81	563	43	257
keen_b-rpstac_s	42	273	73	414	44	235
keen_b-smart_s4	63	300	76	686	50	293
mjtryh_b-dgjaco_s	50	156	81	128	32	107
mjtryh_b-keen_s2	55	280	56	409	11	80
mjtryh_b-pmstub_s	10	259	26	203	12	89
mjtryh_b-rpstac_s	24	184	29	96	8	74
mjtryh_b-smart_s4	18	207	23	157	11	70
progress_b2-dgjaco_s	58	132	103	123	26	121
progress_b2-keen_s2	20	64	24	145	6	46
progress_b2-pmstub_s	4	232	6	42	4	20
progress_b2-rpstac_s	8	106	11	48	4	28
progress_b2-smart_s4	13	124	6	52	3	20
psourt_b-dgjaco_s	56	200	72	166	35	106
psourt_b-keen_s2	62	424	62	566	32	171
psourt_b-pmstub_s	23	321	54	423	35	211
psourt_b-rpstac_s	27	220	11	65	7	33
psourt_b-smart_s4	20	207	26	175	14	84
rpstac_b-dgjaco_s	38	185	57	104	37	98
rpstac_b-keen_s2	28	260	27	178	8	55
rpstac_b-pmstub_s	12	301	11	71	6	36
rpstac_b-rpstac_s	19	266	12	64	7	29
rpstac_b-smart_s4	25	259	25	148	14	83
keen_b-	34	226	87	1108	66	827

mjtryh_b-	41	378	100	1134	47	720
progress_b2-	35	291	116	1161	62	904
psourt_b-	48	381	58	916	42	440
rpstac_b-	31	445	56	599	35	420
Mean	34.20	239.26	54.49	320.89	27.23	185.20
Standard Deviation	17.14	94.05	35.59	335.51	19.16	222.92

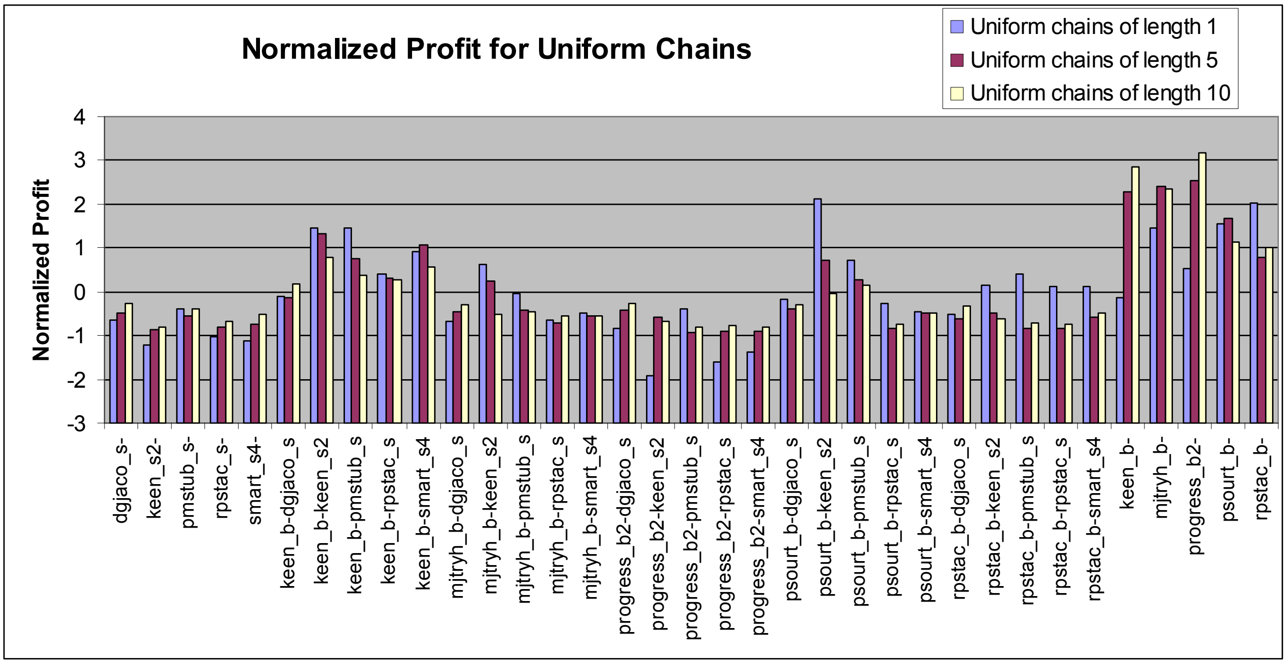

Figure 9: Normalized profits for players in chains with uniform middlemen

When a uniform chain breaks down, it is caused by breakdown between the only two players involved in the chain. Therefore, one should expect results in this experiment to reflect the frequency that the two players fail to agree on prices (see Table 5). Results in Figure 9 support this analysis. Of the 25 middlemen, (keen_b, keen_s2), (keen_b, pmstub_s), (keen_b, rpstac_s), (keen_b, smart_s4), (psourt_b, pmstub_s) and to a certain extent, (psourt_b, keen_s2) consistently scored above average in chains of different length. All but one of them involve keen_b or keen_s2, easy-going players. This is consistent with our expectation.

Also as expected is the fact that the hard-bargainers, rpstac_b and rpstac_s, scored badly in this experiment (except when rpstac_s was combined with keen_b). That is because when these middlemen fail to complete a chain, all members of the chain are copies of themselves. When a chain with mixed middlemen failed to complete, other middlemen did not score either. Therefore, the hard-bargainers' relative scores were not damaged by incompletion, as they would in uniform chains. Note that results in this experiment should not be used to conclude that hard-bargaining is a poor strategy, because in realistic applications, middlemen are likely to be different in the strategies that they use.

Close observation on the detailed runs reveals that middlemen such as (rpstac_b, rpstac_s) and (rpstac_b, pmstub_s) failed to compete most uniform chains with length 5 or 10. Chains were only completed when the days to sell is equal to or one above days to buy. In problems other than those, they dropped their offers or raised their bids too late for the other to respond. Following shows part of the chain negotiation in a chain with five (rpstac_b, pmstub_s) middlemen. There are 8 days to sell and 9 days to buy, in a game where cost is 200 and utility is 1,200.

End-seller: pmstub_s, cost 200

[-56, +220, -4, +485, -4, +544, -4, +583, -4, +609, -4, +627, -4, +637, -4, +640]

Middleman 1: (rpstac_b, pmstub_s)

[-87, +242, -12, +1176, -12, +1479, -12, +1699, -12, +1854, -12, +1964, -12, +2028, -12, +2048]

Middleman 2: (rpstac_b, pmstub_s)

[-135, +266, -43, +2851, -40, +4022, -38, +4949, -36, +5646, -34, +6153, -32, +6457, -30, +6554]

….

It is worth mentioning that Chain Mediator (Version 4.2.1) implements an upper bound on the offer, which is 100 times of the utility. In the above run, middleman 5's initial offer to the end-buyer was capped to (1,200x100=) 120,000.

If we focus on the negotiation between the first and second middlemen, we can see the typical difficulties in this chain negotiation. The first middleman was offered 640, which it marked up to 2,048 and offered to the second middleman. The first middleman reduced its offer to the second middleman slowly, as pmstub_s does. On the buying stream, the second middleman (which had a cost of 30 on day 1) counter-offered 12 to the first middleman. However, rpstac_b kept bidding 12 to the first middleman, despite gradual (though small) bid reductions by the third middleman. This does not allow the first middleman to raise bids to the end-seller. By the time the second middleman raised its bid to the first middleman from 12 to 87, it was too late for the latter to respond even if 87 was an acceptable price (in this case it was not). In this run, we can see that progress was made between middlemen 2 and 3, but this progress did not pass on to the first middleman and beyond soon enough. This illustrates the typical difficulties in uniform chains that contain hard bargainers only.

7 Further Analysis

We have reported above two sets of experiments: chains with mixed (randomly generated) middlemen and chains with uniform middlemen. We have used chains of length 1, 5 and 10 in both sets of experiments. For each chain length, we generated problems with low and high profits. To gain a rough idea on how each player performed, we averaged all the normalized profits in these experiments. The results of the players in uniform, mixed and all chains are contrasted in Figure 10.

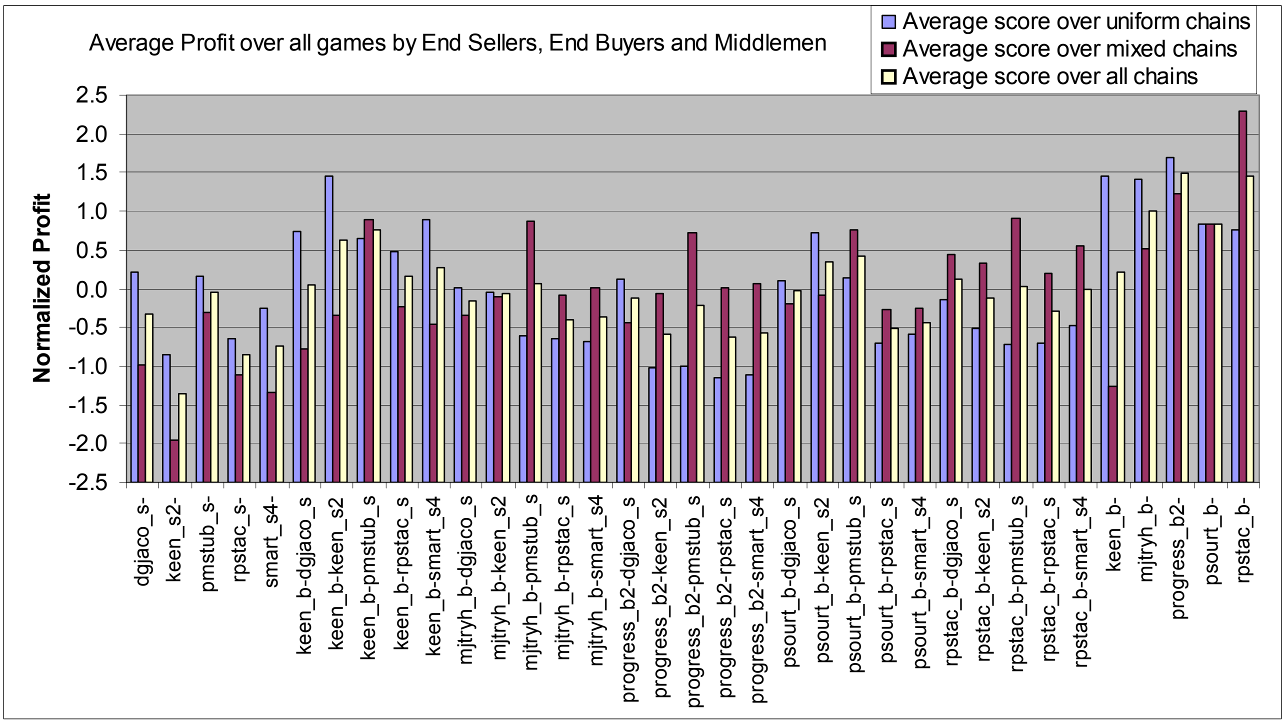

Figure 10: Normalized profits for players in all chains

The overall picture shows that the buyers are, on average, more successful than the middlemen, who are relatively better than the sellers. Amongst the middlemen, (keen_b, pmstub_s) and (psourt_b, pmstub_s) are consistent performers over both uniform and random chains. With the exception of (keen_b, pmstub_s), middlemen that involve keen_b were good at uniform chain negotiation, but poor in chains with mixed middlemen. Middlemen (mtryh_b, pmstub_s), (progress_b2, pmstub_s) and (rpstac_b, pmstub_s) were outstanding in chains with mixed middlemen, but weak in uniform chain negotiation.

Given that different players in the real world tend to use different strategies, are the results in uniform chains relevant to the evaluation of middleman strategies? The answer is probably positive. One can imagine that in a dynamic environment, where middleman may change their strategies, more middlemen will adopt the successful strategies. This is either because weaker middlemen are eliminated, or successful middlemen duplicate themselves, or a combination of both. As more and more middlemen adopt the successful strategies, successful middlemen will deal with more and more middlemen that resemble themselves. Therefore, middlemen that perform badly in uniform chains will find their performance deteriorate.

Next, we attempt to find out which sellers and buyers make good components of middlemen. For this purpose, we looked at the contribution of each player in all the chains reported above. For example, if middleman (keen_b, dgjaco_s) scored a normalized profit of 0.16 in a game, this is credited to both keen_b and dgjaco_s. All the credits are averaged. The results are presented in Table 11 and Figure 12.

Table 11: Average normalized results by components over all chains

Players	Average over individual, uniform chains	Average over individual, random chains	Average over individual, all games
dgjaco_s-	0.1787	-0.3843	-0.0772
keen_s2-	-0.0430	-0.3700	-0.1916
pmstub_s-	-0.2313	0.6411	0.1653
rpstac_s-	-0.5625	-0.2505	-0.4206
smart_s4-	-0.3696	-0.2364	-0.3091
keen_b-	0.9419	-0.3612	0.3496
mjtryh_b-	-0.0945	0.1424	0.0132
progress_b2-	-0.4071	0.2517	-0.1076
psourt_b-	0.0821	0.1331	0.1053
rpstac_b-	-0.2990	0.7859	0.1941

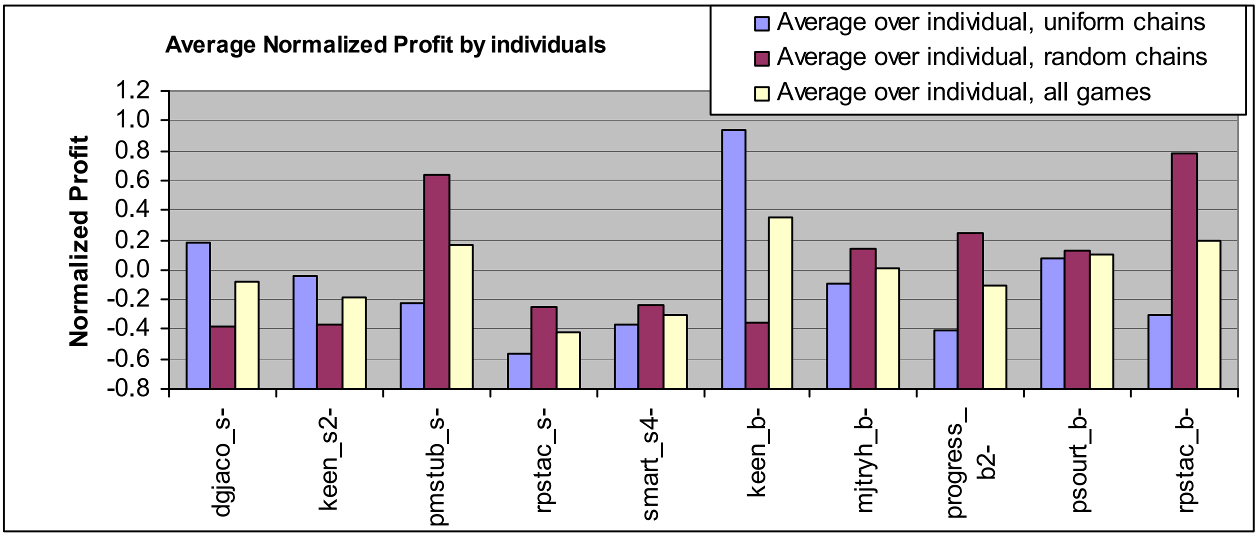

Figure 12: Average normalized profits by individuals in chain negotiation

Over all experiments, keen_b and rpstac_b were the best buyers and pmstub_s was the best seller for forming middlemen. The success of keen_b was mainly due to its scores in uniform chains. Its performance in chains with mixed middlemen was consistent with its performance in pairwise negotiation. The high overall score of both rpstac_b and pmstub_s were due to their relative performance in chains with mixed middlemen. They both drive hard bargains by conceding slowly. Being reasonably predictable by sophisticated opponents, pmstub_s allows the opponent to compromise if it wants to. The success of rpstac_b could be attributed to its sophisticated, fine-tuned strategy. What is less clear is the weak performance of rpstac_s, which strategy mirrors rpstac_b. Another interesting observation is that, despite using a relatively simple strategy, psourt_b was the only buyer that performed above average in both uniform and mixed chains.

It is interesting to compare results in Figure 12 to those presented in Figure 3. First it must be noted that the scales in the two figures are different. The differences between players are less dramatic in chain negotiations than in pairwise negotiations. It is the relative performance that we are focusing on. In chains with mixed middlemen, rpstac_b is as successful as it was in pairwise negotiations. The performance of mjtryh_b is somehow diminished in chain negotiation. Amongst the sellers, the performance of pmstub_s and rpstac_s were comparable in pairwise negotiation. However, in chain negotiation, pmstub_s was outstanding, but rpstac_s was relatively poor.

8 Conclusions and Future Work

In this paper, we have focussed on a simple chain constrained bargaining game, CBG(C). The problem is used as a platform for studying constraint propagation strategies in adverse situation. While each bargainer attempts to maximize its own profit, it has to recognize others' time and price constraints if it were to succeed in this game. Within a limited number of iterations, bargainers have to propagate their constraints effectively to each other in order to complete a chain.

A number of hand-coded, rule-based bargainers have been analysed. We can decompose these bargainers into general components. These include:

- The strategy for opening bid – for example, dgjaco_s demands a small profit margin but rpstac_s demands a high margin. The opening bids have great impact on the chance of completing a chain;
- The general strategy for reducing offers / increasing bids – many bargainers in our experiments, including progress_b, keen_s2, pmstub_s, keen_b and progress_b2, use rules that are reasonably predictable. This allows the opponents to compromise should they want to. However, they also give the opponents chances to exploit the information that they give away. Whether this strategy is successful depends on many factors, including the length of the chains, the opponents that they face and the type of problems that they were given;
- The strategy for predicting the opponent's bottom line – certain bargainers in our experiments, including smart_s4, mjtryh_b, attempt to guess the opponents' bottom lines;
- The strategy for deciding whether to accept an offer/bid when it provides room for profit – keen_b accepts any offer that is below its utility; keen_s2 delays for one round; other buyers/sellers, such as pmstub_s and mjtryh_b, use more sophisticated rules.

Experimental results so far revealed the success and weakness of certain strategies in these components (e.g. pmstub_s's strategy for reducing prices appears to be successful and progress_b2's criteria for accepting offers that are below utility was doubtful).

The search for good middlemen involves (a) finding the right strategies for each component, (b) finding the right combinations of components for each buyer/seller, and (c) finding the right combination of buyers and sellers for each middleman. Results in the experiments show that mediocre buyers/sellers can form successful middlemen; for example, (progress_b2, pmstub_s). The search space for middlemen is large. If one can identify all the major components in the game, and list a sufficient number of sensible strategies for each component, then we have defined a search space of useful middlemen. This gives us a chance to explore the search space systematically, e.g. by playing all the buyer-seller combinations against each other. It may also be possible to develop adaptive systems which react to the current population in the game.

This research is part of a larger project with business motivations. Many extensions of CBG(C) are worth looking into. Following are some examples:

- In the experiments conducted, the middlemen took the time constraints directly from the end-seller and end-buyer. For simplicity, the buyer and the seller of a middleman did not exchange and use both the days-to-sell and days-to-buy information. In reality, middlemen may have their individual time and other constraints.

- Having studied the components in the bargaining agents, we are now in a position to build systems to evolve (as opposed to hard-wire) agents for the above bargaining problem, as proposed in [1][3].
- Each seller (buyer) can negotiate with more than one buyer (seller) at any time, but with limited bandwidth – in this case, it has to decide on when to talk to which buyer (seller).
- Following the above point, the seller (buyer) may be given *n* days to sell/buy *m* pieces of goods – in this case, it has to decide what to sell to which buyer (seller) next, and when to stop negotiating with which buyer (seller).
- Past experience and market information (such as the average profit and the wealth of each agent) could be included in the model.

Acknowledgements

The author has benefited from input by students at University of Essex who have participated in the negotiation game, in particularly Robert Stacey, David Jacob, Philip Stubbings, Michael Tryhorn, Panagiotis Sourtzinos, Russell Briggs, Benjamin Longdon, Shaikh Saqib, Christopher Sharpe, Panagiotis Zoitos, Qosai Samara, Simon Powers, James Dunmore and Procopios Savvides. The authors have also benefited from discussions with Tim Gosling, Ken Binmore, Sheri Markose, Abhinay Muthoo, Mathias Kern and Chris Voudouris.

Bibliography

[1] Axelrod, R., The evolution of strategies in the iterated prisoner's dilemma, in Davis, L. (ed.), Genetic algorithms and simulated annealing, Research notes in AI, Pitman/Morgan Kaufmann, 1987, 32-41

[2] Binmore, K., Game theory and the social contract Vol 1: playing fair, MIT Press, 1994

[3] Binmore, K., Piccione, M. & Samuelson, L., Evolutionary stability in alternating-offers bargaining games, ELSE working papers 41, ESRC Centre on Economics Learning and Social Evolution, 1996

[4] Binmore, K., Game theory and the social contract Vol 2: just playing, MIT Press, 1998

[5] Durfee, E.H., Distributed problem solving and planning, in Weiss, G. (ed), Multiagent systems, a modern approach to distributed artificial intelligence, MIT Press, 1999, 121-164

[6] Freuder, E.C. & Mackworth, A., (ed.), Constraint-based reasoning, MIT Press, 1994

[7] Goldberg, D.E., Genetic algorithms in search, optimization, and machine learning, Reading, MA, Addison-Wesley Pub. Co., Inc., 1989

[8] Holland, J.H., Adaptation in natural and artificial systems, University of Michigan press, Ann Arbor, MI, 1975

[9] Jennings, N.R. and Wooldridge, M. (ed.), Agent Technology: Fundations, Applications and Markets, Springer-Verlag, 1998

[10] Jennings, N.R., On agent-based software engineering, Artificial Intelligence 117 (2), 2000, 277-296

[11] Jennings, N.R., Faratin, P., Lomuscio, A.R., Parsons, S., Sierra, C. & Wooldridge, M., Automated negotiation: prospects, methods and challenges, International Journal of Group Decision and Negotiation 10 (2), 2001, 199-215

[12] Koza, J.R., Genetic Programming: on the programming of computers by means of natural selection, MIT Press, Cambridge, MA, 1992

[13] Lottaz, C., Sam-Haroud, D, Falting, B. & Smith, I., Constraint techniques for collaborative design, Proc., IEEE 10th International Conference on Tools with Artificial Intelligence (ICTAI'98), Taiwan, November 1998, 34-41

[14] Muthoo, A., Bargaining theory with applications, Cambridge University Press, 1999

[15] Noriega, P. & Sierra, C. (ed.), Agent mediated electronic commerce, Lecture Notes in Artificial Intelligence, Vol.1571, Springer-Verlag, 1999

[16] Prosser, P., Distributed asynchronous scheduling, PhD thesis, Department of Computer Science, University of Strathclyde, 1990

[17] Tel, G., Distributed control algorithms for AI, in Weiss, G. (ed), Multiagent systems, a modern approach to distributed artificial intelligence, MIT Press, 1999, 539-580

[18] Tsang, E.P.K., Foundations of Constraint Satisfaction, Academic Press, 1993

[19] Tsang, E.P.K. & Gosling, T., Simple Constrained Bargaining Game, Proceedings, Distributed Constraint Satisfaction Workshop, First International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS-2002), Bologna, Italy, July 15-19, 2002

[20] Vulkan, N., Equilibria for automated negotiations, Games and Economic Behaviour, Vol.35, 2001., 339-388

[21] Vulkan, N., Economic implications of agent technology and e-commerce, The Economic Journal, Vol.1, 1999, 1-9

[22] Weiss, G. (ed), Multiagent systems, a modern approach to distributed artificial intelligence, MIT Press, 1999

[23] Wooldridge, M. and Jennings, N. R. (1995), Intelligent agents: theory and practice, Knowledge Engineering Review, 10(2), 1995, 115-152

[24] Yokoo, M., Asynchronous weak-commitment search for solving distributed constraint satisfaction problems, Proc., First International Conference on Principles and Practice of Constraint Programming (CP95), Lecture Notes in Computer Science 976, Springer-Verlag, 1995, 88-102

[25] Yokoo, M. & Ishida, T., Search algorithms for agents, in Weiss, G. (ed), Multiagent systems, a modern approach to distributed artificial intelligence, MIT Press, 1999, 165-199

[26] The Chain Constrained Bargaining Game, University of Essex, http://cswww.essex.ac.uk/CSP/bargain