Heuristics for Co-opetition in Agent Coalition Formation

Kevin Westwood
Utah State University
kevwestwood@cc.usu.edu

Vicki H. Allan
Utah State University
vicki.Allan@usu.edu

ABSTRACT

Coalitions are often required for multi-agent collaboration. In this paper, we consider tasks that can only be completed with the combined efforts of multiple agents using approaches which are both cooperative and competitive. Often agents forming coalitions determine optimal coalitions by looking at all possibilities. This requires an exponential algorithm and is not feasible when the number of agents and tasks is large. We propose agents use a two step process of first determining the task and then the agents that will help complete the task. We describe polynomial time heuristics for each decision. We measure four different agent types using the described heuristics. We conclude that if agents only consider their own individual gain, the potential profit of the agents and the potential throughput of the system will not be realized.

1 Introduction

Co-opetition is a phrase coined by business professors Brandenburger and Nalebuff (1996), to emphasize the need to consider both competitive and cooperative strategies. Recently there has been considerable research in the area of multi-agent coalitions (Lau and Lim, 2003) (Lau and Zhang, 2003)(Belmonte, et. al, 2004) (Li and Soh, 2004) (Lau and Zhang, 2003) (Blankenburg, Klusch, and Shehory, 2003) (Kraus, Shehory, and Taase, 2004). Multi-agent coalitions have been applied to many areas including manufacturing for optimizing production (Dang, et.al., 2003), information searching in heterogeneous databases (Klusch and Shehory, 1996), and, most recently, the area of e-commerce (Li, et.al., 2003).

A *coalition* is a set of agents that work together to achieve a mutually beneficial goal (Klusch and Shehory, 1996). The formation of a collation is often required because an agent is unable to complete a task alone. However, there may be other motivating forces, such as increasing its utility or completing the task by a deadline.

In the request for proposal (RFP) domain (Kraus, Shehory, and Taase, 2003), a collection of agents is challenged to complete specific tasks, each of which can be divided into subtasks. In this format, the task requirements and utility are specified and requests for proposals from agents are sought.

In the scenario considered here, no single agent possesses all the skills necessary to complete a task, and so coalitions must be formed. The agents attempt to form coalitions that will be able to complete tasks in a way which maximizes the payoff received. The agents are *individually rational* and will not join a coalition unless participating in the coalition is better than not joining a coalition at all. However, agents use heuristics to facilitate accepting less than optimal personal benefit if the global utility can be increased – hence the term, co-opetition.

2 Skilled Request for Proposal Domain

We use the term *Skilled Request For Proposal* (SRFP) to denote a domain in which levels of skill play an important role. In our domain, a requester provides a set of tasks $\mathcal{T} = \{\mathcal{T}_1 \ldots \mathcal{T}_n\}$ to be completed. Task i is divided into a list of subtasks $\mathcal{T}_i = (t_{i1} \ldots t_{im})$. Each subtask t_{ij} is associated with a skill-level pair $(\mathcal{ts}_{ij}, \mathcal{tl}_{ij})$ where a task skill, $\mathcal{ts}$, represents a skill taken from a skill set $\mathcal{S}$, i.e., $\mathcal{ts} \in \mathcal{S} = \{s_1 \ldots s_k\}$ and $\mathcal{tl}$ represents a specific level of skill, $\mathcal{tl} \in \{1 \ldots 10\}$. A set of service agents $\mathcal{A} = \{A_1 \ldots A_p\}$ exists such that each agent k has an associated fee, f_k, and agent skill-level pair $(\mathcal{as}_k, \mathcal{al}_k)$. The payoff value associated with a task is $V(T_i)$.

A coalition $\mathcal{C}_i = \langle \mathcal{T}_i, C_i, \mathcal{P}_i \rangle$ consists a task vector $\mathcal{T}_i = (t_{i1} \ldots t_{ij} \ldots t_{im})$ for task i, consisting of each of the subtasks (indicated by *j*) required to perform the task, a vector of agents comprising the coalition for task i $C_i=(C_{i1}\ldots C_{ij}\ldots C_{im})$ (where $C_{ij}\in \mathcal{A}$, and the elements of the tuple are ordered to indicate C_{ij} performs subtask t_{ij}), and a payment vector $\mathcal{P}_i = (p_{i1}\ldots p_{ij}\ldots p_{im})$ in which $p_{ij} \geq f_{ij}$ for all agents C_{ij} with skill-level (as_{ij},al_{ij}) and fee f_{ij} . Since the system is pareto optimal, all utility is distributed, i.e., $\sum_j p_{ij}=V(T_i)$. Note that, f_{ij} represents the actual cost associated with agent i performing the j^{th} subtask. A rational agent will not consent to performing a subtask for less than it's actual cost.

To facilitate the discussion, we introduce some terminology. The task $T_i = (t_{i1}\ldots t_{im})$ is *satisfied* by the coalition consisting of agents $C_i=(C_{i1} \ldots C_{id} \ldots C_{im})$ $(C_{id}\in \mathcal{A})$ if each agent d of the coalition (C_{id}) has associated skill-level pair (as_{id},al_{id}) such that $as_{id} = ts_{id}$ and $al_{id} \geq tl_{id}$. In other words, a coalition satisfies the needs of a task if for each subtask, there is an agent in the coalition with the required skill at a level equal to or greater than the level specified for the subtask. The difference $al_{id} - tl_{id}$ is termed the *underutilization* of the d^{th} agent in performing task i.

We term a task *viable* if there are available agents to satisfy the requirements of the task. We say an agent *supports* a subtask if the agent can perform an (as of yet) unassigned subtask. A level equal to or greater than the required level is termed an *acceptable level.* The level of skill which is actually required (due to possible missing skill-level pairs in the available task set) is called the *effective level.* Note, the effective level is the minimum of all acceptable levels (for a skill) in the current agent population.

For a particular skill, the base fees are a non-decreasing function of level and are publicly known. In our domain, while the specific fee associated with an agent is peculiar to the agent, the set of fees for a given skill-level pair is taken from a normal distribution with known mean and standard deviation.

3 Previous Work

Lau and Zhang (2003) explore the complexity of exhaustive methods for coalition formation.

The auction protocol proposed by Kraus, Shehory, and Taase (2003) is conducted by a central manager. Agents are randomly ordered at the beginning of each round r = {0,1,2…}. In each round, the agents are asked (in turn) whether they want to respond to previous proposals or initiate a proposal of their own. Each agent gets exactly one turn per round. After every agent has completed its turn in a round, for each satisfied proposal, the associated task and the agents required to complete the task are removed from the auction. At the end of a round, any unsatisfied proposals are discarded. The payoff associated with a task is decreased by a fraction of its original value with each round. Thus, for a task completing in round r with *discount factor* δ, the payment value is $V(T_i)^* \delta^r$. An initiating agent may only propose to agents which follow it in the non-cyclic agent ordering. This limit placed on who can be involved in a proposal makes the protocol unrealistic in modeling real world coalition formation. This limit is not present in our system. Our work also differs from their model in that there is no discount factor. The only motivation for forming coalitions early in our domain is that there are a greater number of tasks and agents from which to select.

In the domain proposed by Kraus, Shehory, and Taase (2003), each subtask is completely unique from all other subtasks encountered. A boolean function $\Phi(A_k, t_{ij})$ determines whether agent A_k can perform subtask t_{ij}. An agent has a 40% chance of having the skills required to complete a "normal task" and a 15% chance of having the skills to complete a "specialized task". This skill structure is limited in its ability to model realistic skill structures as there is only a rough concept of supply and demand on an individual skill basis. There is no concept of using a more highly skilled agent to do a less demanding task. There is also no concept of paying more for a higher level of skill. Kraus, et.al., consider incomplete information, but because averages are known, incomplete information provides merely random changes in the overall behavior, increasing the variance but not changing the cumulative behavior. Compromise is introduced in (Kraus, Shehory, and Taase, 2004) to increase the speed at which coalitions can be formed.

Gerber and Klusch (2003) worry about the ability to create stable coalitions in a dynamic environment where it is not reasonable to compute kernel-stable coalitions each time the state changes due to the high complexity. In addition, an agent's reputation for completing a task (reliability) enters into coalition formation decisions. Interestingly, an agent who attempts to form a coalition (which is later rejected) is penalized in terms of its reliability rating if a substitute coalition can not be formed. Thus, to protect its reliability rating, an agent who has received commitments from some agents must attempt to utilize the partial coalition in proposing a new task. Potentially overlapping coalitions are allowed and must be evaluated, increasing the complexity of heuristics that require coalitions to be disjoint. Similarly, Caillou, Aknine, and Pinson (2002) are concerned with dynamic restructuring of coalitions.

Scully, Madden, and Lyons (2004) use a skill vector that describes speed, and reliability. We propose a combined skill-level pair. Blankenburg, Klusch, and Shehory (2003) introduce the idea forming coalitions when there is a fuzzy knowledge of various information. An agent's evaluation of the coalition is uncertain, and agents can be (fuzzy) members of multiple coalitions to various degrees.

Trust based coalition formation as discussed in (Blankenburg, et.al., 2005) to deal with agents who deliberately misrepresent their costs and coalition valuation to increase their own profits. Belmonte et.al. (2004) allow agents to perform more than one task at a time, and an agent can break an initial agreement by paying a transfer cost. To simplify the tests and comparisons in the model, in the SRFP each agent has a single skill $\mathcal{as}$ in the skill set $\mathcal{S} = \{s_1 \ldots s_k\}$, but multiple skills could be modeled by storing a set of skills for each agent.

4 Auctioning Protocol

The coalition protocol described in Kraus, Shehory, and Taase (2003) is used as a simplification of a parallel model. We employ a similar protocol, consisting of service agents and a central manager that acts as an auctioneer and coalition manager. Service agents and tasks can join the auction at any time. For simplicity, tasks and agents are not allowed to leave the auction.

The auction is a variation of an English auction. The manager has knowledge of a set of tasks $T = \{T_1,\ldots,T_n\}$. Each task has a value $V(T_i)$ that will be paid when the task is complete. The auctioneer posts all tasks and their associated value to be paid. The auction consists of rounds $\{0,\ldots,r_z\}$ that will continue until all tasks are auctioned off, no remaining task is viable, or the time allotted expires. In order to focus on the ability of the heuristics to aid coalition formation, no discount factor is applied.

All negotiation is done through the manager. At the beginning of the auction, the manager randomly orders all agents so as not to give any preference to particular agents. One by one, each agent can either propose a coalition to perform a task or accept a proposal another agent has already made. If agent k chooses to make an offer, the agent is termed the *initiating agent* for the coalition. An agent who chooses to accept a previous offer is termed a *participating agent*. A proposal specifies the task to be done and the set of agents to complete the task. Profits are divided proportionally. After agent k has decided whether to propose a task or accept a previous proposal, the turn passes to agent k+1 (mod p) (who decides to propose or accept a previous offer). All agents accept or reject the proposal only when it is their turn in the round.

Consider the life a proposal initiated by agent k. All agents of the proposed coalition are informed of the proposal at their turn. They can accept or reject. If any agent rejects, when the turn passes to agent k in the next round (after all other agents have had a turn), the agents who agreed to the proposal are informed that the proposal failed. Agent k is allowed to make another proposal, re-propose the same proposal, or accept a previous proposal. If all the agents accept the proposal, when the turn passes to agent k, the task is awarded to the participating agents, and the task and participating agents are removed from the auction.

When an agent *i* is deciding between accepting one of the current offers or making an offer of its own, it has no knowledge of proposed coalitions which do not involve *i* nor does it know how many agents have accepted or rejected the current proposals. This lack of current information reflects the state of information in a parallel proposal environment in which acceptances happen simultaneously. Since agents and tasks are removed from the auction when the initiating agent has a turn in the following round, the information is updated with predictable delay. For the current offers, the task and the net gain (given the proposed coalition) is known.

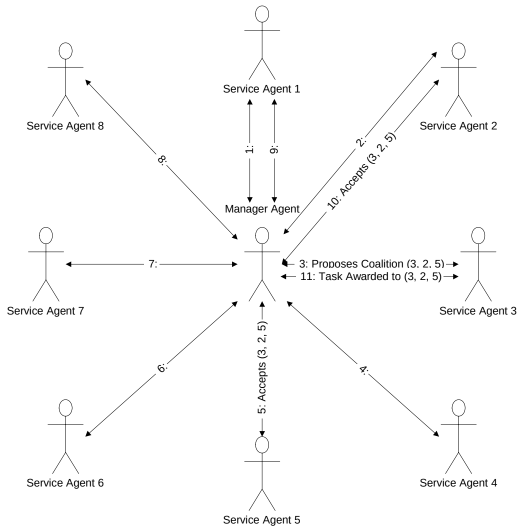

Figure 1: Auction Process

When a task is completed, each agent k is paid their fee f_k plus a portion of the *net gain* of the coalition.

The net gain is calculated as $\mathcal{N}_i = V(T_i) - actualCost_i$, where $actualCost_i = \sum_{j=1}^{m} f_{ij}$. In other words, the net gain is the amount that is left over from the value of the task after the agents are paid their advertised fees. Net gain is divided among the agents. Historically, the portion paid to each agent is divided equally, proportionally according to the agent's individual cost, or differentially, based on demand for a particular skill. In our model, we divide the net gain proportionally. Thus, agent i receives $f_{ij} + (f_{ij}/actualCost_i) * \mathcal{N}_i$.

5 Agent Algorithms

In many coalition formation algorithms, proposing agents calculate all possible coalitions to complete all tasks, an exponential algorithm. Because this quickly becomes unreasonable for even moderately sized problems, we divide the problem into two components and propose the following polynomial time heuristics for each component. In the first component, the agent selects a task to complete. In the second component, the agent selects the set of agents for the coalition to complete the task.

In this paper, we explore four task selection heuristics and two coalition selection heuristics for the proposing agents.

5.1 Task Selection Heuristic

The task selection heuristics are as follows:

1) **Individual Profit Task Selection:** An agent selects a task that will maximize its own profit. For each task, the agent first computes the expected net gain by subtracting the average fee of the effective level of each required skill. Thus, knowledge of the available agents is used to more accurately predict the agents' fees for completing a task. Tasks are ranked by comparing the agent's individual utility in each possible task. The task which yields the maximum individual utility is selected.
2) **Global Profit Task Selection:** An agent selects a task that will maximize global profit. For each task, the agent computes the expected net gain by subtracting the average fee of the effective level of required skill. The task with the maximum net gain is selected.
3) **Best Fit Task Selection:** An agent picks the task for which the sum of the agent underutilization is minimized. The motivation is that when the resources are used wisely future coalitions will be less likely to face resource constraints. Ties are broken using maximum individual profit.
4) **Co-opetitive Selection:** Select the best fit task as long as it is within P% of the maximum individual profit.

5.2 Coalition Selection Heuristic

The coalition selection heuristics are as follows:

1) **Best Profit Coalition Selection:** For each subtask, select the supporting agent with the smallest fee. Since actual costs differ from the expected costs, this is different than merely finding the best fit.
2) **Best Fit Coalition Selection:** For each subtask, select the supporting agent with the smallest underutilization. In general, a closer match of level will conserve resources while getting the best expected cost.

5.3 Acceptance Policy

Agents must have a policy to decide between (a) accepting one of the various proposals or (b) being an initiating agent. Our agents compare existing proposals with the benefits of the proposal they would make (using their task selection heuristic) and select the proposal most closely fitting their criteria. The *contending proposal* for agent i is the "best" proposal (of those proposed to agent i)

Accepting the contending proposal may give the agent a better chance of joining a coalition, particularly when the number of subtasks is small, as an agent knows that at least two agents (itself and the initiator) will accept the proposal. Since an agent i has no information about other proposals (not involving i) and no knowledge of how many agents have accepted the current proposal, the advantage of accepting a current proposal over initiating a proposal is not as great as one might think.

If the contending proposal is within C% of the proposal the agent would initiate, the contending proposal is accepted. Otherwise the proposal is rejected. Several different values for the compromising ratio, C, are evaluated in our experiment. The idea is that with compromising (not holding out for the best possible coalition) coalitions can be formed quicker, resulting in higher utility (attributed to more opportunity due to a richer pool of tasks and agents) (Kraus, Shehory, and Taase, 2004).

6 Results

Results represent the average of tests run 5000 times. Each trial involves 60 agents and 20 tasks (3 subtasks each). Each subtask has a randomly generated agent skill-level pair. Skills vary from 1 to 10 and skill levels vary from 1 to 10. Agent skills and skill levels are matched to fit the tasks. For example, if subtask 1 from Task 1 has an agent skill-level pair of (3,5), an agent is generated with skill 3 level 5. This setup gives an environment where most tasks can be assigned and patterns among the agents can easily be discovered.

For each skill, skill level 1 has a base fee of 5 and skill level 10 has a base fee of 50. Actual fees are assigned using a normal distribution with a median value of the base fee and a standard deviation of 2.5 (half the difference from one skill level to the next). In our tests, payoffs of the generated tasks are assigned as a random percentage (100-200%) of the expected cost of completing the task (based on expected costs of the skill-level required by the subtasks).

Four different agent types are considered, corresponding to the task selection algorithms: Local Profit, Global Profit, Best Fit and Co-opetitive. Local Profit and Global Profit agents use the best profit coalition selection algorithm. Best Fit agents use the best fit coalition selection algorithm. The Co-opetitive agents use the best fit coalition selection algorithm if the best fit task's potential individual profit is greater than 90% of the best profit task's potential individual profit. Otherwise, the Co-opetitive agent uses the best profit coalition selection algorithm. The Co-opetitive agent is interested in helping the throughput of the system, if individual benefit is considered. Thus, a high P% was chosen. For our tests, we use a P value of 90.

All profits are shown as Profit/Optimal profit or the *Profit Ratio*. This representation is important as it normalizes profits from one agent to another. Consider agent A that has an optimal profit of 100 and agent B that has an optimal profit of 10. If agent A receives 50 and agent B receives 5, each agent's performance is equal. Raw results tend to exhibit no clear pattern of behavior as the achievable results are so variable.

Individual optimal profits are determined by exploring all possibilities for the specific agent. Global optimal profits are determined by using a hill-climbing algorithm.

6.1 Compromising Ratio

We need to know what degree of compromise is affective in determining an acceptance policy. We hypothesized that the more competition an agent has or the fewer the number of tasks that are available to complete, the more the agent would want to compromise in accepting other agents' proposals. If the agent does not compromise, it risks the chance of not joining a coalition.

To determine the compromising ratio, we conduct the following experiment. The number of tasks is fixed at 20 with 3 subtasks each. The number of agents fluctuates from 20 to 180. Using an equal distribution of each agent type, we track the individual profit and the number of tasks completed by each agent type.

Figure 2 shows the profit ratio for the Individual Profit agent. On average, having a high compromising ratio maximizes the Individual Profit agent's profit. This same pattern is seen in other agent types as well. The number of tasks that are completed is maximized by Best Fit agents having a high compromising ratio. Because of these results, we fix the compromising ratio at 90% for all agent types in the following tests.

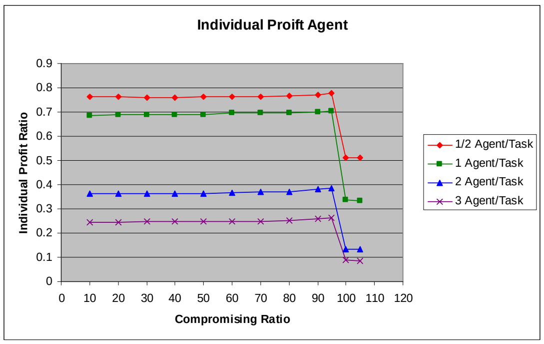

Figure 2: Individual Profit Agent and Compromising Ratio

6.2 Individual Agents

Each agent type has a different objective. The Individual Profit agent desires to increase it's own individual profit. The Global Profit agent desires to maximize the profit of all agents in the system. The Best Fit agent desires to maximize the throughput of the system. The Co-opetitive agent desires to increase the throughput of the system as long as it is still benefiting individually.

We explore the impact each agent type has on the system. We setup tests that gradually increase the percent of a single agent type until it is the only agent type left in the system. Note that the other agents' types in the system exist in equal proportions.

Figure 3 shows the individual profit ratio for the specific agent type as its percent in the system increases. It can be seen that the Individual Profit agent does very well when less than 70% of the agents are Individual Profit Agents. As its percentage in the system increases, their profit decreases. The primary cause of this decrease is that there are fewer tasks completed. See Figure 4.

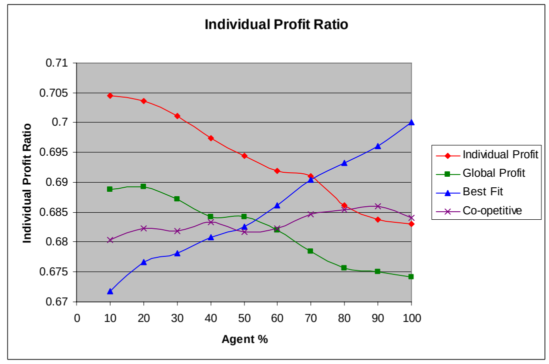

Figure 3: Individual Profit Ratio

The Best Fit agents have the opposite effect. As the percentage of Best Fit agents increase, the number of tasks completed and the individual profit for the Best Fit agents increase as well.

The Co-opetitive agents' profit increases slightly, but is fairly steady. With less than 50% of the same agent type, the Co-opetitive agent has a higher individual profit than the Best Fit agent. The throughput of the Co-opetitive agent is fairly steady as well. As the percentage of Co-opetitive agents increases, the average number of tasks completed stays about the same.

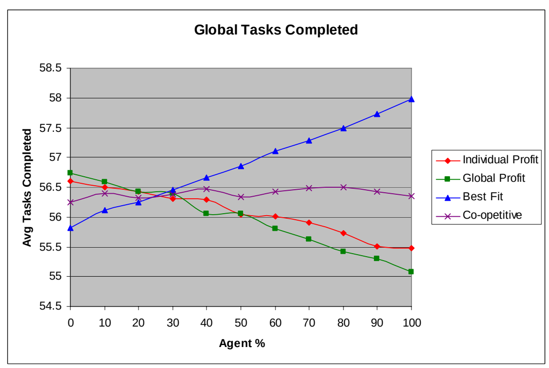

Figure 4: Global Tasks Completed

Figure 5 shows the global profit ratio of the system. This is calculated as the total profit of all agents divided by maximum possible profit of all agents. The Global Profit agents' objective was to maximize the global profit and in doing so increase the throughput of the system. It did not do as well as hypothesized.

The Global Profit is dependent on the number of tasks that are completed. The Best Fit agent maximizes the global profit as its percentage increases past 25%. Both profit agents do not do very well. As their percentage increases, the global profit decreases. The Co-opetitive agent does a little better than the profit agents, but not nearly as well as the Best Fit agent.

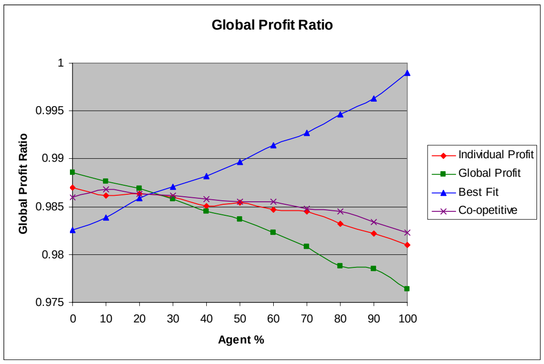

Figure 5: Global Profit Ratio

6.3 Standard Deviation

Variability of agent costs from the base cost has a major impact on how agent types compare to one another. As the variability increases, the need for Best Fit and Co-opetitive agents becomes more evident. Compare Figure 6 to Figure 3. Figure 6 shows the individual profit ratio with agent costs having a standard deviation of 5. All previous figures have a deviation of 2.5.

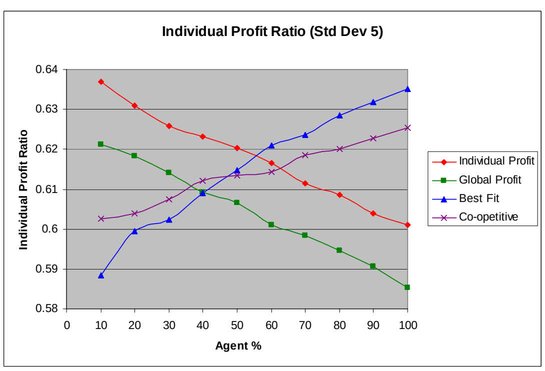

Figure 6: Individual Profit Ratio (Std dev 5)

The Best Fit and Co-opetitive agents cross the Individual Profit agent curve much earlier (55% and 65% respectively). They cross at 70% and 80% with a standard deviation of 2.5. The slope of the Co-opetitive agent increases and comes closer to that of the Best Fit agents.

Compare Figure 7 to Figure 4. At 100% of the same agent type, when the standard deviation is increased from 2.5 to 5, the difference between the number of tasks completed by the Best Fit and Global Profit agents increases from 3 to 6. The slope of the Co-opetitive agent curve increases and is much closer to the Best Fit agent.

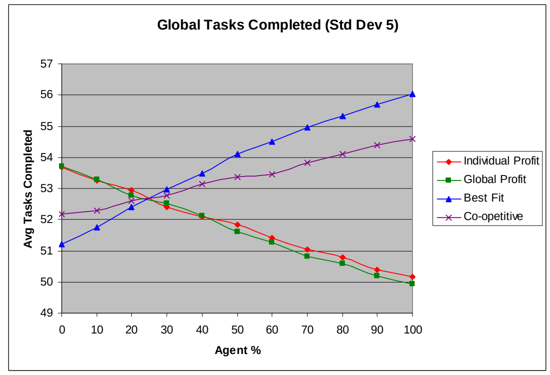

Figure 7: Global Tasks Completed (Std Dev 5)

Compare Figure 8 to Figure 5. At 100%, the spread between the Best Fit and Profit agents' profit ratio is increased from .025 (std dev 2.5) to .045 (std dev 5). The slope of the Co-opetitive agents' global profit ratio is increased and is much closer to that of the Best Fit agent.

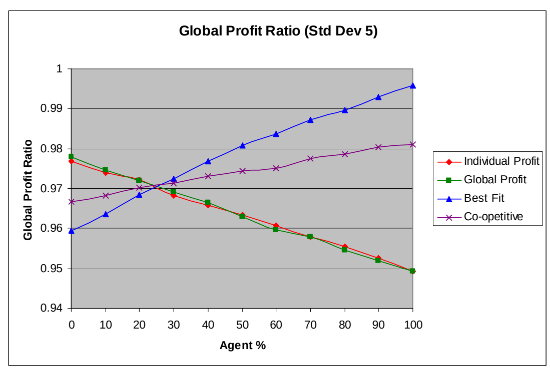

Figure 8: Global Profit (Std Dev of 5)

7 Future Work

Tasks could be made time sensitive. Kraus, Shehory, and Taase (2003) describe a discount factor that decreases the task pay over time. By applying a similar discount factor, agents could be encouraged to join coalitions more quickly. Agents could adjust their compromising ratio according to the discount factor of tasks.

Soh and Li (2004) evaluate their coalition forming strategy by measuring how difficult it is for agents to form coalitions. In our domain, this could be measured by the number of proposals made and how many eventually fail. This information could be used to improve agent decision making skills, so that fewer proposals fail.

8 References

Belmonte, M., Conejo, R., Perez-de-la-Cruz, J.L., Triguero, F., A Robust Deception-Free Coalition Formation Model, SAC '04, March 14-17, 2004, Nicosia, Cyprus

Blankenburg, B., Klusch, M, and Shehory, O, Fuzzy Kernel-Stable Coalitions Between Rational Agents. AAMAS'03, Melboure, Austrailia, July 14-18, 2003.

Blankenburg, B, Dash, R.K., Ramchurn, S.D., Klusch, M, Jennings, N.R., Trusted Kernel-Based Coalition Formation, AAMAS'05, July 25-29, 2005, Utrecht, Netherlands.

Brandenburger, A. M.; Nalebuff, B. J. Co-Opetition: the Game Theory That's Changing the Game of Business, Doubleday Publishing, New York, NY, U.S.A. 1996.

Caillou, P, Aknine, S, and Pinson, S., A Multi-Agent Method for Forming and Dynamic Restructuring of Pareto Optimal Coalitions, AAMAS'02, July 15-19, 2002, Bologna, Italy.

Dang, T., and Frankoviˇc, B., and Budinsk´av, I., Optimal Creation of Agent Coalitions for Manufacturing and Control. Computing and Informatics, Vol. 22, 2003, 1001–1031, V 2003-Jun-19

Gerber, A. and Klusch, M., Forming Dynamic Coalitions of Rational Agents by Use of the DCF-S Scheme. AAMAS '03, Melboure, Austrailia, July 14-18, 2003.

Klusch, M., and Shehory, O. A Polynomial Kernel-Oriented Coalition Formation Algorithm for Rational Information Agents, in Proceedings of ICMAS '96, 157-164, Kyoto Japan, 1996

Kraus, S., and Shehory, O., and Taase, G., Coalition Formation with Uncertain Heterogeneous Information. AAMAS '03, Melbourne, Austrailia, July 14-18, 2003.

Kraus, S., and Shehory, O., and Taase, G., The advantages of Compromising in Coalition Formation with Incomplete Information. AAMAS, pp. 588-595, Third International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 2 (AAMAS'04), 2004.

Lau, H., Lim, W., Multi-Agent Coalition Via Autonomous Price Negotiation in a Real-time Web Environment. Intelligent Agent Technology, 2003. IAT 2003. IEEE/WIC International Conference on 13-16 Oct. 2003 Page(s):580 – 583, 2003

Lau, H., Zhang, L., Task Allocation via Multi-Agent Coalition Formation: Taxonomy, Algorithms and Complexity. Proceedings of the 15th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'03), 2003

Li, C., and Rajan, U., and Chawla, S., and Sycara, Katia. Mechanisms for Coalition Formation and Cost Sharing in an Electronic Marketplace. Proceedings of the 5th International Conference on Electronic commerce. Pittsburgh, Pennsylvania, 2003

Li, X., Soh, L., Learning How to Plan and Instantiate a Plan in Multi-Agent Coalition Formation. Intelligent Agent Technology, 2004. (IAT 2004). Proceedings IEEE/WIC/ACM International Conference on 2004 Page(s):479 – 482, 2004

Scully, T., and Madden, M. G., and Lyons, G., Coalition Calculation in a Dynamic Agent Environment. Proceedings of the 21st International Conference on Machine Learning. Banff, Canada, 2004.

Soh, L-K, and Li, X. Adaptive, Confidence-Based Multiagent Negotiation Strategy AAMAS'04, July 19-23 2004, New York, USA.