Compromise Strategies for Constraint Agents

Eugene C. Freuder and Peggy S. Eaton
Computer Science Department
University of New Hampshire
Durham, New Hampshire 03824
ecf,pse@cs.unh.edu

Abstract

We describe several joint problem solving strategies for a team of multi-interest constraint-based reasoning agents. Agents yield in one area of conflict in order to gain concessions in another area. For example, in a distributed meeting scheduling problem, individuals with different preferences may be willing to give up the choice of the location for the meeting if allowed to choose the time of the meeting. Compromise benefits the overall solution quality by allowing each agent an opportunity to participate in the solution. We demonstrate the utility of this approach using a collection of agents collaborating on random graph coloring problems. We propose several simple metrics for evaluating solutions from the perspective of individual agents, and additional metrics for evaluating the solutions as compromises. Finally, we experimentally evaluate the performance of the strategies with respect to the metrics.

Introduction

Individuals may bring different priorities to a problem. How they can best work together to produce a compromise solution? This is, of course, an old issue in human affairs. Nowadays we envision computer agents grappling with this issue, perhaps representing our preferences. We address this issue here in the context of constraint satisfaction problems (CSPs), which are widely encountered in artificial intelligence.

We propose several simple strategies for joint problem solving that lead to compromise solutions. We propose several simple metrics for evaluating solutions from the perspective of individual agents, and additional metrics for evaluating the solutions as compromises. Finally, we experimentally evaluate the performance of the strategies with respect to the metrics.

CSPs are composed of variables, values and constraints. A solution assigns a value to each variable such that all the constraints, specifying acceptable value combinations, are satisfied. We provide a simple, initial model of preference where each agent has a ranking for the potential values for variables. We employ coloring problems here as a tested. Coloring problems require assigning colors to variables, where specified pairs of variables cannot have the same color. They model basic scheduling and resource allocation problems. Here we assume that each agent ranks the colors in order of preference.

We use this domain to begin an investigation of questions like: What is a good compromise strategy if maximizing the sum of the values assigned to the solution by the participants is less important than minimizing the disparity among those values? (This is a situation familiar to any parent who has had to provide treats for several siblings.)

We describe the compromise strategies, the individual metrics, and the compromise metrics in the following section. Next, we present our experiments, describing the experimental design and then graphing the behavior of the different strategies with respect to the different metrics. In the last section we briefly relate our work to previous work and suggestion directions for further research.

Compromise Strategies and Metrics

In this section we describe four joint problem solving strategies and evaluation metrics for constraint agents. The strategies we propose emphasise the emergence of a compromise solution among a team of agents. The strategies are useful when agents have competing goals but perceive an advantage to working together in a fair way and may be useful in application areas where user preferences play an important role, such as scheduling, intelligent user interfaces, and telecommunication.

Strategies

The joint problem solving strategies have the following characteristics:

- All group members participate in generating a problem solution.
- Agents are not guaranteed an individual optimal solution.
- The strategy doesn't unfairly favor any agent.
- There is no central controller monitoring the problem solving process.

- Agents consider only their own preferences when selecting a value to assign to a variable.

Turn taking strategy Turn-taking is a simple strategy where agents take turns assigning values to variables. During problem solving the agent whose turn it is sends a message to the team containing the variable-value assignment. Agents perform backtrack search for the solution together. Each agent knows when a conflict occurs and they know the agent responsible for the variable assignment. At that point, the agent originally assigning the variable a value chooses again. When a solution is found, the agents report their individual metrics. Of course, turn-taking doesn't guarantee an optimal solution for individual agents, but it does guarantee each an opportunity to make some variable assignments.

Average preference strategy The average preference strategy is an a priori computation of the average preference utility for the values of each variable. The value selected for assignment has the highest average preference utility. When a solution is found agents compute their metrics based upon their original set of preference utilities.

Concession strategy The concession strategy is a variation of turn-taking; agents making a variable assignment will concede their turn to another team member if the other agent has a higher preference utility for a variable-value assignment.

Lowest score strategy During search agents track their scores for the current labeling of problem. When a variable is to be assigned a value the agent with the lowest score chooses based upon its own preferences.

The average preference strategy may be appropriate in domains where agents are willing to exchange all their preferences. For example, agents in a course scheduling application represent the interests of teachers, students, and administrators; the scheduling preferences for each agent can be stated and shared prior to problem solving. An important goal in this domain is to achieve high solution qualities for the group while minimizing disparities among the agents.

A strategy that supports information hiding, such as turn-taking, may be important in situations where the agents are willing or required to work together to solve problems but because of privacy reasons they are only willing to share partial information during problem solving. For example, agents from different telecommunication companies may be willing to cooperate to solve a routing problem but would like to do so by maximizing their own preferences while cooperating with others. Compromise strategies may also be important to intelligent user interface agents, such as (Maes 1994), (Sycara & Zeng 1996), and (Marc Andreoli *et al.* 1995), where the primary purpose of the agent is representing the preferences and priorities of humans.

Metrics

We propose simple metrics to compare how well individual agents fare, how well the group performs when using each compromise strategy, and the efficiency of each strategy. The selection of an appropriate compromise strategy is dependent upon the type of group interaction desired and the type of measurements used. The sum and product metrics are similar to the metrics proposed by (Rosenschein & Zlotkin 1994) for evaluating 2-agent negotiation protocols. We consider the efficiency of the compromise strategies by comparing the number of constraint checks generated during problem solving.

Individual metrics The product and sum measures are an indicator of an agent's preference for a particular problem solution. The sum metric is the sum of the preference utilities for each variable-value assignment in the solution. The product metric is the sum of the log(preference utility of value) for each variable-value assignment in the solution.

Group Metrics

The compromise strategies are compared using the following metrics:

- median of individual metrics for each agent
- maximum solution quality of the group
- minimum solution quality of the group
- difference between maximum and minimum solution qualities (Max - Min)
- number of constraint checks

The solution quality for the above metrics can be computed by using either the product or sum measure of solution quality. The minimum sum and minimum product over the agents indicates the minimum solution quality generated by the agents on the team. The maximum and median metrics are used to determine which strategy returns the highest solution qualities for the team of agents. Teams of agents may also be concerned with the disparity among the solution qualities of the agents; (Max - Min) is a good indicator of disparity. A strategy where team members have a similar solution quality will have a low (Max - Min) score. The average number of constraint checks is an indicator of team problem solving efficiency of a particular strategy.

Experiments

We evaluate the utility of the proposed compromise strategies and metrics using solvable random coloring problems. Coloring problems are representative of scheduling and resource allocation problems. In these problems, colors must be assigned to variables so that related variables do not have the same color.

The experiments were run on a set of 100 random coloring problems with the following characteristics:

- number of variables = 30
- domain values (colors) for each variable = 6
- constraint tightness = 0.833
- constraint density = 0.1
- each problem is solvable
- team size = 3

We chose these problem parameters because the problems were easy to solve. We had similar results when we tested the turn taking strategy on sets of problems at other densities (0.4, 0.3, and 0.25). We are currently performing experiments with the other strategies at various densities and we are running experiments with teams of ten agents.

The agents each use identical search algorithms and CSP representations except for their preference vectors. The preference vectors were randomly assigned when the problem representation was created. The problem representation of each agent was augmented with a preference utility for each value associated with a variable. The preference utilities were assigned on a scale of 1 to the maximum domain size. For example, if the domain size is 6, the maximum preference value 6 can only be assigned to one of the domain values of that variable. The product and sum metrics are used to gauge the quality of the overall solution by combining preference utilities for the variable-value assignments that have been made.

Agent	Turns	Avg	Conc	Lowest
1	38	43	39	38
2	40	43	39	39
3	39	43	39	39

Individual Product Metric

Agent	Turns	Avg	Conc	Lowest
1	124	137	126	124
2	127	136	128	125
3	125	137	126	125

Individual Sum Metric

The compromise algorithms are based upon simple backtracking where the variables are lexically ordered. When a variable is selected for instantiation the agents run a value selection function to choose the value for the variable based upon the compromise strategy and backtrack proceeds.

The agents exchange messages containing their choice for a variable-value assignment. In the turn-taking, concession, and lowest-score strategies the agents do not know about the other agents preference values; the agents only know the values the other agents have chosen during the search process. A more sophisticated belief model might keep track of the other agent's selections and use that information as the basis of negotiation during later stages of backtracking.

The results reported in the tables below are for 97 of the 100 problems. The team of agents were able to find a compromise solution to 97 of the 100 problems using the different strategies. However, on 3 problems the agents had difficulty (very long processing times) finding a solution regardless of the strategy used by the agent. As an initial attempt to address this problem in our model we allowed the agents using the turn-taking strategy to restart problem solving after rotating their turn-taking order. Given this change the team was able to solve 2 of the 3 problems. This same technique was used for the concession and lowest-score strategy and proved useful because turn-taking is used when there is a tie in these strategies; the agents were able to find a solution to the same 2 of 3 problems. The average preference strategy uses the same set of preferences for all agents so the agents rotating positions does not affect problem solving. The team using the average preference strategy was unable to solve the three problems.

	Turns	Avg	Conc	Lowest
average	39	43	39	39
max	40	43	39	39
min	38	43	39	38
max-min	2	0	0	1

Group Product Metrics

	Turns	Avg	Conc	Lowest
average	125	137	126	125
max	127	137	128	125
min	124	136	126	124
max-min	3	1	2	1

Group Sum Metrics

Discussion

The graphs in figures 1-5 show solution qualities for each strategy and group metric. To select a compromise strategy for a team the designer must consider:

- lowest solution quality generated by an agent on the team
- median solution quality - the team score
- highest solution quality generated by an agent on the team
- disparity of scores among team members
- problem-solving performance - constraint checks
- information-hiding

Trade-offs are necessary; no one strategy wins in every performance area. A team requiring a low disparity among team members and information-hiding will select the concession strategy. While a team requiring

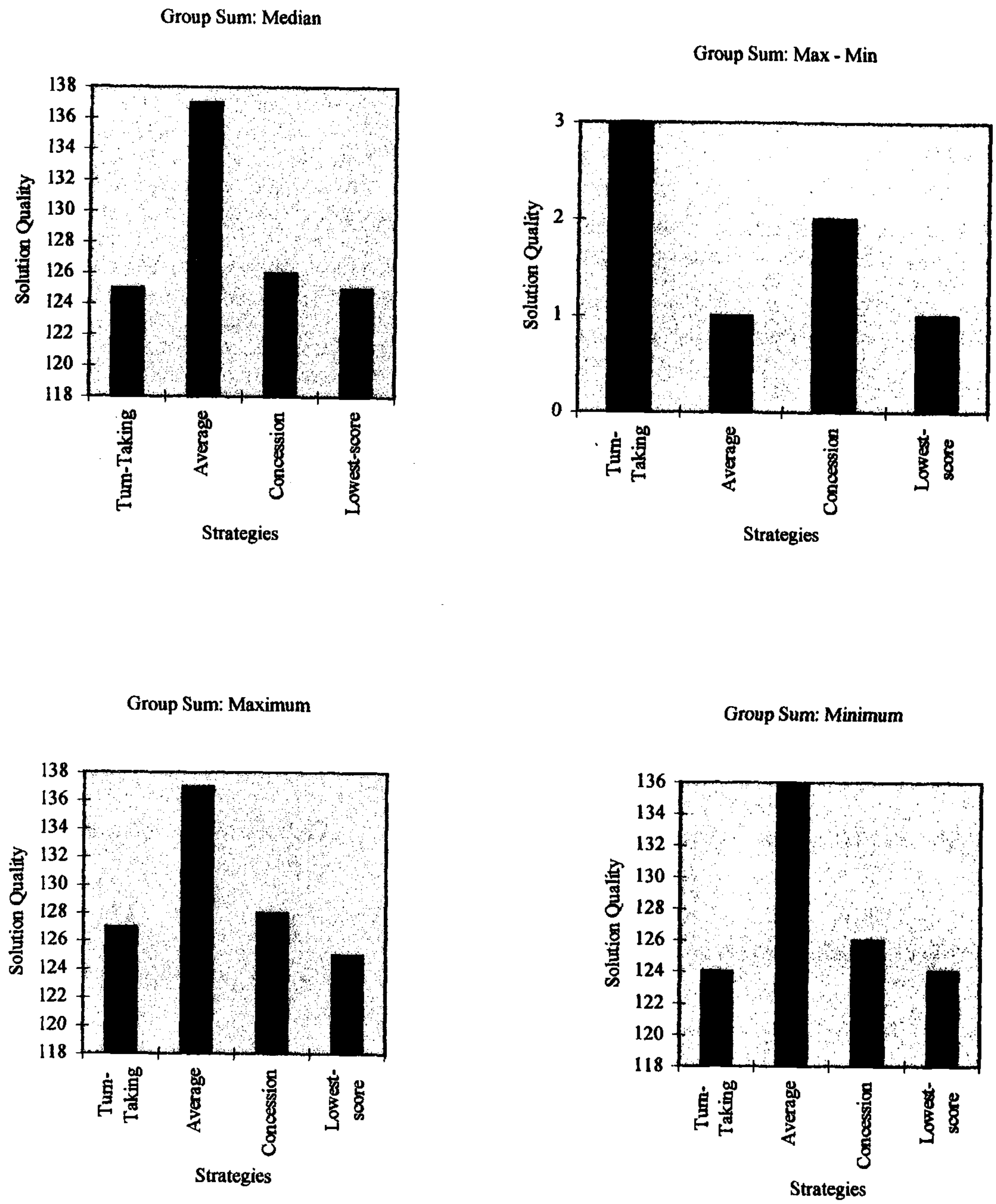

Figure 1 - Sum Metrics

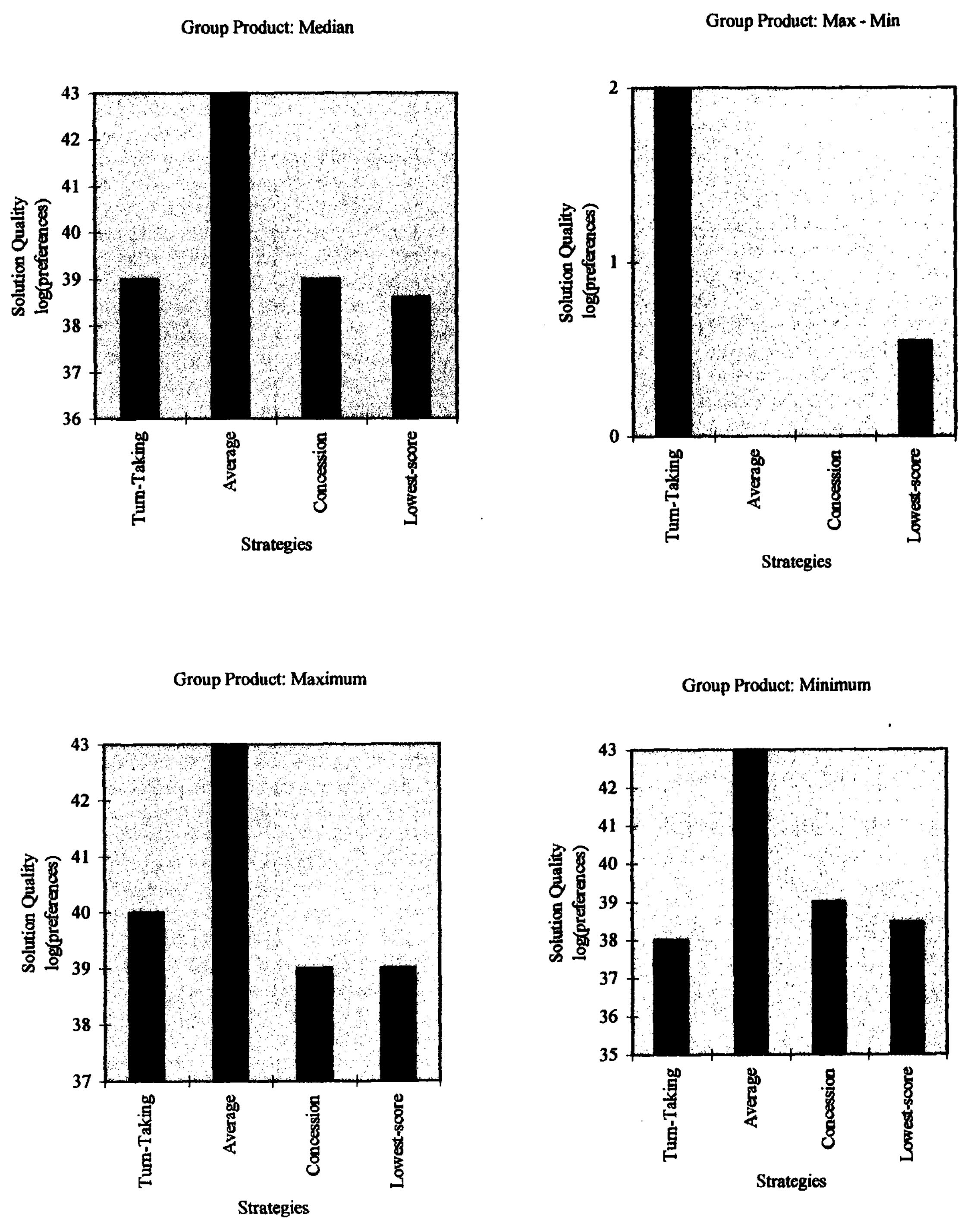

Figure 2 - Product Metrics

high problem-solving performance would choose turn-taking.

The average preference strategy provides high solution quality and low disparity among agents so performs very well as a compromise strategy but there are two issues that must be considered.

1. The strategy requires agents to exchange all preference utilities before beginning problem solving.
2. The strategy is not as efficient as other strategies.

The Max group metric identifies which strategy produced the highest average solution quality. The average preference metric performs best by both the sum and product measures; the concession, turn-taking, and lowest-score strategies are similar.

Max - min measures the difference in solution qualities over the set of agents, so a low score is best. This measure is useful when we are interested in everyone on the team being equally happy. The average preference strategy and the lowest-score strategy generate solutions in a way that minimizes the disparity of the scores among the agents. The greatest disparity occurs when the agents use the turn-taking strategy.

The table below shows the number of constraint checks performed by an agent during problem solving. The turn-taking strategy has the best problem-solving performance characteristics by this metric. However, the constraint check disparity is due to a few problems with many constraint checks. We plan to further investigate the performance characteristics.

	Turns	Avg	Conc	Lowest
average	315	324	324	324
median	331	544	2702	375
max	1104	11397	22926	2415
min	210	208	192	234

Agent Constraint Checks

Related Work and Conclusion

Related Work

Rosenschein and Zlotkin (1994) applied game theory to the design of interaction protocols for multiple agents. Our use preference utilities is similar to the use of utilities in worth oriented domains. Worth oriented domains are a class of domains where agents assign a worth to states of the world; 2-agent interaction protocols are evaluated in competitive worth oriented domains. We use evaluation metrics to measure both individual agent performance and group performance. Our metrics are different because of the way we compute the solution quality. Solution quality in (Rosenschein & Zlotkin 1994) compares the utilities of individual variables using both sum and product metrics. We compute the sum (product) across the solution for an individual agent and then find the median in the set of experiments for individual agents. The group solution is a comparison of the performance of the team metrics. The computation we use focuses on the quality of the complete solution rather than the quality of the pieces of the solution.

(Wellman 1994) uses a market economy model to design the interaction protocol of multiple agents solving distributed configuration design problems. Preferences in the producer/consumer example are represented by utility functions; consumer agents use the utility functions to decide whether or not to consume a product. The focus of the work is to provide a framework supporting decentralized multi-agent interactions.

(Liu & Sycara 1994) focus on handling overconstrained meeting scheduling problems using agent preferences. Agents agree or disagree to schedule a meeting using their own preferences but they do so given a global representation of the problem, so they can relax local constraints that may not be important given a particular meeting. Agents exchange meeting scheduling constraints to create a global representation of the problem. This global information exchange is similar to the way the agents in our system exchange preferences when computing a value ordering based upon average preferences.

Future Work

Areas of future work include:

- increasing the number of agents involved in problem solving,
- comparing strategies across other metrics, such as the distance from an optimal solution,
- modifying the turn taking strategy so the agent chooses which variable to instantiate as well as the value for the variable.

We are interested in extending the compromise strategies so agents can improve the solution by heuristic repair. The concession and lowest-score strategies may be modified so that the compromise process continues until consensus is reached. In this variation the agents compute the quality of a given a solution to the problem. If the solution quality is below a minimal threshold the agent proposes changing the solution; each agent may propose, agree, or disagree to solution changes but no changes can diminish the group solution quality. Although communication costs will increase in this scenario the individual solution quality obviously improves while maintaining a minimal group solution quality. Consensus is reached when no further changes are proposed by the agents.

Conclusion

The contributions of this work include the application of joint problem solving strategies among a group of constraint agents that have different preferences. We consider the meaning of a good group solution and propose simple metrics to evaluate the quality of the solution.

The turn-taking, average preference, concession, and lowest-score strategies provide a protocol for mutual yielding when self-interested agents must work together. In each case, individual agents benefit because their preferences are considered during joint problem-solving.

Acknowledgments: This material is based on work supported by the National Science Foundation under Grant No. IRI-9504316.

References

Eaton, P. S., and Freuder, E. C. 1996. Agent cooperation can compensate for agent ignorance in constraint satisfaction. In Tambe, M., and Gmytrasiewicz, P., eds., *Agent Modeling Papers from the 1996 AAAI Workshop*, 24–29. AAAI Press. Technical Report WS-96-02, Portland, OR.

Liu, J., and Sycara, K. P. 1994. Distributed meeting scheduling. In *Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society*. Atlanta, Georgia.

Maes, P. 1994. Agents that reduce work and information overload. *Communications of the ACM* 37(7):30–40.

Marc Andreoli, J.; Borghoff, U. M.; Pareschi, R.; and Schlichter, J. H. 1995. Constraint agents for the information age. *Journal of Universal Computer Science* 1(12):762–789.

Rosenschein, J. S., and Zlotkin, G. 1994. *Rules of Encounter: designing conventions for automated negotiation among computers.* Cambridge, Massachusetts: MIT Press.

Sycara, K., and Zeng, D. 1996. Coordination of multiple intelligent software agents. *International Journal of Cooperative Information Systems* 5(2/3):181–212.

Wellman, M. P. 1994. A computational market model for distributed configuration design. In *12th National Conference on Artificial Intelligence.* Seattle, Washington.

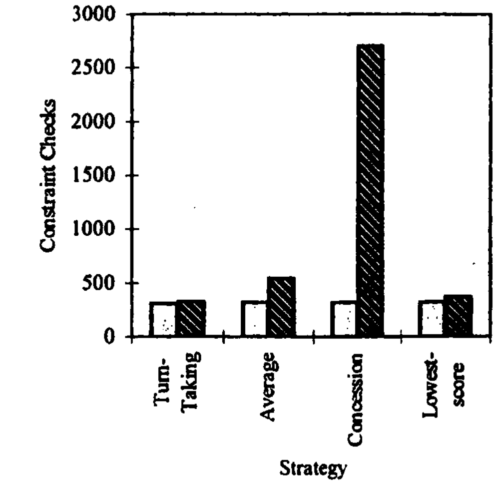

Figure 3: Agent Constraint Checks

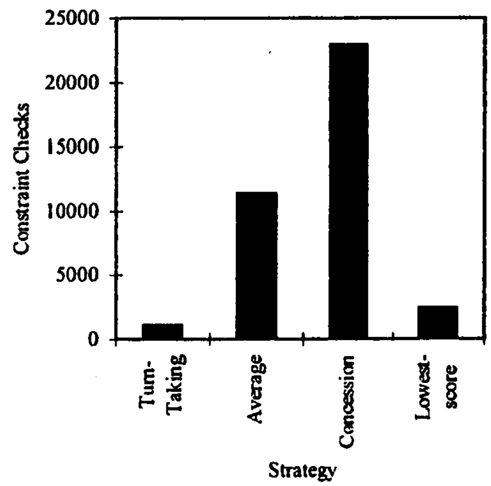

Figure 4: Agent Constraint Checks - Maximun

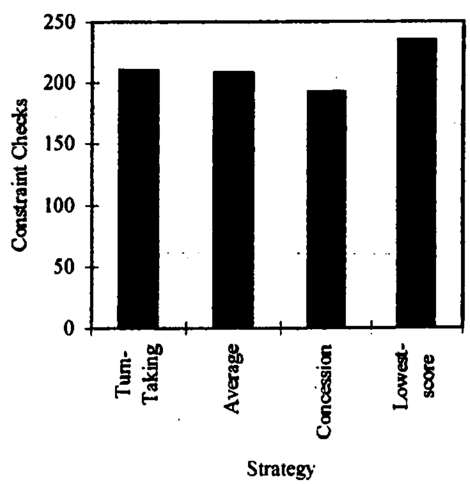

Figure 5: Agent Constraint Checks - Minimum