A Genetic Algorithm for Conflict Resolution in Concurrent Production Development

Yan Yuhong, Hu Yongtong, Liu Ping,

Zheng Danian*, Ma Changchao, Yang Shiyuan,

Department of Automation, Tsinghua University

*** Department of Precision Instruments, Tshinghua University**

Beijing, P. R. China, 100084

ABSTRACT

Current design of product involves the cooperation of many functions such as manufacturing, logistics, marketing, and sales. It can be represented as multiagent system with each agent representing as a kind of expertise [2]. Each agent has its own evaluation criteria and egoistic goals. The design process can be viewed as an optimization process of multiple participants and multiple criteria. In this paper, we use GA as a heuristic technique to search in large feasible solution space, and a tool to synthesize different counter-proposals from different agents. These counter-proposals are used in GA operations to speed up the evolutionary process. We also provide an experiment, in which the process of designing gear pump is simulated. The system structure and agent structure are based on our former work [1].

1. INTRODUCTION

A. Conflict in Concurrent Product Development

Concurrent Production Development (CPD) involves the cooperation of many functions such as manufacturing, assembling, logistics, marketing, and sales. With the cooperative work of experts from different domain, the flaws of the product are removed in the early stage of product design, so as to decrease the cost and increase the competition ability of the product.

The environment of a CPD system can be represented as multiagent system (MAS) with each agent representing a kind of expertise [2]. Each agent has its own evaluation criteria and egoistic goals. During their cooperation work, conflicts arise inevitably, for the agents tend to maximize their local benefits from their points of view. Although in some cases, the interests of agents are totally antagonistic, in many other cases, the agents have a integral advantage to achieve. The agents in the CPD system belong to the later category. They share a common goal, to design what is really best.

This Project is supported by 96' CIMS Foundation from China National CIMS-ERC

B. Previous Work on Negotiation

Negotiation is an important approach to resolve conflicts. However, modeling and supporting negotiations provide an exciting challenge to decision science, operations research and artificial intelligence.

There are two kinds of methods to deal with negotiation problems: *analytical approaches* and *knowledge-based approaches.* Analytical approaches, such as Game Theory [7] and theories of decision-making [8] depend on specific mathematical model, and can only solve some simplified problem.

On the other hand, *knowledge-based* systems, which belongs to AI, involve a symbolic representation of the problem and simulating the human behaviors in negotiation. A lot of former work has been dedicated to design the negotiation rules[4], build knowledge base, and develop negotiation protocol. One can image, a general negation model can not be accomplished without huge and complex work. While this approach is currently primitive, it is much more flexible and promising than the analytical ones.

C. Genetic Algorithm applied to support negotiation

The fundamental motivation that drive us to use Genetic Algorithms in supporting negotiation is the view that the processing of negotiation can be viewed as negotiators jointly searching a multi-dimensional space and then agreeing to a single point in the space.

The GA supporting negotiation in this paper have the following features,

1) It is executed by a group of agent.

2) Complex information exchange is avoided in the

IEEE Int. Conf. on Man System and Cybernetics, Orlando, USA, Oct. 1997

process of negotiation.

3) Compromising is implicated in the process.

In this paper, the goal of MAS system is to design complex mechanic product. In our experiment, the process of designing gear pump is simulated.

2. FUNDAMENTAL OF GENETIC ALGORITHM

Genetic Algorithms (GAs) have recently received much attention as a robust stochastic searching algorithms for various optimization problems. This class of methods is based on the principles of natural selection and natural genetics that combine the notion of survival of the fittest, jet random search and parallel valuation of nodes in search space.

Genetic algorithms differ from traditional optimization methods in the following ways[4],

1)Genetic algorithms use a coding of the parameter set rather than the parameters themselves.

2) Genetic algorithms search from a population of search nodes instead of from a single one.

3) Genetic algorithms use probabilistic transition rules.

A genetic algorithm consists of a string representation ("genes") of the nodes in the search space, a set of genetic operators for generating new search nodes, a fitness function to evaluate the search nodes, and a stochastic assignment to control the genetic operators.

Typically, a genetic algorithm consists of the following steps:

```
procedure    GA program
begin
    t=0
    initialize P(t)
    evaluate P(t)
    while (not termination-condition) do
    begin
        t=t+1
        select P(t) from P(t-1)
        alter P(t)
        evaluate P(t)
    end
end
```

Genetic algorithms use the notion of survival of the fittest by passing "good" genes to the next generation of strings and combining different strings to explore new search points. The construction of a genetic algorithm for any problem can be separated into four distinct and yet related tasks

1) the choice of the representation of the string

2) the design of the genetic operators

3) the determination of the fitness function, and the determination of the probabilities controlling the genetic operators.

Each of the above four components greatly affects the solution as well as the performance of the genetic algorithm. In the following section, we examine each of them for the problem of negotiation.

3. GENETIC ALGORITHMS IN CONCURRENT PRODUCT DEVELOPMENT

A. Description of Negotiation Process

The inputs of a CPD system are the requirements of customer, which is collected by a sales agent. The output is the complete design scheme described by many attributes, such as physical characteristics, critical component characteristics, critical product and process parameters.

Every attribute has a feasible arrange. The upper stream attributes influence the down stream attributes. These attributes are determined by different agents sequentially and interdependently. The space of possible solutions is considerably large. The goal of the CPD method is to obtain a optimal solution based on the cooperative work of agents.

Agents tend to determine the attributes in their own conveniences, for their local knowledge and interest. Two agents would have different opinions to one attribute. Conflicts arises inevitably. Unfortunately, any agent in the CPD group can only partially determine the scheme, no agent has the knowledge to choose the best scheme. The conflicts can only be resolved via many loops of negotiation. In the process of negotiation, the scheme is evaluated and modified by agents with different opinions, until finally the scheme is accepted by all the agents in the group.

In this way, the process of negotiation can be viewed as negotiators jointly searching a multi-dimensional space and then agreeing to a single point in the space. The final scheme can be viewed as a satisfying solution in the space. For this reason, some heuristic searching technique can be used in negotiation process [7].

B. Why Genetic Algorithm?

Genetic Algorithm is chosen because it has a very flexible structure and can be modified to have multi-participators.

In the process of evaluation, all agents contribute partial evaluations, the overall evaluation is synthesized from all the partial evaluations. In this way, the distributed agents can influence the searching direction and the final result.

Besides mutation process, counter proposals are used to form new chromosomes. In detail, each proposal modifies the scheme locally. These changes will be sent to other members. In order to form a new and feasible solution, other member are forced to concede. In common negotiation process, the concession is hard to achieve. However, in GA, by giving each proposal a chance, it can be made without discussion. Because GA searches from many nodes, poor proposals will not affect the final result. This process matches the essential of mutation, which is to diverse the population. The increased amount of chromosomes will be reduced after next reproduction.

There are other reasons to use GA. First, GA searches from different nodes. It is similar to product design process, in which engineers consider several possible schemes concurrently. Moreover, GA has the feature of dynamic restructuring problem, which is compatible with the feature of negotiation process. After each generation, the chromosomes are renovated.

C. Agent Model in Genetic Algorithm

Although an intelligent agent has many other functions and can execute complex actions, in our GA-based method, the agents, except the Manager Agent, can be modeled by a tuple,

$$AGT = (S, OP, COST) \qquad (1)$$

in which,

S : solution space

OP : operation set, $OP = \{op_1, op_2, ...\}$,

op_1 realizes the solution state to s_1

$COST$: $COST = \{\cos t_1, \cos t_2, ...\}$, $\cos t_i$ is, a real, the cost of implementing op_i ,

Agent has the ability to evaluate the presented design scheme and give the COST of implementation of such scheme. Certainly, any agent in the CPD group can only determine the attributes in its own domain. All other relevant attributes are also its concern. If it finds that some state of the world will give it more benefit, it rises a counter-proposal to try to change current decision.

Manager Agent is the agent that manages genetic operations and collects partial evaluations from other agents. Counter proposals are also processed by it.

4. IMPLEMENTATION OF GENETIC ALGORITHM

A. Encode

Let A_i represents an attribute in the design scheme.
The whole scheme is
$A_1,. A_2, \ldots A_i, \ldots, A_n$
One scheme is a state of solution space.

One chromosome consists of all the information in one scheme. Each attribute is correspondent to one gene and is coded respectively. In one chromosome, the attributes are arranged linearly with upper stream attribute in the front, as displayed in figure 1.

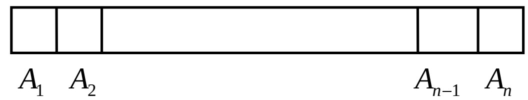

Fig. 1 string of chromosome

Some attributes, such as geometry size, are continuous ones, and have various ranges. While others, such as material attribute, are discrete, and have only countable choices. In this situation, float and binary codes are mixed in a chromosome, according to the nature of attributes.

B. Initial Population

P(0) is produced randomly. Designer Agent provides the attributes about physical characteristics. Other agents determine their correspondent attributes based on the upper stream attributes. All possible choices can form a chromosome, while only the best ones are chosen as members of first generation.

C. Evaluation and Fitness Function

To each chromosome, every agent gives its partial evaluation. The evaluation is present as COST, which is the cost for it to realize such a design scheme. It is the instinct of a agent to minimize its COST.

Manager agent synthesizes all the evaluations made by the other agents in the group, and gives the fitness to each scheme using the following formula.

$$Fitness_i = 1 / \sum^{j} COST_{j,i} \qquad (2)$$

Where i is the order of chromosome, j is the order of agent.

D. Genetic Operations: Reproduction, Crossover, Mutation

Manager Agent implements the genetic operations.

Reproduction: The children are selected from the population using the Roulette Wheel Selection Algorithm [5]. By contributing their evaluations of partial design scheme, agents influence the search direction.

Crossover: A cutting site is chosen at random and the “children” are obtained by each taking the first part from one parent and the second from the other. The crossover process provides a chance to diverse the scheme.

Mutation: Mutation operates on a single chromosome. By changing one or more elements on the chain of symbols, new genes are created. The counter proposals are used to modify the design schemes in this step. This can be viewed as a kind of mutation. This change will inform other agents. In order to form a feasible solution, the relevant agents are forced to change their parts in the scheme,. The new attributes are coded and they replace the old ones in the chain of symbols.

```
procedure GA program supporting negotiation
begin
    t=0
    Generate the first generation P(0)
    Evaluation(P(0))
    while (not termination-condition) do
    begin
        t=t+1
        P(t)=Crossover(P(t))
        P(t)=Mutation(P(t))
        Present counter-proposals
        Produce new possible chromosome from
            counter-proposals
        Evaluation(P(t))
        P(t+1)=Reproduction(P(t))
    end
end
```

In common negotiation process, the concession is hard to be made. But in our view, everyone should have a chance to realize his suggestions to the improvement of global features to see if his method works.

5. DEMONSTRATION

Consider, for example, the process of designing a gear pump. The CPD team consists of Manager Agent, Designer Agent, Sales Agent, Manufacturing Agent, and Transportation Agent. For the system structure, see [1].

The requirements of a certain new gear pump are given by Sales Agent. Normally the required throughput is given. The goal of the team is to design a scheme with lowest cost.

Transportation Agent argues that a smaller gear pump is better, so that the transportation cost is lower.

Designer Agent suggests using a standard gear pump, so that the blueprint needs little modification. Even though a larger gear pump is adopted for a relative smaller throughput, it may be more economical. With a larger size, the turning speed of the gear can be lower, and the pressure in the housing is lower too, so that gears with lower precision can be used.

Manufacturing Agent concerns the cost of manufacturing. If the gear size is smaller, the cost of its material and processing would be lower. While a smaller gear with higher precision can be more expensive than a larger gear with lower precision.

This example illustrates the compromising between the standard larger gear pump and the non-standard but smaller gear pump.

Chart 1 draws the evolutionary process of four critical attributes of the best scheme in each generation. Z and m are two important parameters of a gear. The dimension of the gear is proportional to Z*m. Chart 2 draws the evolutionary process of the costs of the four agents of the best scheme in each generation. The two charts begin from generation 100 to generation 200, because the first 100 generations are similar to generations from 100 to 110.

Before generation 110, the schemes are not very satisfying. At least one agent presents a very high cost. Starting from generation 140, the total cost drops down considerably and monotonously, with little local adjustment of partial cost.

From generation 148 to generation 170, there is a process of compromising. It was showed in chart 1. In Generation 148, the best scheme is a gear pump with a small gear high turning speed. In this stage, the cost of Transportation Agent reaches the local optimal point, while the cost of Designer Agent is in normal arrange, and the cost of Manufacturing Agent is very high, because it has to introduce an expensive method to process the gear with high precision.

Then another scheme with larger gear and non-standard throughput is reached. The Cost of Designer Agent and Transportation are a little higher, but the cost of Manufacturing Agent is considerably lowered.

After that the Designer Agent adopted the scheme with a standard throughput, so old blueprint can be used and the cost of design is reduced. In this way, although the cost of Manufacturing Agent turned a little higher, the total cost of the scheme became lower.

From generation 164, there is no significant modification to the scheme. This means the satisfying solution is found.

The parameters of GA is listed blow,

Population Size	Generation	Pc	Pm
20	200	0.8	0.1

Although the parameters above are feasible, they may not optimal.

6. CONCLUSION

In this paper, The GA is used as a tool to resolve conflicts in cooperative work. Compared with other methods, GA avoids the complexity of developing negotiation rules and exchanging information. It fulfilled the requirements of negotiation through *evaluating scheme*, *proposing* counter proposals, *dynamic restructuring* problem and *avoiding* dead lock. This method provides a novel method to support negotiation between multiagent.

By applying GA, we make following progress:

(1) In the mutation step, agents are forced to concede. The complexity of negotiation process is simplified.

(2) By evaluating the partial design scheme, different opinions influence the search direction, so that multiple agents help to choose the most feasible solution.

7. REFERENCES

[1] Hu Yongtong, Liu Ping, Yan Yuhong, Zheng Danian, Ma Changchao, Jurgen Bode, Ren Shouju, P959-964, Proceedings of 95'IEEE Conference on System, Man, and Cybernetics.

[2] Park, H., Cutkosky, M. R. et al, "An Agent-based Approach to Concurrent Cable Harness Design", Artificial Intelligence for Engineering, Analysis, and Manufacturing (1994).8.45-61

[3] Zbigniew Michalewicz, "Genetic Algorithm + Data Structures = Evolution Programs", second, Extended Edition, published by 1994

[4] Martin, S., Szapiro, T., Haigh, K., "Genetic Algorithms Approach to a Negotiation Support System", IEEE Trans. Systems, Man, and Cybernetics, Jan./Feb., Vol. 21, No. 1, P102-114

[5] a paper on the web site hass.berkeley.edu

[6]Sycara, K. P. "Machine Learning for Intelligent Supporting of Conflict Resolution", Journal of Decision Support System, Vol. 10, 1993,p121-136

[7]Zlotkin, G., Rosenchein, J. S., "Compromise in negotiation Exploiting Worth Functions over States", Artificial Intelligence, Vol. 84, 1996, p.151-176

[8]Kersten. G., E., "NEGO-Group decision support System", inform. Manage. Vol. 8, No. 5 p237-246, 1985

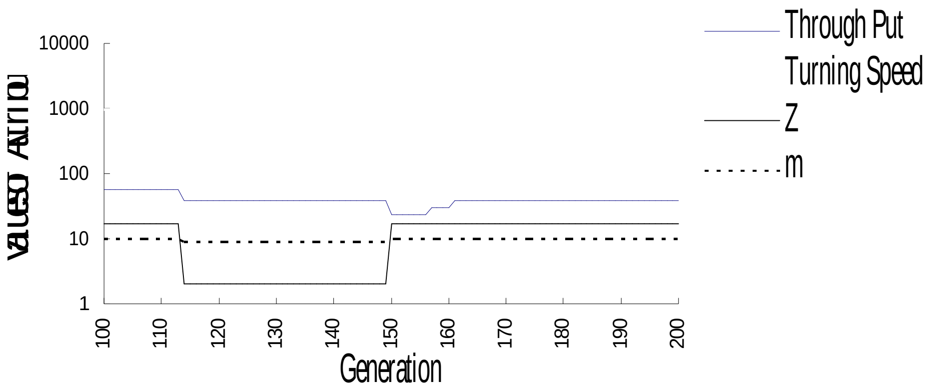

Chart 1. Evolution of Characteristics

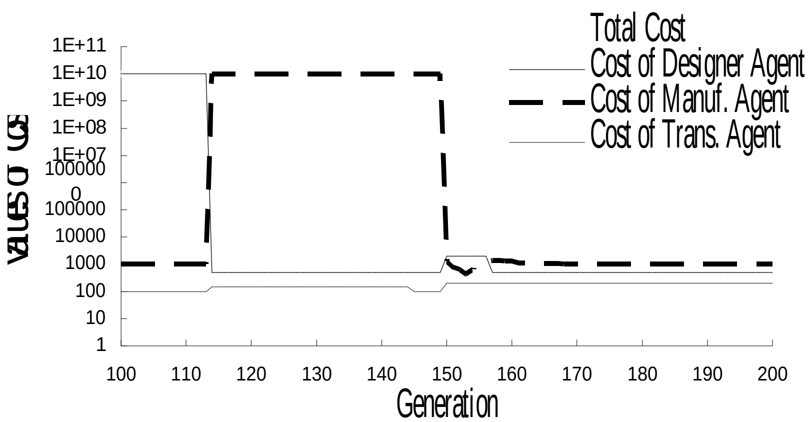

Chart 2. Evolution of Costs