Proceedings of the 2001 Winter Simulation Conference
B. A. Peters, J. S. Smith, D. J. Medeiros, and M. W. Rohrer, eds.

SOLVING SEQUENTIAL DECISION-MAKING PROBLEMS UNDER VIRTUAL REALITY SIMULATION SYSTEM

Yang Xianglong
Feng Yuncheng
Li Tao

System Simulation Laboratory
School of Economics & Management
Beijing University of Aeronautics & Astronautics
Beijing, 100083, P.R. CHINA

Wang Fei

Institute of International Economy
State Development Planning Commission
People's Republic of China
Rm. B-912, Guohong Building
Beijing, 100038, P.R. CHINA

ABSTRACT

A large class of problems of sequential decision-making can be modeled as Markov or Semi-Markov Decision Problems, which can be solved by classical methods of dynamic programming. However, the computational complexity of the classical MDP algorithms, such as value iteration and policy iteration, is prohibitive and will grow intractably with the size of problems. Furthermore, they require for each action the one step transition probability and reward matrices, which is often unrealistic to obtain for large and complex systems. Here, we provide the decision-maker a sequential decision-making enviroment by establishing a virtual reality simulation system, where the uncertainty property of system can also be shown. In order to obtain the optimal or near optimal policy of sequential decision problem, simulation optimization algorithms as infinitesimal perturbation analysis are applied to complex queuing systems. We present a detailed study of this method on the sequential decision-making problem in Boeing-737 assembling process.

1 INTRODUCTION

Sequential decision-making means that the decision-maker makes a series of actions according to the system status as well as his preference to form a decision policy. Many problems of sequential decision-making under uncertainty ranging from manufacturing to computer communication, of which the underlying probability structure is a Markov process, can be modeled as Markov or Semi-Markov Decision Problems (MDPs or SMDPs). Such problems can be solved by classical methods of stochastic dynamic programming. The framework of dynamic programming and MDPs developed by Bellmen (1957) and extended by Karlin (1955), Howard (1960), Blackwell (1965) is quite extensive and rigorous. Well known algorithms, such as value iteration, policy iteration, and linear programming can find optimal solution of MDPs. However, they require computation of the corresponding one step transition probability matrix and the one step transition reward matrix using the distributions of the random variables that govern the stochastic processes underlying the system. For complex systems with large state spaces, the burden of developing the expressions for transition probabilities and rewards could be enormous (Das et al. 1999). Also, dynamic programming suffers from the curse of dimensionality and from the curse of modeling, which is why it is of little use in solving problems with a large state space and complex probability structures. In the absence of better approaches, problem-specific heuristic algorithms are often used to reach acceptable near-optimal solutions.

Recently, computer simulation-based reinforcement learning (RL) methods of stochastic approximation have been proposed as viable alternatives for obtaining near-optimal policies for large scale MDPs with considerably less computational effort than what is required for dynamic programming (DP) algorithm. RL has two distinct advantages over DP. Firstly, it avoids the need for computing the transition probability and the reward matrices. The reason being that it uses discrete event simulation (Law and Kelton 1991) as its modeling tool, which requires only the probability distribution of the process random variables but the one-step transition probabilities. Secondly, RL methods can handle problems with very large state spaces since its computational burden is related only to value function estimation, that is just the advantage of computer simulation (Das et al. 1999).

However, the decision-making rules are predefined in RL method that means the fixed actions should be made according to corresponding states. So it can not reflect the prejudice of decision-maker and limit his freedom in the

sequential decision-making process. In this paper, a new decision mode (3W+N) of simulation-based sequential decision is proposed and a virtual reality (VR) simulation environment is designed for solving such problems. This new Interactive VR simulation system is especially derived from research of sequential decision-making, and it is also built and tested in local network (Wang 2000).

In this paper, we first discuss the new decision mode (3W+N) of simulation-based sequential decision, and then the simulation optimization algorithm on sequential decision-making under VR simulation enviroment is proposed. How to realize the VR enviroment and simulation optimization in such enviroment is also discussed. At last, the experiment for solving sequential decision problem involving Boeing-737 section-48 assembling manufacturing process is presented, which is one of the joint ventures between China and U.S in the area of aircraft manufacturing.

2 3W+N DECISION MODE

Decision-maker can take a serial of actions in accordance with the system states and his preference in sequential decision-making process. By this way, the risk in sequential decision-making process should be less than that making decisions before the implementation of the system. Many sequential decision-making problems in engineering can be modeled as Markov or Semi-Markov Decision Problems, which can be solved by classical methods of dynamic programming in theory. But dynamic programming suffers from the curse of dimensionality and from the curse of modeling, which is why it is of little use in solving problems with a large state space and complex probability structures in real-life.

So a new decision mode (3W+N) based on experiment for such research is proposed. It is designed for solving Markov Decision Problems under VR simulation environment. Virtual Reality (VR) is a rapidly developed integrated technology in recent years. It is a way for decision-makers to visualize, manipulate and interact with computers and extremely complex data. By this way, the virtual space is created to provide the interaction between human and virtual "real" system displayed on computer. In this virtual world, users can experience the interactive behavior such as looking, hearing, moving and so on. Moreover, the interactive dynamic three-dimensional (3D) simulation under VR environment is constructed by combining computer simulation and VR technique that is called VR simulation mechanism. In this virtual "real" world, decision-maker can obtain the dynamic states of system running and "enter" it to make series of decisions according to the system states and his preference, the decision epochs is no longer determined by predefined criterion. Therefore, decision-maker can obtain more useful information from this near real-world system to make the sequential decision process more intuitive and reliable.

In this way, we propose the 3W+N decision mode that means **Where, What, When** and **Number** of decisions. All of which depend on the current states of the system and the preference of the decision-makers**.**

- **Where** is to choose the place where the decision-maker to make an action. A properly chosen decision-making place can greatly represent the decision-maker's taste or preference.
- **What** is to decide to take which kind of actions and the intensity of actions, all the actions will construct a decision set, from which the sequential policy will be given.
- **When** and **Number** is to decide the time to make decision and how many times in the decision process.

Hence, this kind of decision model is different from the traditional one, because it can really represent the true life. Such more agile and intuitive decision mode can reflect the random and nonperiodic property of sequential decision in real-life.

It is obvious that the decision place, time, numbers and actions intensity are all stochastic in 3W+N decision mode. Then such decision process can be viewed as the combination of two processes.

First one is formal Markov Process, it can be shown as:

$$S_t = \left\{ s_t^1, s_t^2, \cdots, s_t^m \right\} \tag{1}$$

where s_t denotes the system state at the t decision-making epoch. At decision making epoch t , where $S_t = i, i \in S$ and the decision taken is $X_t = x, x \in X$, X_t denotes the set of possible actions at time t .

In the second process, suppose that x_t is the decision taken at epoch t . Then the next decision is taken at epochs $t+\tau$ where τ is a random variable between the decision period. Suppose that X_t and S_t be finite set, the transition probability from s_t^i at epoch t to $s_{t+\tau}^j$ at decision epoch $t+\tau$ is independent to the "past" states, i.e.

$$P\left(s_{t+\tau}^j \middle| s_t^i, x_t^i, s_{t-v}^k, x_{t-v}^k, \cdots, \right) = P\left(s_{t+\tau}^j \middle| s_t^i, x_t^i\right) \tag{2}$$

where $S_t = S_n$ provided that $t = T$. The implication is that such stochastic decision process is a Markov chain too. Howard (1960) has proposed the proof that the Bellman Theory (1957) is also applicable to stochastic decision process.

To construct a whole decision mode, a performance criteria is necessary, which can sum over the time horizon and represent the result of sequential decisions. For in-

stance, the value of WIP (work-in-process) is the most important performance for manufacturing system. Indeed only the criteria represented the effect of **3W+N** can be chosen as the system performance criteria.

3 SEQUENTIAL DECISION OPTIMIZATION ALGORITHM

Perturbation analysis (PA) is a kind of gradient-based estimation method in optimization, the most famous one is infinitesimal perturbation analysis (IPA) by which all partial gradients of an objective function are estimated from a single simulation run. The idea is that if an input variable into a system is perturbed by an infinitesimal amount, the sensitivity of the output variable to the parameter can be estimated by tracing its pattern of propagation. This will be a function of the fraction of propagations that diminished before having a significant effect on the response of interest. IPA assumes that an infinitesimal perturbation in an input variable does not affect the sequence of events but only makes their occurrence times slide smoothly. The fact that all derivatives can be derived from a single simulation run, which represents a significant advantage in terms of computational efficiency. On the other hand, the estimators derived using IPA are often biased and inconsistent (Yolanda and Anu 1997). IPA estimation will be unbiased provided that the structured condition is satisfied. The essential of such condition is that if the events occurred in succession exchange the occurrence sequence because of perturbation, system states will keep immovability (Huang 1997).

Such four basic problems is involved in IPA algorithm: rules of perturbation generation, rules of perturbation propagation, computation of sample gradient and the statistics properties of sample gradient related to performance gradient. In simulation enviroment, the perturbation generation process can be synchronously treated with random variables (RV) sampling process. Assumed that the interested parameter is θ and $F(x,\theta)$ is the cumulative distributed function (CDF) of RV X, then the sampling process can be realized by using inverse-transform method — the sampling of X is $x = F^{-1}(u,\theta)$, where u is the sampling of uniform distributed RV U in interval (0,1) (Law and Kelton 1991). If the parameter perturbation is $\Delta\theta$, then the sampling corresponding to $\theta+\Delta\theta$ is $x+\Delta x = F^{-1}(u,\theta+\Delta\theta)$, i.e. the sampling perturbation of RV X is Δx. When the parameter perturbation is infinitesimal, the ratio of RV perturbation on the parameter perturbation is:

$$D_\theta x = D_\theta F^{-1}(u,\theta) = -D_\theta F / D_x F \qquad (3)$$

then the perturbation of parameter is transformed to the sampling gradient of RV.

In our research, IPA is applied to assembling queuing network. Boeing-737 48-section consists of 9 subassemblies, which are composed of hundreds of components or parts. Those subassemblies, components and parts are assembled on the special jigs and equipment named as "FAJ" and "FME" and each subassembly possesses its own "FAJ" and "FME". Then an assembling queuing network is constructed that subassemblies, components and parts can be viewed as customers and "FAJ" and "FME" as servers. The product structure tree is also provided as Figure 1.

Obviously it is a very complex queuing network, we choose the "Texas Star" assembling queuing system as the

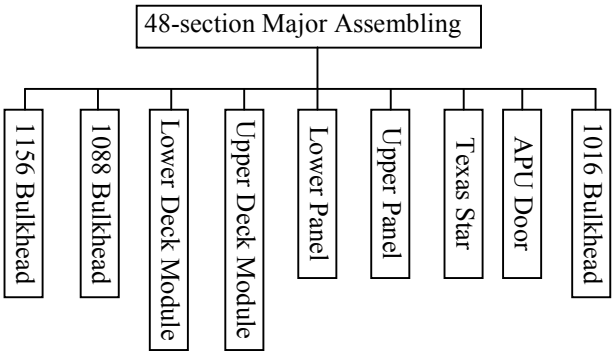

Figure 1: B737 48-section Assembling Product Structure Tree

research object on sequential decision to expatiate the algorithm for simplicity. "Texas star" assembling process is shown as Figure 2:

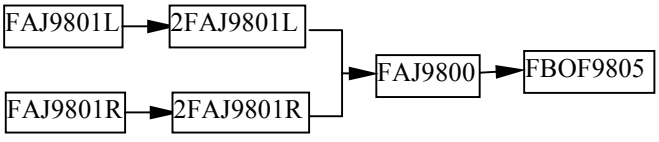

Figure 2: Assembling Process of "Texas Star"

The manufacturing equipment named with their codes used in assembling process as servers are shown in Figure 2. Left X-beam and Right X-beam are all composed of two beams, so the first beams of them are processed on FAJ9801L and FAJ9801R separately. Then Left X-beam and Right X-beam are assembled with two beams on 2FAJ9801L and 2FAJ9801R. The matching process of Left and Right X-beam and the Back Beam to form Texas star is in succession. After refined processing on FBOF9805 and inspecting, the whole process ends. There are two sets of equipments to assembling Left and Right X-beam and Texas star, one of them are standby. The detail assembling process can be viewed in Figure 5. According to 3W+N decision mode, the key working procedure and equipment can be chosen as decision making place (Where), as well starting or stopping standby equipment and assembling

process and changing the arrival rate or service rate can be chosen as actions or decisions (What) in this assembling model. Decision-maker can make actions at the epochs of equipment failure or deficiency of assembling ability (When) and decide the decision numbers (Number) according to the system states and his preference. Here, the performance measure includes the value of WIP C(W) that can be expressed as $\sum_i v_i N_i$, where N_i and v_i are the number and cost of the i^{th} part or component in system. We can use the average sojourn time to compute WIP in simulation. At the same time, it also should include the variable cost derived from decision C(D), such as the running cost of standby equipment and the salary of workers that can be expressed as $\sum_j v_j M_j$, where M_j and v_j are the number of j^{th} equipment and the corresponding cost on this equipment. Usually, decreasing the value of WIP will cause the increasing of variable cost. So our objective function is:

$$min\{C|C = C(W) + C(D)\}. \quad (4)$$

Here, define the system before the perturbation is added as Nominal System (NS) and the sample path of NS as Nominal Path (NP). The relevant system and sample path to perturbation are PS and PP. Define the Generalized Busy Period (GBP) provided that $Q \geq n$, where Q is the number of customers in system and n is the number of servers. Then in decision model, the actions made as changing the arrival rate or service rate and starting or stopping the standby equipment at certain working procedure will affect the system in 3 aspects in this decision model. The first aspect is perturbation generation to the number of parts at currently working procedure, the second is direct perturbation propagation to the number of parts at working procedure just after the former one and the third is indirect perturbation propagation to the number at other working procedure. Such 3 type perturbations have their algorithm separately. Here, the second algorithm is present to explain the PA method on the direct perturbation propagated working procedure:

Step 1.Define the variables in system: arrival time of customer (*ARRT*) and its perturbation (*PARR*), the reference time of PA (*NTIM)*, the arrival time increment of the first customer in PP (*NARR*), the time increment while service begins (*NSER*), the adjust of dispatch interval (*BTDP*), the number of served customer (*M*) and the perturbation of waiting time (*PAWT*) and its total time (*SPAWT*).

Step 2. Mark the value of *INDEX* of arriving customers type: (1)$Q<n$ *or* (2)$Q \geq n$.

Step 3. According to the *INDEX* to decide the operating type when the customer is served, then go to step *INDEX*+3.

Step4. *NTIM=ARRT*, *NARR=PARR*, *NSER=PARR*, *PAWT*=0, *BTDP*= the service time of customer, then go to step 7.

Step 5. If this customer is served in PP in advance, i.e. *ARRT+PARR<NTIM+NARR*, then

{if this customer is the firstly arriving in GBP, then
{*NSER=ARRT+PARR-NTIM*}
else
{*NSER*=max{*NSER*, *ARRT+PARR-NTIM*},
the waiting time of customer in PP
is NSER+NTIM-ARRT-PARR,
PAWT= waiting time in PP - waiting time in NP}
}

else

{if the immediately predecessor customer is served in PP in advance, then
{*NSER=NSER+BTDP*}
else
{ *NSER=NSER+* the dispatch interval of the customer in NP,
NSER=max{*NSER*, *ARRT+PARR-NTIM*}
PAWT = NSER-NARR- waiting time in NP}
}

Step 6. If this customer is not served in PP in advance, then

{ if n>1, then
{*BTDP*=2*service time of customer/(n-1)}
else
{ *BTDP*= the dispatch interval of the customer in NP}
}

Step 7. *SPAWT=SPAWT+PAWT* when service ends, *M=M*+1, If simulation ends, the expected perturbation of sojourn time at this server is *SPAWT/M;* else, go to step 2.

Then we can obtain the gradient sampling from the parameter perturbation to performance measure of system. In the step n of iteration, assume that the iteration step length is α_n and estimation of iterated direction d_n from the perturbation process, the iterated parameter θ_{n+1} in iteration step $n+1$ can be received as:

$$\theta_{n+1} = \theta_n - \alpha_n d_n . \quad (5)$$

Usually, iteration is processed once after the predefined number of entities have been served in simulation and the terminated condition is often expressed as:

$$|\theta_{n+1} - \theta_n| < \varepsilon \quad (6)$$

where ε is a predefined small positive number. Then the latest estimation of parameter θ_{n+1} is used to estimate the performance measure as the optimal parameter in simulation. Here, the gradient sampling or estimation of iterated direction in each step is obtained by PA.

In sequential decision process, decision-maker wants to receive reasonable support from simulation optimization technique such as IPA. For example, if decision-maker is not satisfied with the processing capability according to the VR simulation system states and his preference and then he will make a decision to change the system parameter such as service rate. IPA can provide the improved system parameters at such decision epochs. Between two successive decision epochs, IPA can get an iteration result to modify the system parameters at present epoch according to the optimization regulation. In the formal optimization regulation, iteration occurs when a predefined number of entities have been served. Because of the randomness of sequential decision, the iteration number is not a constant and then it will be determined according to the decision epochs. If the time is too short between two epochs, then the sample size is not enough to estimate the gradient and modify the parameters. So this is a noticeable problem in solving sequential decision problem using IPA. But as a Single Running Optimization (SRO) method, we take advantage from its executing efficiency.

4 VR SIMULATION-BASED DECISIONS AND ITS OPTIMIZATION ENVIROMENT AND EXPERIMENT RESULT

In order to realize such decision-making system under virtual reality simulation, a new interactive VR simulation system is established. This system is simulation based with the focus on the representation and participation. Now many VR systems have been presented, but most of them need expensive equipment, group of experts, lots of time to construct a perfect complex VR system. However, the most important role is simulation in our research. We need simulation to provide decision-maker the statistical information of the system being modeled and give the system states automatically. The major function of the VR system is to provide the decision-maker a near real world 3D animation to observe the running of system. Basing on this opinion, VRML (Virtual Reality Modeling Language) is selected to establish such virtual reality environment, which is an easy, economic and powerful tool to construct the VR world. Then we can focus on the simulation aspect and spend little time to code it and realize the Virtual Reality Simulation. Because of being designed for Web, VRML has no strict requirement for hardware. But it needs a web explorer plug-in to explain the VRML file and show the VR scene, this plug-in is called VRML player such as Cosmo Player an so on. With the explanation to VRML file, users can observe the virtual world from the monitor and control the moving objects and making decisions in the VR scene by using keyboard or mouse.

Moreover, VRML provides the external interface to realize the application of simulation just as our demand. Because of the web property of VRML, it provides the JAVA class as External Application Interface (EAI). By using its external classes, users can obtain the handle of system object Browser**.** Some methods of browser object are very useful, such as: *getBrowser(), createVrmlFromString(), getNode(), addRoute() and deleteRoute()*. From the browser, we can get the handle of object in VRML scene by methods *getBrowser()* and control its behavior, and display the simulation process by VRML. Combining these method and field *addChildren* and *removeChildren* in VRML nodes, entities can be added and removed dynamically in the virtual world. Using the correct combination of *TimeSensor*, *PositionInterpolator* and *OrientationInterpolator* Node, the *Route()* method can drive the object move as the proper logic at the suitable time. (Wang 2000 and Marrin 1997). Moreover, simulation programming has the natural interface to JAVA, and the latter is an object-oriented programming language. Many JAVA-based simulation programs have appeared, such as JavaSim(Little, M.C, 1996), SimJava (McNab, R., 1996), etc. They provide the chance to combine the Java-based simulation program and VRML file, the latter two can be integrated into one web page to display the simulation animation process. The VRML browser reads and explains the VRML file and builds a 3D virtual world. The VRML file just tells the browser how to build the predefined virtual world, the simulator accomplishes the simulation work and drives the virtual world to run dynamically. Then simulation programming can combine the VRML by importing VRML external interface and JAVA-based simulations class to construct VR simulation system easily (The VRML Consortium Incorporated 1997).

In our research, such VR simulation system is constructed to provide the function to making sequential decision and optimizing the parameters with above algorithm. All these functions can be realized through the Java Applet in HTML pages and JDBC driver support to access database used to store the simulation data and display the improved parameters. The system framework can be viewed as Figure 3 and the VR simulation-based system is shown in Figure 4.

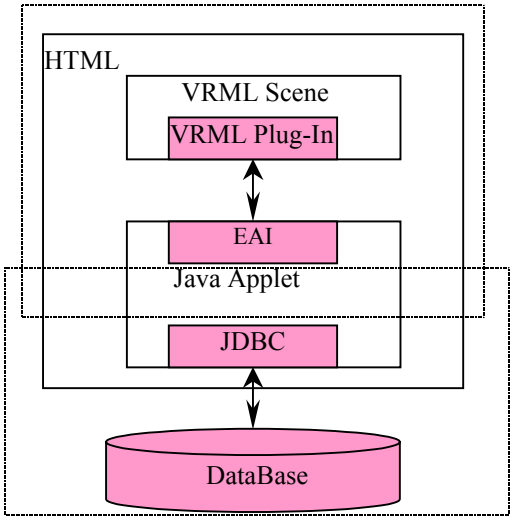

Figure 3: Mechanism of VR Simulation-Based Decision and Optimization System

In the VR simulation enviroment shown in Figure 4, the separate 3D objects in this scene are established with 3DSMax, the famous 3D animation software, and can be convert into the VRML file format by using the function "Export to…. " Some other software as Internet 3D Space Builder, AC-3D etc., can also provide such function. By this way, we can build complex 3D models more easily but the size of these files is often bigger than that directly built in VRML. There is a tip to settle this problem partially: in some 3D building software, a function is provided — "Optimize" to remove some duplicate vertices and surfaces from objects and do not influence their 3D effect. Then the converted file size will be reduced. In VRML file, the world coordinate space for all objects is defined, and all objects included in the file by the "Inline" node. And the animation according to simulation result is realized with the method of combining VRML EAI and Java-based simulator as mentioned above. In Figure 4, the left scene in Internet Explorer is a VR scene and the right one is Java Applet where the input parameter can be changed for decision-making and the simulation output is displaying. The buttons in Applet are used to start animated or numeric simulation, pause simulation, restart simulation and run optimization to obtain improved parameters.

By using such simulation system a model was developed for practical assembling system "Texas Star" as mentioned above, which saves lots of cost and time for the cooperated factory. In this assembling process, the supervisor should take account of the restrictions of the loading capacities, and decide whether the backup machine should be start or not when the main assembling machine is failed or the production capacity is not enough. The supervisor will

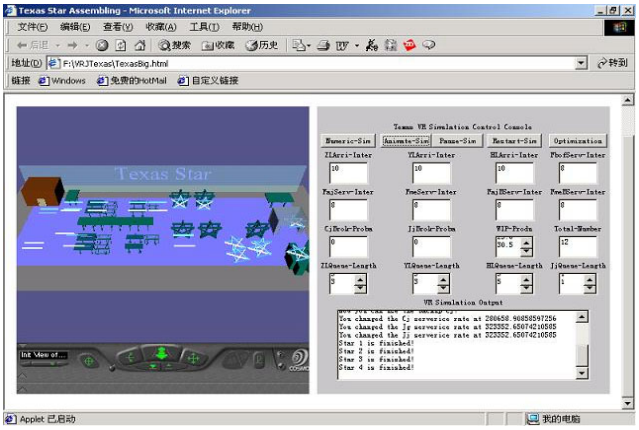

Figure 4: VR Simulation-based System of Boeing-737 "Texas Star" Assembling

decide the next action based on present working status and final objectives. In this assembling model, we have set 5 decision-making place (where), 10 kinds of decisions (what), and the decision epochs and times can be decide by the supervisor to make the minimal performance measure C in formula (4).

Then an experiment of such sequential decision and optimization under VR simulation enviroment is presented. In this experiment, actions include changing the arrival rate of Left X-beam, Right X-beam and Back Beam, starting the standby equipment of X-beam and suspending certain assembling service and so on. The initial experiment condition of average arrival and service rate is shown in Table 1.

Table 1: Arrival and Service Rate of Parts and Equipment

Left X-beam	Right X-beam	Back Beam	FAJ9801L/R
30	35	45	25
Standby FAJ9801L/R	FAJ9800	Standby FAJ9800	FBOF9805
25	30	30	35

The numbers in Table 1 marked with Left X-beam, Right X-beam and Back Beam are their average arrival rate and the numbers marked with the name of equipment are their average service rate. The arrival of customers is Poisson process and service time is subject to exponential distribution.

By running simulation in VR enviroment, we can get the information from VR scene and simulation output in Applet that the assembling capacity of Left, Right X-beam and Back Beam is absence to match that of refined processing on "Texas Star". The length of queues increases rapidly to make the overstock of WIP, which is shown in Figure 5.

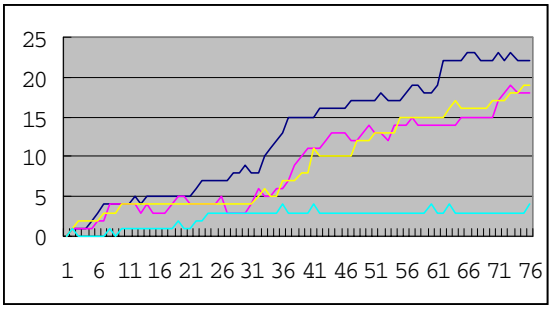

Figure 5: Number of WIP in Initial Simulation

The vertical axis shows the number of WIP and the horizontal one shows the number of products completed assembling in simulation.

In order to solve such problem, sequential decision is made as following sequence shown in Table 2:

Table 2: Decision Sequence in Experiment

Sequence	Place	Action	Time
1	Standby FAJ9801L/R	Starting standby equipment	1317
2	FAJ9800	Decreasing arrival rate of Back Beam	1968
3	Standby FAJ9801L/R	Starting standby equipment	3541
4	FBOF9805	Suspending service	5500
5	FAJ9801L/R	Decreasing arrival rate of Left X- Beam	5734

From sequential decision as above and run optimization at decision epochs with IPA, we can obtain new result of simulation running as Table 3 and Figure 6:

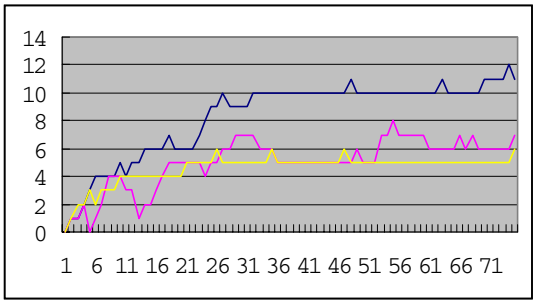

Figure 6: Number of WIP in Modified Simulation

Table 3: The comparing experiment results

	Initial Experiment	Modified Experiment
Total number of products	160	160
Simulation length	8294.9	7020.7
Number of WIP	82.5	28.5
Queue length of Left X-beam	12	8
Queue length of Right X-beam	8	5
Queue length of Back Beam	8	5
Queue length of Refined "Star"	3	1

The numbers of WIP, Queue length of Left X-beam, Right X-beam, Back Beam and Refined "Star" in Table 3 are all their average in the simulation. It can be viewed that the simulation result processed with sequential decision and optimization is better that initial one.

5 CONCLUSION

This paper proposes the optimization algorithm on sequential decision problem based on VR simulation system. By using such simulation system, a model about real system was developed to provide the decision-maker a virtual world to making decisions with the support of optimization algorithm. The VR Boeing-737 48-section assembling simulation system makes our cooperation factory save lots of cost and time in establishing the assembling line. The properties of man-computer interaction in VR and describing the uncertainty in simulation provide the feasible method and convenient experiment space to analyze the man-in-the-loop decision problem. This simulation system can be not only used to train decision-makers: either novice decision-makers or potentially established decision-makers by comparing the decision result to optimized one, but also to operate the real-life facility. Then we can see this special simulation environment is the new trends of today's simulation software, Visual Simulation and VR Simulation. As for further development, the integrated development environment is under establishing. In such VR simulation system, every simulation user can easily code and compile the model and construct the VR scene according to his demand.

ACKNOWLEDGMENT

This work is supported by National Science Foundation of China Grant 79930900 to Beijing University of Aeronautics and Astronautics.

REFERENCES

Bellman, R. E. 1957. *Dynamic Programmin*g. Princeton University Press, Princeton, NJ

Blackwell, S. J. 1965. Discrete dynamic programming. *Ann. Math. Stat*., 33 226-235

Cinlar, E. 1975. *Introduction to Stochastic Process*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey

Das, T., Gosavi, A., Mahadevan, S. and Marchallek, N. 1999. Solving semi-Markov decision problems using average reward reinforcement learning. *Management Science*, Vol. 45, No. 4

Howard, R. 1960. *Dynamic Programming and Markov Processes*. MIT Press, Cambridge, MA.

Huang Hongxuan 1997. *Perturbation Theory of Discrete Event System and Single Run Optimization via Simulation*. Doctor Degree Dissertation of Beijing University of Aeronautics and Astronautics, Beijing

Karlin, S. 1955. The structure of dynamic programming models, *Naval Res. Logist*., Quart. 2 285-294

Law, A. M., W. D. Kelton 1991. *Simulation Modeling and Analysis*. McGraw Hill, New York

Little, M.C. 1996. *JavaSim Homepage*, <http://marlish.ncl.ac.uk:8080/JavaSim>

Marrin, C. 1997. *Proposal for a VRML 2.0 Informative Annex.* http://reality.sgi.com/cmarrin/vrml/externalAPI.html

McNab, R. 1996. *A Guide to the SimJava Package*, Department of Computer Science, University of Edinburg, UK,<http://www.dcs.ed.ac.uk/home/hase/simjava/simjava-1.0>

The VRML Consortium Incorporated. 1997. *The Virtual Reality Modeling Language* International Standard ISO/IEC 14772-1:1997

Wang Fei 2000. *Research on Theory and Application for Simulation Decision under Virtual Reality*. Doctor Degree Dissertation of Beijing University of Aeronautics and Astronautics, Beijing

Wang Fei, Feng Yuncheng, Wei Youshuang 2000. Virtual Reality Simulation Mechanism on WWW. *In Proceedings of AreoScience 2000 of SPIE*

Yolanda Carson, Anu Maria 1997. Simulation Optimization: Methods and Application, *Proceedings of the 1997 Winter Simulation Conference*, ed. S. Andradóttir, K. J. Healy, D. H. Withers, and B. L. Nelson. 118-126

AUTHOR BIOGRAPHIES

YANG XIANGLONG is a Ph.D. student at the School of Economics & Management, Beijing University of Aeronautics & Astronautics. His current research area is Virtual Reality technology and simulation optimization.

FENG YUNCHENG is a Professor at the School of Economics & Management, Beijing University of Aeronautics & Astronautics. He has served as honorary President of Computer Simulation Association of China. His current research interests include simulation optimization, simulation output analysis, manufacturing system simulation, Virtual Reality technology and Web based simulation.

WANG FEI had obtained his Ph.D. degree at Beijing University of Aeronautics & Astronautics in 2000. His current research interests include Macroeconomics analysis and WTO strategy of China.

LI TAO is a Ph.D. student at the School of Economics & Management, Beijing University of Aeronautics & Astronautics. His current research interests include multimedia simulation and Web-based simulation.