Grid Scheduling Optimization Under Conditions of Uncertainty

Zeng Bin, Luo Zhaohui, and Wei Jun

Department of management, Naval University of Engineering,
Wuhan, China
zbtrueice@163.com

Abstract. One of the biggest challenges in building grid schedulers is how to deal with the uncertainty in what future computational resources will be available. Current techniques for Grid scheduling rarely account for resources whose performance, reliability, and cost vary with time simultaneously. In this paper we address the problem of delivering a deadline based scheduling in a dynamic and uncertain environment represented by dynamic Bayesian network based stochastic resource model. The genetic algorithm is used to find the optimal and robust solutions so that the highest probability of satisfying the user's QoS objectives at a specified deadline can be achieved. It is shown via a simulation that the new methodology will not only achieving a relatively high probability of scheduling workflow with multiple goals successfully, but also be resilient to environment changes.

Keywords: workflow scheduling, grid computing, genetic algorithm, optimal scheduling scheme.

1 Introduction

A Grid's ability to produce an efficient plan or schedule for its task execution is critical to its service performance. Given the dynamic nature of a complex resource environment [1], an effective resource management system is to create the robust schedules at the right nodes and at the right time. Therefore, environmental conditions affect the feasibility of Grid's schedules, making some schedules more likely to succeed than others.

There are some approaches that try to solve the problem. The multicriteria resource selection method implemented in the Grid Resource Management System (see [2] and [3]) has been used for the evaluation of knowledge obtained from the prediction system. Nevertheless, due to incomplete and imprecise information available, results of performance prediction methods may be accompanied by considerable errors (to see examples of exact error values please refer to [4] and [5]). Another aid to avoid uncertain problems comes in the form of contracts ([6],[7]). In the simplest contracts, clients use initial estimates of job completion time to bind Grid resources, and then monitor job progress during execution to determine if the contract will be met for reacting swiftly and appropriately to

K. Li et al. (Eds.): NPC 2007, LNCS 4672, pp. 51–60, 2007.
© IFIP International Federation for Information Processing 2007

recover if it is not. However, the contract based work is limited to the system where estimates may be gathered a priori and where clients may monitor runtime progress. Some research such as FailRank [8] try to monitor the Grid sites with the highest potential to feature some failure. Thomas model the Grid scheduling system as a collection of queues where servers break down and are subsequently repaired to investigate the penalty of prolonged delays in information propagation to the scheduler [9]. Anyhow, current distributed service scheduling research has not presented a complete solution that incorporates uncertainty.

This paper introduces a framework for devising a robust Grid scheduler to maximize the probability of successfully achieving the QoS objectives, while minimizing its variability. A normative model of the stochastic Grid resource environment, based on a dynamic Bayesian network [10], to infer indirect influences and to track the time propagation of schedule actions in complex Gird environment is developed. For a user specified QoS requirements and resource constraints, a near-optimal Grid scheduling strategy is obtained via genetic algorithms, where the DBN serves as a fitness evaluator for candidate schedules.

2 DBN Model for Job Scheduling Optimization

The stochastic scheduling problem faced by an uncertain Grid system can be defined as follows: given an initial system state, determine optimal scheduling strategy that will bring the system to a specified objective state at a specified time with a relatively high probability. The objective, in our case, is the set of desired QoS objectives. The process to solve this problem is to:

1. Represent the joint dynamics of the jobs and its assigned resources;
2. Optimally select appropriate scheduling strategy;
3. Assess the probability of successfully achieving the desired QoS objectives under resources constraints.

A dynamically evolving DBN-based scheduling model $G_k = G(t_k) = (V, E, P_k)$, which can be viewed as a Bayesian network at time t_k, combines knowledge about the jobs and its execution environment. G_k is a directed acyclic graph consisting of a set of nodes V and a set of directed edges E with a fixed structure. Every node is considered as a random variable and can assume Boolean values. For each node $v_i \in V$, we define a probability mass function (pmf) $P_k(v_i) = P\{v_i(t_k)\}$ to characterize the environment uncertainty at time t_k.

The dynamic evolution of the DBN-based scheduling model unfolds through a finite horizon timeline as in shown in Fig. 1, which is discretized into T time steps (from t_1 to t_T). The solid arcs are used to represent the causal relationship in a single time step, and the dashed edges are used to show the temporal evolution of the model between neighboring time steps.

Based on Fig. 1, The definition of our DBN model for scheduling jobs is described below:

1. System state: A state consists of current execution tasks, execution time and current location, with $S(t_k) = \{P_k(v_i) | v_i \in V_k\}$ to portray the overall state of the Grid at time t_k;

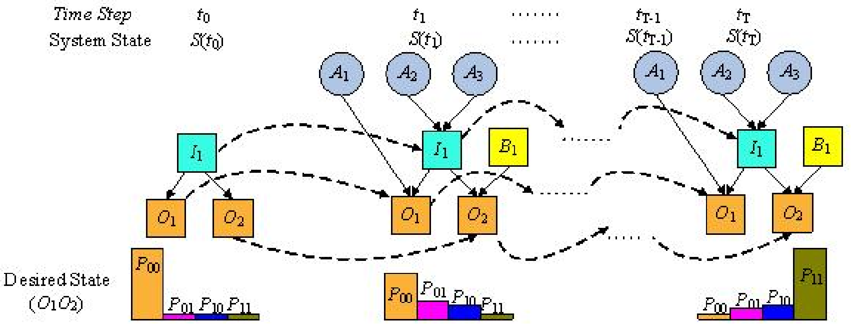


Fig. 1. Time evolution of scheduling model as a DBN

2. Objectives: Objectives are regarded as desired states of DBNs. O $= \{O_n|1 \le n \le N_O\}$ specified by the desired probability of QoS satisfied ('1' or '0') and the corresponding terminal time t_{O_n} for each QoS objective: $O_n(t_{O_n}) = 1$ or $O_n(t_{O_n}) = 0$. Here $N_O = |O|$ is the total number of QoS objectives specified by users;
3. Critical system events: regarded as noise factors, whose occurrence is beyond the control of the Grid resource management system, but will affect the resource dynamics: $B = \{B_j|1 \le j \le N_B\}$. $N_B = |B|$ is the total number of system events in the environment. In many cases, one has partial knowledge of the statistical properties (e.g., means and variances, probability distributions) for these events. For instance, if event B_1 in Fig. 1 occurs with a probability that is uniform between [0.2,0.6] at time t_k, then $P_k\{B_1 = 1\} = p_1$, $P_k\{B_1 = 0\} = 1 - p_1$, where $p_1 \sim U[0.2, 0.6]$. The prior pmfs in the model are normally acquired from domain experiences or analysis of Grid feedback sources [8]. Note that some events may have inhibiting effects in that they reduce the probability of achieving certain desired QoS objectives;
4. Actions: An action in the model is to allocate a time slot on a service or node resource to a task. Actions are regarded as control factors, which can be employed by an Grid scheduling system to influence the state of the environment: $A = A_q|1 \le q \le N_A$, where $N_A = |A|$ is the total number of feasible actions. Each action will take a value of "true" or "false" at each time step once the scheduling system determines a strategy. That is, $P_k\{A_q = 1\} = 1$ if action A_q is activated at time step t_k; otherwise, $P_k\{A_q = 1\} = 0$. Without loss of generality, we assume that $(r_q + 1) \ll 2^T$ feasible choices for action A_q from a domain $\Omega_{A_q} = \{a_q^0, a_q^1, a_q^2, \wedge, a_q^{r_q}\}$ are available. Each element $a_q^i(0 \le i \le r_q)$i n this set maps to a time series based actions '$a_q^i(t_1) \wedge a_q^i(t_T)$' with $a_q^i(t_k) \in \{0, 1\}(0 \le k \le T)$ which representing task q is mapped to resource node i at time t_k. Let C_{a_q} be the cost of selecting schedule a_q for action A_q. A strategy under a given initial environment state $S(t_0)$ is a set of series for all the actions: $R = \{(a_1, a_2, \wedge, a_{N_A})|a_q \in \Omega_{A_q}\}$,Thus, the cost of the strategy is $C_R = \sum_{q=1}^{N_A} c_{a_q}$.

5. Intermediate states are defined to differentiate those states that are not desired finishing state, but are useful in connecting the actions and events to the QoS objectives. They can be regarded as the predefined states of a workflow. All the intermediate states form a set $I = \{I_m | 1 \leq m \leq N_I\}$ with $N_I = |I|$. Fig. 1 shows that only desired states and intermediate states are connected by diachronic edges;
6. Direct influence dependencies between all the objects of the system and their mechanisms are specified by conditional probability tables (CPTs) in Bayesian networks, which can attained from the priori analysis of system feed sources.
7. The total resource available for the Grid is constrained by C_{budget}.

Conceptually, the problem is to achieve the desired objective states $\{O\}$ with a high probability at specified times. The mathematical formulation of the scheduling strategy optimization problem is as follows:

$$\begin{aligned}\max_S(P\{O(t_k)|S(t_0), R\}) &= \max_S(P\{O_1(t_{O_1})O_2(t_{O_2}) \wedge O_{N_O}(t_{O_N})|S(t_0), R\}) \\ &= \max_S\left(\prod_{n=1}^{N_O} P\{O_n(t_{O_n})|S(t_0), R\}\right) \qquad (1)\end{aligned}$$

Subject to:

$$C_R = \sum_{q=1}^{N_A} C_{a_q} \leq C_{budget} \qquad (2)$$

3 Applying DBNs to a Robust Scheduler

3.1 Framework of the Solution Approach

As shown in Fig. 2, We combine concepts of robust design, DBNs and genetic optimization algorithms to solve the scheduling optimization problem. DBNs integrated with probability evaluation algorithms are used to model the dynamics of the Grid resources and to calculate the probability of desired QoS objectives at specified deadline. Monte Carlo runs are made to account for uncertainty in system parameters in the inner loop of DBN. That is, disturbances are introduced by randomly choosing DBN parameters (prior pmfs of events and conditional probabilities). In each Monte Carlo run, DBN will evaluate the joint probability of achieving the desired QoS objectives. The histogram provided by the results of Monte Carlo runs is approximated as a Gaussian density (based on the Central Limit Theorem) with sample mean and sample variance. Using the sample mean and variance and following robust design techniques, a signal-to-noise ratio (SNR) is computed; this criterion maximizes the probability of achieving the desired QoS objectives while minimizing its variability. A genetic algorithm is employed in the outer loop to optimize the scheduling strategies.

Conceptually, the probability of achieving the desired QoSs is a function of actions A, exogenous events B and time t_k, that is $P(O) = f(A, B, t_k)$. In iterations of the genetic algorithm, since we choose candidate scheduling strategies,

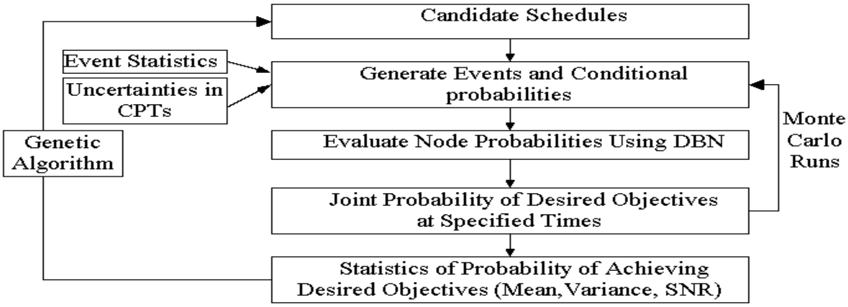

Candidate Schedules
Event Statistics
Uncertainties in CPTs
Generate Events and Conditional probabilities
Genetic Algorithm
Evaluate Node Probabilities Using DBN
Monte Carlo Runs
Joint Probability of Desired Objectives at Specified Times
Statistics of Probability of Achieving Desired Objectives (Mean,Variance, SNR)

Fig. 2. Framework Overview

thereby fixing the values of A, the probability will be a function of events B and time t_k, that is, $P(O|A) = g(B, t_k)$. Then, in each Monte Carlo run of DBN inference, for the given sequences of actions A, we estimate the occurrence probabilities of exogenous events B. Consequently, from a single Monte Carlo run, we have $P(O|A, B) = h(t_k)$. We can see that Monte Carlo runs inside the DBN inference makes it possible to measure the robustness of a schedule in an uncertain environment in terms of the signal-to-noise ratio.

3.2 Probability Propagation Through DBN

Based on the DBN of Fig. 1, we extended from the initial static Bayesian Network by introducing dummy nodes for all the intermediate and desired states. Dummy nodes are defined as: $V_i^0 = \{v_i^0 | v_i \in I \cup O\}$ with $P_{k+1}\{v_i^0\} = P_k\{v_i\}$. The probability will propagate vertically from causal nodes to effect nodes, and propagate horizontally from one time step to the next as follows:
(1) Set the initial pmfs of nodes:$P_1\{v_i^0\} = P_0\{v_i\}$ based on known $S(t_0)$;
(2) Let $k = 1$;
(3) Select an action strategy: $R = \{(a_1, a_2, \cdots, a_{N_A}) | a_q \in \Omega_{A_q}, 1 \leq q \leq N_A\}$, where if $a_q(t_k) = 1$, set $P_k(A_q = 1) = 1$;else $P_k(A_q = 1) = 0$;
(4) Randomly select probability mass functions of exogenous events $P_k\{B_j\}$;
(5) Calculate probability mass functions of the intermediate and desired objectives using Bayesian model averaging:

$$P_k\{v_i\} = \sum_{\pi(v_i), v_i^0} P\{v_i | \pi(v_i), v_i^0\} \cdot P_k\{\pi(v_i)\} P_k\{v_i^0\}, v_i \in I \cup O$$

Where $\pi(v_i)$ is the possible instantiation of the parent nodes of v_i;
(6) Propagate the current probability mass functions to the next time step:

$$P_{k+1}\{v_i^0\} = P_k\{v_i\};$$

(7) Let $k = k + 1$. If $k \leq T$, go back to step (3); otherwise, stop.

3.3 Scheduling Optimization with GA

Our implementation of GA for strategy optimization is explained in detail in the following.

1) Chromosome Representation: In section 2, the feasible actions are given by $A = \{1 \leq q \leq N_A\}$ with $A_q \in \{a_q^0, a_q^1, \wedge, a_q^{r_q}\}$. Thus, the chromosome can be represented as a series of integer genes $= (\omega_1\omega_2 \cdots \omega_q)$,where $0 \leq \omega_q \leq r_q$. If $\omega_q = 1$, a_q^1 is picked for A_q, that is, task q is assigned to node or service 1, If $\omega_q = 2$, a_q^2 is picked for A_q, and so on. In other words, the gene is coded to represent the assignment of a task to resource at what time steps, and the whole chromosome is a code representing a schedule [11,12].

2) Population Initialization: It is the first step in GA. In our problem, we generate the initial schedule randomly. Thus, for any individual $\omega = (\omega_1\omega_2 \wedge \omega_q)$, in the initial population, $\omega_q(1 \leq q \leq r_q)$ satisfying the constraints of cost and resource budgets is selected from $\{0, 1, \wedge, r_q\}$.

3) Fitness function: DBN performs the inner loop inference to compute the evaluation function for GA. The evaluation function will map the population candidate into a partially ordered set, which will be input to the next step, i.e., population selection. DBN is used to obtain the probability of achieving the desired effects at certain time slices for a given strategy $P\{O_1(t_{O_1})O_2(t_{O_2}) \wedge O_{N_O}(t_{O_N})|S(t_0), R\}$. In a noisy environment, this probability is a random variable because of the uncertainty in the statistical description of exogenous events B. In the DBN loop, we generate a histogram of this probability via Monte Carlo runs, the sample mean and variance are computed via:

$$\mu = \frac{1}{M}\sum_{i=1}^{M} P_i\{O_1(t_{O_1})O_2(t_{O_2}) \wedge O_{N_O}(t_{O_N})|S(t_0), R\} \tag{3}$$

$$\sigma^2 = \frac{1}{M-1}\sum_{i=1}^{M} (P_i\{O_1(t_{O_1})O_2(t_{O_2}) \wedge O_{N_O}(t_{O_N})|S(t_0), R\} - \mu)^2 \tag{4}$$

Signal-to-noise ratio (SNR) provides a measure of goodness or fitness of a strategy. SNR is computed via:

$$SNR = -10log_{10}\left[\frac{1}{\mu^2}\left(1 + 3\frac{\sigma^2}{\mu^2}\right)\right] \tag{5}$$

The optimized evaluation function, SNR, corresponds to a schedule that has high probability of success, and that is also robust to changes in the environment (unforeseen events, uncertainty in parameters, etc.).

4)Selection function: Since SNR is negative in our case, we use the normalized geometric ranking method as follows. When population is $\{S_i|1 \leq i \leq N_P\}$, the probability of selecting S_i is defined as:

$$P(\text{select } S_i) = \frac{q(1-q)^{r-1}}{1-(1-q)^{N_P}} \tag{6}$$

Where q is a specified probability of selecting the best individual, r is the rank of the individual with the best individual ranked at '1'.

5) Genetic operators: Mutation and crossover are basic operators to create new population based on individuals in the current generation. Since our chromosome is a series of integers, we employ the following genetic operators to generate individuals for the new strategy:

$$\text{Uniform mutation: } \omega_q' = \begin{cases} U(0, r_q), & \text{if the } q^{th} \text{ gene is selected for mutation} \\ \omega_q, & \text{otherwise} \end{cases} \tag{7}$$

Integer-valued simple crossover generates a random number l from $U(1, N_A)$, and creates two new strategies ω_i' and ω_j' through interchange of genes as follows:

$$\omega_i' = \begin{cases} \omega_i, & \text{if } (i < l) \\ \omega_j, & \text{else} \end{cases} \qquad \omega_j' = \begin{cases} \omega_j, & \text{if } (i < l) \\ \omega_i, & \text{else} \end{cases} \tag{8}$$

6) Termination criteria: We Define a maximum number of generations and stop at a predefined generation.

4 Experimental Results and Analysis

To validate our work, we conducted experiments in a small eight-node Grid of which includes 2 sensors (sen1, sen2), one storage server (st1), two IBM X260 servers (ws1 ,ws2) and three workstations (ws3, ws4, ws5). We simulate a simple workflow job including four tasks tk1, tk2, tk3, tk4.

The feasible actions of the workflow are to schedule tasks to perform their work: A_1— tk1 execute FFT1 algorithm on the signal data read from sen1; A_2— tk2 execute FFT2 algorithm on the signal data read from sen2; A_3— tk3 execute correlation analysis algorithm on the frequency data read from tk1 and tk2 and save the result to st1; A_4— tk4 read the result from st1 and transfer it to a remote monitor.

We generate three exogenous events, where: B_1—network channel to sen1 is congested; B_2—network channel to sen2 is congested; B_3—storage server st1 is overloaded. Each event has an approximate probability, based on the result

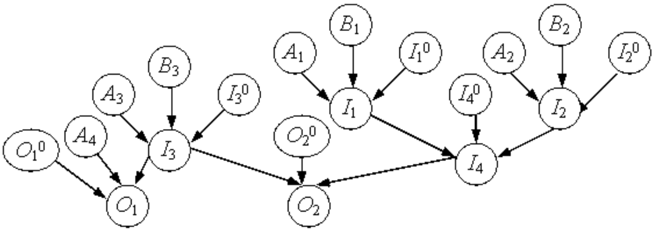
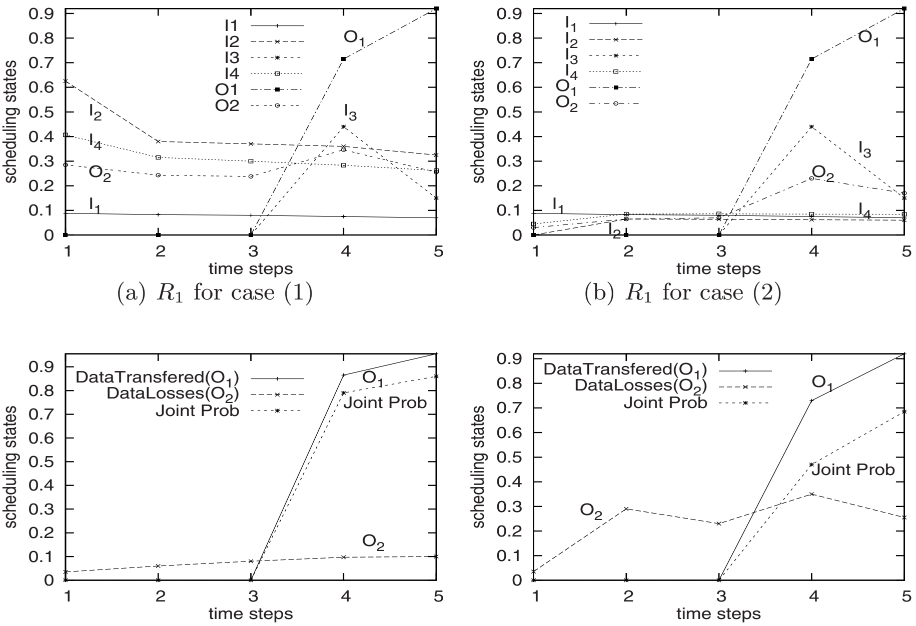


Fig. 3. Workflow Scenario

of benchmark software on the network and computers if in real application. However, the time at which the events happen is unpredictable.

Desired QoS objectives are defined as: O_1— the result is transferred to remote monitor; O_2—keep the measurement error to a minimum.

The following intermediate states are designed to connect actions or events to the desired objectives: I_1 —measure from sen1; I_2 —measure from sen2; I_3 —store the correlation result to st1; I_4 — the error in FFT algorithm. Since the events may happen at arbitrary times, the problem is, given a possible combination of events, which schedule will maximize the probability of achieving the desired objectives. Consider two cases: (1) sen1 and sen2 are congested at time t_1. Whenever tk4 will transfer data, the storage is overloaded immediately; (2) sen1 is congested at time t_1 and sen2 is congested at time t_2; the st1 will act as in case (1). The results from these two cases under schedule $R_1(a_1^1, a_2^2, a_3^2, a_4^1)$ are illustrated in Fig. 4(a) and Fig. 4(b), respectively. In this scenario, we assumed the probabilities of events happen are: $P\{B_1 = 1\} = 0.8, P\{B_2 = 1\} = 0.7, P\{B_3 = 1\} = 0.8$. Since sen1 and sen2 are separately congested in case (2), the FFT algorithm will be under a moderate packet loss probability. On the other hand, in case (1), the combination of two events may put the tk1 and tk2 in the measurement under severe loss probability. Thus, the computation error will be higher in case (1).



(a) R_1 for case (1)

(b) R_1 for case (2)

(c) R_2 for case (2)

(d) R_3 for case (2)

Fig. 4. Simulation Result

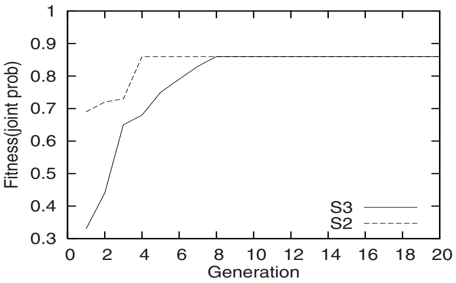


Fig. 5. Schedule Optimization through GA

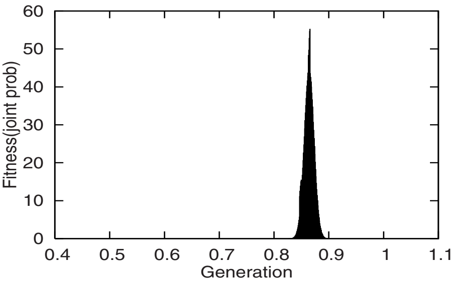


Fig. 6. 1000 Monto Carlo Runs for R_2

Now, we focus on case (2) to see which action strategy will be better. Comparing R_1 with $R_2(a_1^1, a_2^2, a_3^1, a_4^1)$ and $R_3(a_1^1, a_2^3, a_3^2, a_4^1)$, we can see from Fig. 4(c) and Fig. 4(d) that R_2 is the best among these three schedules because all the data are immediately processed. As a consequence, the computation error due to packet loss is low. Fig. 4(c) and Fig. 4(d) depict the joint probability of achieving both of the desired objectives: $P\{O_1(t_k) = 1, O_2(t_k) = 0\}$. Fig. 5 is the result from genetic algorithm, where we use $P\{O_1(5) = 1, O_2(5) = 0\}$ as a fitness measurement.

Table 1. Potential Actions for GA

Action	A_1					A_2					A_3		A_4	
	a_1^1	a_1^2	a_1^3	a_1^4	a_1^5	a_2^1	a_2^2	a_2^3	a_2^4	a_2^5	a_3^1	a_3^2	a_4^1	a_4^2
t1	1	0	0	0	0	1	0	0	0	0	0	0	0	0
t2	0	1	0	0	0	0	1	0	0	0	0	0	0	0
t3	0	0	1	0	0	0	0	1	0	0	0	0	0	0
t4	0	0	0	1	0	0	0	0	1	0	1	0	1	0
t5	0	0	0	0	1	0	0	0	0	1	0	1	1	1

Additionally, we consider a scenario where the data from benchmark is noisy. We suppose $P\{B_1(t_1) = 1\} = P_1$, $P\{B_2(t_2) = 1\} = P_2$, $P\{B_3(t_4) = 1\} = P_3$, where P_1 is uniformly distributed between [0.6,1], P_2 is uniformly distributed between [0.5,0.9] and P_3 uniformly distributed between [0.7,0.9]. Results of $P\{O_1(5) = 1, O_2(5) = 0\}$ from 1000 Monte Carlo runs are shown in the histograms of Fig. 6, with the Gaussian distribution superimposed. The sample mean and standard deviation are 0.8641 and 0.0089, respectively. The two-sided 95% confidence region of this schedule is (0.8467, 0.8816). A narrower confidence region means better control of the environment.

5 Conclusion

This paper introduced a general methodology, based on an integration of dynamic Bayesian networks and the genetic algorithms, to optimize schedules for Grid. The main contributions of this paper are: the use of DBN to compute time-dependent probability propagation for desired objectives; use of GA to optimize job scheduling; introduction of signal-to noise ratio (SNR) as a measure of robustness of a strategy in an uncertain environment.

References

1. Real, R., Yamin, A., da Silva, L., Frainer, G., Augustin, I., Barbosa, J., Geyer, C.: Resource scheduling on grid: handling uncertainty. In: Proceeding of the Fourth International Workshop on Grid Computing, pp. 205–207 (2003)
2. Kurowski, K., Nabrzyski, J., Oleksiak, A., Weglarz, J.: Multicriteria aspects of Grid resource management. Grid resource management: state of the art and future trends table of contents, 271–293 (2004)
3. Domagalski, P., Kurowski, K., Oleksiak, A., Nabrzyski, J., Balaton, Z., Gombás, G., Kacsuk, P.: Sensor Oriented Grid Montoring Infrastructures for Adaptaive Multicriteria Resource Management Strategies. In: Proceedings of the 1st CoreGrid Workshop, pp. 163–173 (2005)
4. Smith, W., Taylor, V., Foster, I.: Using Run-Time Predictions to Estimate Queue Wait Times and Improve Scheduler Performance. In: Proceedings of the IPPS/SPDP '99 Workshop on Job Scheduling Strategies for Parallel Processing, pp. 202–219 (1999)
5. Smith, W., Foster, I., Taylor, V.: Predicting Application Run Times Using Historical Information. Lecture Notes on Computer Science, pp. 122–142 (1998)
6. Sample, N., Keyani, P., Wiederhold, G.: Scheduling under uncertainty: planning for the ubiquitous grid. In: Proceedings of the 5th International Conference on Coordination Models and Languages, pp. 300–316 (2002)
7. Li, J., Yahyapour, R.: Learning-Based Negotiation Strategies for Grid Scheduling. In: Proceedings of CCGRID'06, pp. 576–583 (2006)
8. Zeinalipour-Yazti, D., Neocleous, K., Georgiou, C., Dikaiakos, M.: Managing failures in a grid system using failrank. Technical Report TR-2006-04, Department of Computer Science, University of Cyprus (2006)
9. Thomas, N., Bradley, J., Knottenbelt, W.: Stochastic analysis of scheduling strategies in a Grid-based resource model. IEEE Proceedings Software 151, 232–239 (2004)
10. Santos, L., Proenca, A.: Scheduling under conditions of uncertainty: a bayesian approach. In: Proceedings of the 5th International Conference on Coordination Models and Languages, pp. 222–229 (2004)
11. Kim, S., Weissman, J.: A genetic algorithm based approach for scheduling decomposable data grid applications. In: Proceedings of 2004 International Conference on Parallel Processing, pp. 406–413 (2004)
12. Di Martino, V., Mililotti, M.: Sub optimal scheduling in a grid using genetic algorithms. Parallel Computing 30, 553–565 (2004)