OPTIMIZATION OF DISCRETE EVENT DYNAMIC SYSTEMS
Based on Single Sample Path Analysis
(CX1a-02)

Prashant Goel
Supervisor: Prof. Xiren Cao
Overview

This project verifies through simulation, the correctness of a newly proposed optimization algorithm that has been developed from two well-known research areas: Markov decision process (MDP) theory and single sample path-based sensitivity analysis, by Prof. Xiren Cao, HKUST. It is shown that in parameterized systems, “it can be applied to a part of the system that is affected by the changes of the values of the parameters”. One main closed network manufacturing process was simulated to verify the proposed relations. The problem was converted into an embedded Markov chain and then discrete event simulation was performed, using Java. The theoretical results were computed using C and were used to test and evaluate the simulation. The simulation with the theoretical results was then used to verify the relations. It was concluded that performance sensitivities could be obtained without estimating the potentials for all the states and without knowing the transition probability matrix. Optimal performance can be achieved by simply analyzing a single sample path of a system.

The Algorithm

The algorithm avoids finding the transition probability matrix to evaluate a system. Finding the transition probability matrix can be very complex in complex systems. The system is modeled as an embedded Markov chain.

The simulation takes two sets of input parameters and will give their performance difference, $\eta' - \eta$. $\eta$ and $\eta'$ are the performance measures depending on how they are defined.
The System Simulated

Figure 1: The Closed Manufacturing System simulated in the project.

Description:
- consists of two components, Machine 1(M₁) and Machine 2(M₂).
- N parts circulating between the two machines
- M₁ can perform three operations (1, 2 and 3); service times are exponentially distributed with rates λ₁, λ₂ and λ₃, respectively.
- M₂ can perform only one operation. The service rate is again exponentially distributed, with rate λ₄.
- P₁ and P₂ denote the probabilities a part will leave M₁ after completing operation 1 or 2, respectively.
- A part will move from operation 1 to operation 2 with probability 1 - P₁ and likewise from operation 2 to operation 3 with probably 1 - P₂.
- After a part departs M₁ it can either go back to M₁ with probability 1 - θ(n) or to M₂ with probability θ(n), where n is the number of parts in the buffer for M₁, the part being processed by M₁ inclusive.

The algorithm is verified through the relation:
\[ η' - η = \sum_{n=1}^{N} p'(n)[θ'(n) - θ(n)]d[(n,1),(n-1,1)], \]

where, η is the average number of parts at M₁. θ(n) and θ'(n) are two different sets of probabilities.

The left-hand side is calculated theoretically and the right-hand side through simulation.
\[ p(n) = \frac{\text{No. of Service Completions at M₁ with n customers}}{\text{No. of Total Transitions}} \]

and \( d[(n,1),(n-1,1)] \) is found by averaging the number of parts at M₁ over the periods from state (n-1,1) to (n,1). A state (n,i), is defined as n parts at M₁ and operation i being performed by M₁.
Results

| 1st Run, N = 5 | λ₁ = 5  λ₂ = 3  λ₃ = 4  λ₄ = 2 | p₁ = 0.3  p₂ = 0.6 | θ₁ = 0.4  θ₂ = 0.7  θ₃ = 0.5  θ₄ = 0.8  θ₅ = 0.2 | θ₁′ = 0.68  θ₂′ = 0.52  θ₃′ = 0.81  θ₄′ = 0.3  θ₅′ = 0.1 |
| 2nd Run, N = 10 | λ₁ = 7  λ₂ = 9  λ₃ = 6  λ₄ = 4 | p₁ = 0.7  p₂ = 0.2 | θ₁ = 0.8  θ₂ = 0.8  θ₃ = 0.7  θ₄ = 0.4  θ₅ = 0.5  θ₆ = 0.2  θ₇ = 0.6  θ₈ = 0.3  θ₉ = 0.1  θ₁₀ = 0.9 | θ₁′ = 0.68  θ₂′ = 0.52  θ₃′ = 0.81  θ₄′ = 0.3  θ₅′ = 0.1  θ₆′ = 0.55  θ₇′ = 0.34  θ₈′ = 0.19  θ₉′ = 0.27  θ₁₀′ = 0.38 |

Table 1: The values of the parameters.

| 1st Run | Theoretical (η' - η) | Simulated $\sum_{n=1}^{N} p(n)[\theta'(n) - \theta(n)]\delta([n,1),(n-1,1)]$ |
| 2nd Run | 0.424594329 | 0.42521563056 |
| 2nd Run | 0.107920839 | 0.106285857 |

Table 2: Results. Algorithm verified. Left hand side computed theoretically, right hand side simulated.

The theoretical and simulated values match. The simulation was successful and the algorithm can be used to evaluate parameterized systems.