Solving Markov decision processes for network-level post-hazard recovery via simulation optimization and rollout

(Invited Paper)

Yugandhar Sarkale[1], Saeed Nozhati[2], Edwin K. P. Chong[1], Bruce R. Ellingwood[2], Hussam Mahmoud[2]

***Abstract*— Computation of optimal recovery decisions for community resilience assurance post-hazard is a combinatorial decision-making problem under uncertainty. It involves solving a large-scale optimization problem, which is significantly aggravated by the introduction of uncertainty. In this paper, we draw upon established tools from multiple research communities to provide an effective solution to this challenging problem. We provide a stochastic model of damage to the water network (WN) within a testbed community following a severe earthquake and compute near-optimal recovery actions for restoration of the water network. We formulate this stochastic decision-making problem as a Markov Decision Process (MDP), and solve it using a popular class of heuristic algorithms known as *rollout*. A simulation-based representation of MDPs is utilized in conjunction with rollout and the Optimal Computing Budget Allocation (OCBA) algorithm to address the resulting stochastic simulation optimization problem. Our method employs non-myopic planning with efficient use of simulation budget. We show, through simulation results, that rollout fused with OCBA performs competitively with respect to rollout with total equal allocation (TEA) at a meagre simulation budget of 5-10% of rollout with TEA, which is a crucial step towards addressing large-scale community recovery problems following natural disasters.**

I. INTRODUCTION

Natural disasters have a significant impact on the economic, social, and cultural fabric of affected communities. Moreover, because of the interconnected nature of communities in the modern world, the adverse impact is no longer restricted to the locally affected region, but it has ramifications on national or international scale. Among other factors, the occurrence of such natural disasters is on the rise owing to population growth and economic development in hazard-prone areas [1]. Keeping in view the increased frequency of natural disasters, there is an urgent need to address the problem of community recovery post-hazard. Typically, the resources available to post-disaster planners are limited and relatively small compared to the impact of the damage. Under these scenarios, it becomes imperative to assign limited resources to various damaged components in the network optimally to support community recovery. Such an assignment must also consider multiple objectives and cascading effects due to the interconnectedness of various networks within the community and must also successfully adopt previous proven methods and practices developed by expert disaster-management planners. Holistic approaches addressing various uncertainties for network-level management of limited resources must be developed for maximum effect. Civil infrastructure systems, including power, transportation, and water networks, play a critical part in post-disaster recovery management. In this study, we focus on one such critical infrastructure system, namely the water networks (WN), and compute near-optimal recovery actions, in the aftermath of an earthquake, for the WN of a test-bed community.

Markov decision processes (MDPs) offer a convenient framework for representation and solution of stochastic decision-making problems. Exact solutions are intractable for problems of even modest size; therefore, approximate solution methods have to be employed. We can leverage the rich theory of MDPs to model recovery action optimization for large state-space decision-making problems such as our. In this study, we employ a simulation-based representation and solution of MDP. The near-optimal solutions are computed using an approximate solution technique known as *rollout*. Even though state-of-the-art hardware and software practices are used to implement the solution to our problem, we are faced with the additional dilemma of computing recovery actions on a fixed simulation budget without affecting the solution performance. Therefore, any prospective methodology must incorporate such a limitation in its solution process. We incorporate the Optimal Computing Budget Allocation (OCBA) algorithm into our MDP solution process [2], [3] to address the limited simulation budget problem.

II. TESTBED CASE STUDY

A. Network Characterization

This study considers the potable water network (WN) of Gilroy, CA, USA as an example to illustrate the proposed methodology. Gilroy, located 50 kilometers (km) south of the city of San Jose, CA is approximately 41.91 km^2 in area, with a population of 48,821 [4]. We divide our study area into 36 grid regions to define the properties of infrastructure systems, household units, and the population. Our model of the community maintains adequate detail to study the

[1]Yugandhar Sarkale and Edwin K. P. Chong are with Department of Electrical and Computer Engineering, Colorado State University, Fort Collins, CO 80523-1373, USA Yugandhar.Sarkale, Edwin.Chong@colostate.edu

[2]Saeed Nozhati, Bruce R. Ellingwood and Hussam Mahmoud are with the Department of Civil and Environmental Engineering, Colorado State University, Fort Collins, CO 80523-1372, USA Saeed.Nozhati, Bruce.Ellingwood, Hussam.Mahmoud@colostate.edu

This work was supported by the National Science Foundation under Grant CMMI-1638284. This support is gratefully acknowledged. Any opinions, findings, conclusions, or recommendations presented in this material are solely those of the authors and do not necessarily reflect the views of the National Science Foundation.

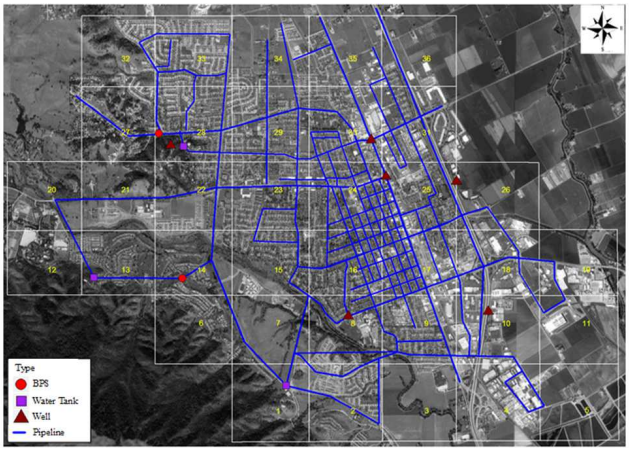

Fig. 1. The modeled Water Network of Gilroy

performance of the WN at a community level under severe earthquakes. The potable water of Gilroy is provided only by the Llgas sub-basin [5]. The potable water wells, located in wood-frame buildings, pump water into the distribution system. The Gilroy municipal water pipelines range from 102 mm to 610 mm in diameter [5]. In this study, a simplified aggregated model of WN of Gilroy adopted from [5] is modeled. This model shown in Fig. 1, includes six water wells, two booster pump stations (BPS), three water tanks (WT), and the main pipelines.

B. Seismic Hazard Simulation

The San Andreas Fault (SAF), which is near Gilroy, is a source of severe earthquakes. In this study, we assume that a seismic event of moment magnitude $Mw = 6.9$ occurs at one of the closest points on the SAF projection to downtown Gilroy with an epicentral distance of approximately 12 km. Ground motion prediction equations (GMPE) determine the conditional probability of exceeding ground motion intensity at specific geographic locations within Gilroy for this earthquake.

The Abrahamson et al. [6] GMPE is used to estimate the Intensity Measures (IM) and associated uncertainties. Peak Ground Acceleration (PGA) is considered for the above-ground WN facilities and wells, whereas Peak Ground Velocity (PGV) is considered as IM of pipelines.

C. Fragility and Restoration Assessment of Water Network

The physical damage to WN components can be assessed by seismic fragility curves. We use the fragility curves presented in HAZUS-MH [7] for wells, water tanks, and pump stations based on the IM of PGA. This study adopts the assumptions in [8] for water pipelines. The failure probability of a pipe is bounded as follows:

$$1 - G_{\varepsilon PGV}(-CL\mu_{PGV}) \leq E[P_f] \leq 1 - E[exp(-CL\mu_{PGV})] \quad (1)$$

where P_f is the failure probability of a pipe, L is the length of pipe, μ_{PGV} is the average PGV for the entire length of the water main, and $G(\cdot)$ is the moment-generating function of εPGV (the residual of the PGV). The term C for water pipe segment i is $C = K \times 0.00187 \times PGV_i$, where K is a coefficient determined by the pipe material, diameter, joint type, and soil condition based on the guidelines prepared by the American Lifeline Alliance [9]. Adachi and Ellingwood [8] demonstrated that the Upper Bound (UB) and exact solutions (1) are close enough so that in practical applications the UB assessment (conservative evaluation) can be used.

Repair crews, replacement components, and tools are considered as available units of resources to restore the damaged components of WN following the hazard. One unit of resources is required to repair each damaged component [10], [11]. However, the available units of resources are limited and depend on the capacities and policy of the entities within the community. To restore the WN, the restoration times based on exponential distributions synthesized from HAZUS-MH [7] are used, as summarized in Table I. The pipe-restoration time in the WN is based on repair rate or number of repairs per kilometer.

TABLE I
THE EXPECTED REPAIR TIMES (UNIT: DAYS)

	Damage States			
Component	Minor	Moderate	Extensive	Complete
Water tanks	1.2	3.1	93	155
Wells	0.8	1.5	10.5	26
Pumping plants	0.9	3.1	13.5	35

III. PROBLEM DESCRIPTION AND SOLUTION

A. MDP Framework

We provide a brief description of MDP [12] for the sake of completeness. An MDP is a controlled dynamical process useful in modelling of wide range of decision-making problems. It can be represented by the 4-tuple $\langle S, A, T, R \rangle$. Here, S represents the set of states, and A represents the set of actions. Let $s, s' \in S$ and $a \in A$; then T is the state transition function, where $T(s,a,s') = P(s' \mid s,a)$ is the probability of going into state s' after taking action a in state s. R is the reward function, where $R(s,a,s')$ is the reward received after transitioning from s to s' as a result of action a. In this study, we assume that $|S|$ and $|A|$ are finite; R is bounded and real-valued and a deterministic function of s, a and s'. Implicit in our presentation are also the following assumptions: First order Markovian dynamics (history independence), stationary dynamics (reward function is not a function of absolute time), and full observability of the state space (outcome of an action in a state might be random, but we know the state reached after action is completed). In our study, we assume that we are allowed to take recovery actions (decisions) indefinitely until all the damaged components of our modeled problem are repaired (infinite-horizon planning). In this setting, we have a stationary policy π, which is defined as $\pi : S \to A$. Suppose that decisions are made at discrete-time t; then $\pi(s)$ is the action to be taken in state s (regardless of time t). Our

objective is to find an optimal policy π^*. For the infinite-horizon case, π^* is defined as

$$\pi^* = \arg\max_{\pi} V^{\pi}(s_0), \tag{2}$$

where

$$V^{\pi}(s_0) = E\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, \pi(s_t), s_{t+1})\right] \tag{3}$$

is called the value function for a fixed policy π, and $0 < \gamma < 1$ is the discount factor. Note that the optimal policy is independent of the initial state s_0. Also, note that we maximize over policies π, where at each time t the action taken is $a_t = \pi(s_t)$. Stationary optimal policies are guaranteed to exist for discounted infinite-horizon optimization criteria [13]. To summarize, our presentation is for infinite-horizon discrete-time MDPs with the discounted value as our optimization criterion.

B. MDP Solution

A solution to an MDP is the optimal policy π^*. We can obtain π^* with linear programming or dynamic programming. In the dynamic programming regime, there are several solution strategies, namely value iteration, policy iteration, modified policy iteration, etc. Unfortunately, such exact solution algorithms are intractable for large state and actions spaces. We briefly mention here the method of value iteration because it illustrates the Bellman's equation [14]. Studying Bellman's equation is useful for defining Q value function. Q value function will play a critical role in describing the rollout algorithm. Let V^{π^*} denote the optimal value function for some π^*; Bellman showed that V^{π^*} satisfies:

$$V^{\pi^*}(s) = \max_{a \in A(s)} \left\{ \gamma \cdot \sum_{s'} P(s' \mid s,a) \cdot \left[V^{\pi^*}(s') + R(s,a,s') \right] \right\}. \tag{4}$$

Equation (4) is known as the Bellman's optimality equation, where $A(s)$ is the set of possible actions in any state s. The value iteration algorithm solves (4) by using Bellman backup repeatedly, where Bellman backup is given by:

$$V_{i+1}(s) = \max_{a \in A(s)} \left\{ \gamma \sum_{s'} P(s' \mid s,a) \cdot \left[V_i(s') + R(s,a,s') \right] \right\}. \tag{5}$$

Bellman showed that $\lim_{i \to \infty} V_i = V^{\pi^*}$, where V_0 is initialised arbitrarily.[1] Next, we define the Q value function of policy π:

$$Q_{\pi}(s,a) = \gamma \cdot \sum_{s'} P(s' \mid s,a) \cdot \left[V^{\pi}(s') + R(s,a,s') \right]. \tag{6}$$

The Q value function of any policy π gives the expected discount reward in the future after starting in some state s, taking action a and following policy π thereafter. Note that this is the inner term in (4).

[1]On a historical note, Lloyd Shapely's paper [15] included the value iteration algorithm for MDPs as a special case, but this was recognised only later on [16].

C. Simulation-Based Representation of MDP

We now briefly explain the simulation-based representation of an MDP [17]. Such a representation serves well for large state and action spaces, which is a characteristic feature of many real-world problems. When $|S|$ or $|A|$ is large, it is not feasible to represent T and R in a matrix form. A simulation-based representation of an MDP is a 5-tuple $\langle S, A, R, T, I \rangle$, where S and A are as before, except $|S|$ and $|A|$ are large. Here, R is a stochastic real-valued bounded function that stochastically returns a reward r when input s and a are provided, where a is the action applied in state s. T is a simulator that stochastically returns a state s' when state s and action a are provided as inputs. I is the stochastic initial state function that stochastically returns a state according to some initial state distribution. R, T, and I can be thought of as any callable library functions that can be implemented in any programming language.

D. Problem Formulation

After an earthquake event occurs, the components of the water network remain undamaged or exhibit one of the damage states as shown in Table I. Let L' be the total number of damaged component at t. Let t_c represent the decision time when all components are repaired. There is a fixed number of resource units (M) available to the decision maker. At each discrete-time t, the decision maker has to decide the assignment of unit of resource to the damaged locations; each component cannot be assigned more than one resource unit. When the number of damaged locations is less than the number of units of resources (because of sequential application of repair actions, or otherwise), we retire the extra unit of resources so that M is equal to the number of damaged locations.

- *States S:* Let s_t be the state of the damaged components of the system at time t; then s_t is a vector of length L', $s_t = (s_t^1, \ldots, s_t^{L'})$, and s_t^l is one of the damaged state in Table I where $l \in \{1, \ldots, L'\}$.
- *Actions A:* Let a_t denote the repair action to be carried out at time t. Then, a_t is a vector of length L', $a_t = (a_t^1, \ldots, a_t^{L'})$, and $a_t^l \in \{0,1\}\ \forall l,t$. When $a_t^l = 0$, no repair work is to be carried out at l. Similarly, when $a_t^l = 1$, repair work is carried out at l.
- *Simulator T:* The repair time associated with each damaged location depends on the state of the damage to the component at that location (see Table I). This repair time is random and is exponentially distributed with expected repair times shown in Table I. Given s_t and a_t, T gives us the new state s_{t+1}. We say that a repair action is complete as soon as at least one of the locations where repair work is carried out is fully repaired. Let's denote this completion time at every t by $\hat{t}_t$. Note that it is possible for the repair work at two or more damaged locations to be completed simultaneously. Once the repair action is complete, the units of resources at remaining locations, where repair work was not complete, are also available for

reassignment along with unit of resources where repair was complete. The new repair time at such unrepaired locations is calculated by subtracting $\hat{t}$ from the time required to repair these locations. It is also possible to reassign the unit of resource at the same unrepaired location if it is deemed important for the repair work to be continued at that location by the planner. Because of this reason, preemption of repair work during reassignment is not a restrictive assumption, on the contrary, it allows greater flexibility to the decision maker for planning. Because the repair times are random, the outcomes of repair actions are random as not the same damaged component will be repaired first even if the same repair action a_t is applied in s_t (We would like to stress again that the state-dependent random repair time is exponentially distributed with expected repair times shown in Table I). Hence, our simulator T is stochastic. Alternative formulation where outcome of repair action is deterministic is also an active area of research [18]–[20].

- *Rewards R:* We wish to optimally plan decisions so that maximum people will get water in minimum amount of time. We combine these two competing objectives to define our reward as:

$$R(s_t, a_t, s_{t+1}) = \frac{r}{t_{rep}}, \tag{7}$$

where r is the number of people who have water after action a_t is completed, and t_{rep} is the total repair time (days) required to reach s_{t+1} from any initial state s_0. Note that the total repair time t_{rep}, after an action a_t is completed, is the sum of the completion time $\hat{t}_t$, at each t. Therefore, the state-action dependent definition of the reward function in (7) is based on the time period required to complete an action (completion time $\hat{t}_t$), and captures the time-critical aspect of the recovery actions in its definition, which plays an important part in post-hazard recovery problems. Also, note that our reward function is stochastic because the outcome of our action a_t is random.

- *Initial State I:* We have already described the stochastic damage model of the components for the modeled network in Section II-B and Section II-C. The initial damage states associated with the components will be provided by these models.
- *Discount factor γ:* In our simulation studies, γ is fixed at 0.99.

E. Rollout

The rollout algorithm was first proposed for stochastic scheduling problems by Bertsekas and Castanon [21]. Instead of the dynamic programming formalism [21], we motivate the rollout algorithm in relation to the simulation-based representation of our MDP. Suppose that we have access to a non-optimal policy π, and our aim is to compute an improved policy π'. Then, we have:

$$\pi'(s_t) = \arg\max_{a_t} Q_\pi(s_t, a_t), \tag{8}$$

where the Q function is as defined in (6). If the policy defined in (8) π' is non-optimal, it is a *strict* improvement over π [13]. This result is termed as *policy improvement theorem*. Note that the improved policy π' is generated as a greedy policy w.r.t. Q_π. Unlike the exact solution methods described in Section III-B, we are interested here in computing π' only for the current state. Methods that use (8) as the basis for updating the policy suffer from the *curse of dimensionality*. Before performing the policy improvement step in (8), we have to first calculate the value of Q_π. Calculating the value of Q_π in (8) is known as *policy evaluation*. Policy evaluation is intractable for large or continuous state and action spaces. Approximation techniques alleviate this problem by calculating an approximate Q value function. Rollout is one such approximation technique that utilises monte-carlo simulations. Particularly, rollout can be formulated as an approximate policy iteration algorithm [17], [22]. An implementable (programming sense) stochastic function (simulator) $SimQ(s_t, a_t, \pi, h)$ is defined in such a way that its expected value is $Q_\pi(s_t, a_t, h)$, where h is a finite number representing horizon length. In the rollout algorithm, *SimQ* is implemented by simulating action a_t in state s_t and following π thereafter for $h-1$ steps. This is done for all the actions $a_t \in A(s_t)$. A finite horizon approximation of $Q_\pi(s_t, a_t)$ (termed as $Q_\pi(s_t, a_t, h)$), is required; our simulation would never finish in the infinite horizon case because we would have to follow policy π indefinitely. However, $V^\pi(s_t)$, and consequently $Q_\pi(s_t, a_t)$, is defined over the infinite horizon. It is easy to show the following:

$$|Q_\pi(s_t, a_t) - Q_\pi(s_t, a_t, h)| = \frac{\gamma^h R_{max}}{1-\gamma}. \tag{9}$$

The approximation error in (9) reduces exponentially fast as h grows. Therefore, the h-horizon results apply to the infinite horizon setting, for we can always choose h such that the error in (9) is negligible. To summarize, the rollout algorithm can be presented in the following fashion for our problem:

Algorithm 1 Uniform Rollout (π,h,α,s_t)

```
for i = 1 to n do
    for j = 1 to α do
        ã^{i,j} ← SimQ(s_t, a_t^{i,j}, π, h)        ▷ See algorithm 2
    end for
end for
ã_t^i ← Mean(ã^{i,j})
k ← arg max ã_t^i
return a_t^k
```

In Algorithm 1, n denotes $|A(s_t)|$. Note that Algorithm 2 returns the discounted sum of rewards. When $h = t_c$, we term the rollout as complete rollout, and when $h < t_c$, the rollout is called truncated rollout [21]. It is possible to analyse the performance of uniform rollout in terms of uniform allocation α and horizon depth h [17], [23].

Algorithm 2 Simulator $SimQ(s_t, a_t^{i,j}, \pi, h)$

$s_{t+1} \leftarrow T(s_t, a_t^{i,j})$
$r \leftarrow R(s_t, a_t^{i,j}, s_{t+1})$
for $p = 1$ to $h-1$ **do**
  $s_{t+1+p} \leftarrow T(s_{t+p}, \pi(s_{t+p}))$
  $r \leftarrow r + \gamma^p R(s_{t+p}, \pi(s_{t+p}), s_{t+1+p})$
end for
return r

F. Optimal Computing Budget Allocation

In the previous section, we presented the rollout method for solving our MDP problem. In the case of uniform rollout, we allocate a fixed rollout sampling budget α to each action, i.e., we obtain α number of rollout samples per candidate action to estimate the Q value associated with the action. In the simulation optimization community, this is analogous to total equal allocation (TEA) [24] with a fixed budget α for each simulation experiment (a single simulation experiment is equivalent to one rollout sample). In practice, we are only interested in the best possible action, and we would like to direct our search towards the most promising candidates. Also, for large real-world problems, the simulation budget available is insufficient to allocate α number of rollout samples per action. We would like to get a rough estimate of the performance of each action and spend the remaining simulation budget in refining the accuracy of the best estimates. This is the classic exploration vs. exploitation problem faced in optimal learning and simulation optimization problems.

Instead of a uniform allocation α for each action, non-uniform allocation methods have been explored in the literature pertaining to Algorithm 1 called as *adaptive rollout* [25]. An analysis of performance guarantees for adaptive rollout remains an active area of research [25]–[27]. These non-uniform allocation methods guarantee performance without a constraint on the budget of rollouts. Hence, we explore an alternative non-uniform allocation method that would not only fuse well into our solutions (adaptively guiding the stochastic search) but would also incorporate the constraint of simulation budget in its allocation procedure. Numerous techniques have been proposed in the simulation optimization community to solve this problem. We draw upon one of the best performers [28] that naturally fits into our solution framework—OCBA. Moreover, the probability of correct selection $P\{CS\}$ of an alternative in OCBA mimics finding the best candidate action at each stage in Algorithm 1. Formally, the OCBA problem [29] for Section III-D can be stated as :

$$\max_{N_1,\ldots,N_n} P\{CS\} \text{ such that } \sum_{i=1}^{n} N_i = B, \tag{10}$$

where B represents the simulation budget for determining optimal a_t for s_t at any t, and N_i is the simulation budget for the i^{th} action at a particular t. At each OCBA allocation step (for the definition of the allocation step, see variable l in [29]), barring the best alternative, the OCBA solution assigns an allocation that is directly proportional to the variance of each alternative and inversely proportional to the squared difference between the mean of that alternative and the best alternative.

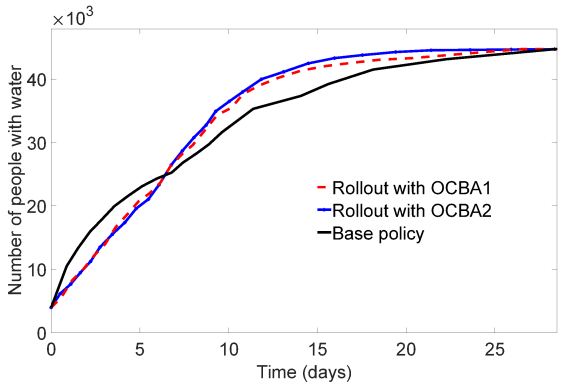

Fig. 2. Performance comparison of rollout vs base policy for 3 units of resources.

Here, we only provide information required to initialize the OCBA algorithm. For a detailed description of OCBA, including the solution to the problem in (10), see [29]. The key initialization variables, for the OCBA algorithm [29], are k, T (not to be confused with T in this paper), Δ, and n_0. The variable k is equal to variable n in our problem. The value of n changes at each t and depends on the number of damaged components and units of resources. The variable T is equal to per-stage budget B in our problem. More information about the exact value assigned to B is described in Section IV. We follow the guidelines specified in [30] to select n_0 and Δ; n_0 in the OCBA algorithm is selected equal to 5, and Δ is kept at 15% of n (within rounding).

IV. SIMULATION RESULTS

We simulate 100 different initial damage scenarios for each of the plots presented in this section. There will be a distinct recovery path for each of the initial damage scenarios. All the plots presented here represent the average of 100 such recovery paths. Two different simulation plots of rollout fused with OCBA are provided in Fig. 2 and Fig. 3. They are termed as rollout with OCBA1 and rollout with OCBA2. The method applied is the same for both cases; only the per-stage simulation budget is different. A per-stage budget (budget at each decision time t) of $B = 5 \cdot n + 5000$ is assigned for rollout with OCBA1 and $B = 5 \cdot n + 10000$ for rollout with OCBA2. Fig. 2 compares the performance of rollout fused with OCBA and base policy. The rollout algorithm is known to have the "lookahead property" [21]. This behavior of the rollout algorithm is evident in the results in Fig. 2, where the base policy initially outperforms the rollout policy, but after about six days the former steadily outperforms the later. Recall, that our objective is to perform repair actions so that maximum people will have water in minimum amount of time. Evaluating the performance of our method in meeting this objective is equivalent to checking the area under the curve of our plots. This area represents

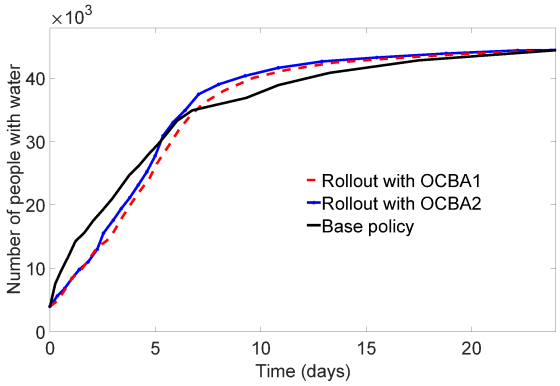

Fig. 3. Performance comparison of rollout vs base policy for 5 unit of resources.

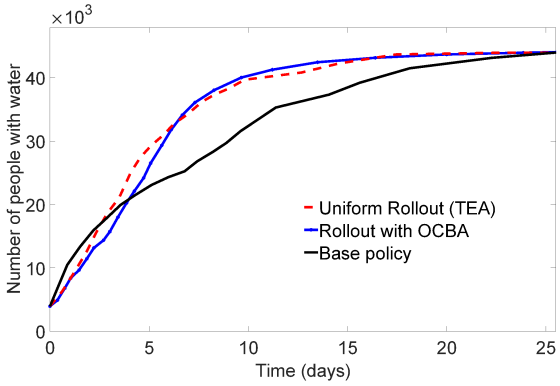

Fig. 4. Performance comparison of uniform rollout (TEA), rollout with OCBA and base policy for 3 units of resources.

the product of the number of people who have water and the number of days for which they have water. A larger area represents that greater number of people were benefitted as a result of the recovery actions. The area under the curve for recovery with rollout (blue and red plots) is more than its base counterpart (black). A per-stage budget increase of 5000 simulations in rollout with OCBA2 with respect to rollout with OCBA1 shows improvements in the recovery process.

In the plots shown in Fig. 3, we use $M = 5$. In the initial phase of planning, it might appear that the base policy outperforms the rollout for a substantial amount of time. However, this is not the case. Note that the number of days for which the base policy outperforms rollout, in both Fig. 2 and Fig. 3, is about six days, but because the number of resource units has increased from three to five, the recovery is faster, giving an illusion that the base policy outperforms rollout for a longer duration. It was verified that the area under the curve for recovery with rollout (blue and red curves) is more than its base counterpart (black curve). Because OCBA is fused with rollout here, we would like to ascertain the exact contribution of the OCBA approach in enhancing the rollout performance.

For the rollout with OCBA in Fig. 4, $B = 5 \cdot n + 20000$, whereas $\alpha = 200$ for the uniform rollout simulations. The recovery as a result of these algorithms outperforms the base policy recovery in all cases. Also, rollout with OCBA performs competitively with respect to uniform rollout despite a meagre simulation budget of 10% of uniform rollout. The area under the recovery process in Fig. 4, as a result of uniform rollout, is only marginally greater than that due to rollout with OCBA. Note that after six days, OCBA slightly outperforms uniform rollout because it prioritizes the simulation budget on the most promising actions per-stage. Rollout exploits this behavior in each stage and gives a set of sequential recovery decisions that further enhances the outcome of the recovery decisions. We would like to once again stress that such an improvement is being achieved at a significantly low simulation budget with respect to uniform rollout. Therefore, these two algorithms form a powerful combination together, where each algorithm consistently and sequentially reinforces the performance of the other. Such synergistic behavior of the combined approach is appealing. Lastly, our simulation studies show that increments in the simulation budget of rollout results in marginal performance improvement for each increment. Beyond a certain increment in the simulation budget, the gain in performance might not scale with the simulation budget expended. A possible explanation is that small simulation budget increase might not dramatically change the approximation of Q value function associated with a state-action pair. Thus, π' in (8) might not show a drastic improvement compared to the one computed by a lower simulation budget (policy improvement based on Q approximation that utilises lower simulation budget).

V. Future Work

For future work, we would like to leverage the availability of multiple base polices in the aftermath of hazards in our framework and incorporate *parallel rollout* in the solution method [31]. We anticipate further improvements to the performance demonstrated here when OCBA is fused with parallel rollout. In the future, we will also present the inter-relationship in other critical infrastructure systems like electrical power, roads, bridges, and water networks and the impact such dynamic interactive system has on the recovery process post-hazard. We are also interested in exploring the social impact of the optimized recovery process. We will examine how to incorporate meta-heuristics to guide the stochastic search that determines most promising actions [32].

References

[1] S. Nozhati, B. R. Ellingwood, H. Mahmoud, and J. W. van de Lindt, "Identifying and Analyzing Interdependent Critical Infrastructure in Post-Earthquake Urban Reconstruction," in *Proc. of the 11th Natl. Conf. in Earthq. Eng.* Los Angel., CA: Earthq. Eng. Res. Inst., Jun 2018.

[2] T. Sun, Q. Zhao, and P. B. Luh, "A Rollout Algorithm for Multichain Markov Decision Processes with Average Cost," in *Posit. Syst.*, R. Bru and S. Romero-Vivó, Eds. Springer Berl. Heidelb., 2009.

[3] L. Péret and F. Garcia, "Online Resolution Techniques," *Markov Decis. Process. in Artif. Intell.*, pp. 153–184, 2010.

[4] "The Association of Bay Area Governments (City of Gilroy Annex)," 2011. [Online]. Available: http://resilience.abag.ca.gov/wp-content/documents/2010LHMP/Gilroy-Annex-2011.pdf

[5] Semseler, R.G. and T. Akel, "2010 Urban Water Management Plan (City of Gilroy)." [Online]. Available: http://www.ci.gilroy.ca.us/265/Water-Management-Plan

[6] N. A. Abrahamson, W. J. Silva, and R. Kamai, *Update of the AS08 ground-motion prediction equations based on the NGA-West2 data set*. Pac. Earthq. Eng. Res. Cent., 2013.

[7] Department of Homeland Security, Emergency Preparedness and Response Directorate, FEMA, Mitigation Division, *Multi-hazard Loss Estimation Methodology, Earthquake Model: HAZUS-MH MR1, Advanced Engineering Building Module*, Wash., DC, Jan 2003. [Online]. Available: https://www.hsdl.org/?view&did=11343

[8] T. Adachi and B. R. Ellingwood, "Serviceability Assessment of a Municipal Water System Under Spatially Correlated Seismic Intensities," *Comput.-Aided Civ. and Infrastruct. Eng.*, vol. 24, no. 4, pp. 237–248, 2009.

[9] J. Eidinger *et al.*, "Seismic fragility formulations for water systems," *Am. Lifelines Alliance, G&E Eng. Syst. Inc.*, 2001. [Online]. Available: http://homepage.mac.com/eidinger

[10] M. Ouyang, L. Dueñas-Osorio, and X. Min, "A three-stage resilience analysis framework for urban infrastructure systems," *Struc. Saf.*, vol. 36, pp. 23–31, 2012.

[11] H. Masoomi and J. W. van de Lindt, "Restoration and functionality assessment of a community subjected to tornado hazard," *Struct. and Infrastruct. Eng.*, vol. 14, no. 3, pp. 275–291, 2018.

[12] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, 1st ed. N. Y., NY, USA: John Wiley & Sons, Inc., 1994.

[13] R. A. Howard, *Dynamic Programming and Markov Processes*. Camb., MA: MIT Press, 1960.

[14] R. Bellman, *Dynamic Programming*, 1st ed. Princet., NJ, USA: Princet. Univ. Press, 1957.

[15] L. S. Shapley, "Stochastic games," *Proc. of the Nat. Acad. of Sci.*, vol. 39, no. 10, pp. 1095–1100, 1953. [Online]. Available: http://www.pnas.org/content/39/10/1095

[16] L. Kallenberg, "Finite state and action MDPs," in *Handb. of Markov Decis.Process.* Springer, 2003, pp. 21–87.

[17] A. Fern, S. Yoon, and R. Givan, "Approximate policy iteration with a policy language bias: Solving relational Markov decision processes," *J. of Artif. Intell. Res.*, vol. 25, pp. 75–118, 2006.

[18] S. Nozhati, Y. Sarkale, B. Ellingwood, E. K. P. Chong, and H. Mahmoud, "Near-optimal planning using approximate dynamic programming to enhance post-hazard community resilience management," *submitted for publication*, vol. abs/1803.01451, 2018. [Online]. Available: https://arxiv.org/abs/1803.01451

[19] S. Nozhati, B. Ellingwood, H. Mahmoud, Y. Sarkale, E. K. P. Chong, and N. Rosenheim, "An approximate dynamic programming approach to community recovery management," in *Eng. Mech. Inst. Conf. (EMI 2018)*, Camb. - Boston, MA, May – Jun 2018.

[20] S. Nozhati, Y. Sarkale, B. R. Ellingwood, E. K. P. Chong, and H. Mahmoud, "A modified approximate dynamic programming algorithm for community-level food security following disasters," in *Proc. 9th Int. Congr. Environ. Model. and Softw. (iEMSs 2018)*, Ft. Collins, CO, Jun 2018.

[21] D. P. Bertsekas and D. A. Castanon, "Rollout algorithms for stochastic scheduling problems," *J. of Heuristics*, vol. 5, no. 1, pp. 89–108, Apr 1999. [Online]. Available: https://doi.org/10.1023/A:1009634810396

[22] M. G. Lagoudakis and R. Parr, "Reinforcement learning as classification: Leveraging modern classifiers," in *Proc. of the 20th Int. Conf. on Mach. Learn. (ICML-03)*, 2003, pp. 424–431.

[23] C. Dimitrakakis and M. G. Lagoudakis, "Algorithms and bounds for rollout sampling approximate policy iteration," in *Recent Adv. in Reinf. Learn.*, S. Girgin, M. Loth, R. Munos, P. Preux, and D. Ryabko, Eds. Berl., Heidelb.: Springer Berl. Heidelb., 2008, pp. 27–40.

[24] M. C. Fu, C. H. Chen, and L. Shi, "Some topics for simulation optimization," in *2008 Winter Simul. Conf.*, Dec 2008, pp. 27–38.

[25] C. Dimitrakakis and M. G. Lagoudakis, "Rollout sampling approximate policy iteration," *Mach. Learn.*, vol. 72, no. 3, pp. 157–171, Sep 2008. [Online]. Available: https://doi.org/10.1007/s10994-008-5069-3

[26] ——, "Algorithms and bounds for rollout sampling approximate policy iteration," *CoRR*, vol. abs/0805.2015, 2008. [Online]. Available: http://arxiv.org/abs/0805.2015

[27] A. Lazaric, M. Ghavamzadeh, and R. Munos, "Analysis of classification-based policy iteration algorithms," *J. of Mach. Learn. Res.*, vol. 17, no. 19, pp. 1–30, 2016. [Online]. Available: http://jmlr.org/papers/v17/10-364.html

[28] J. Branke, S. E. Chick, and C. Schmidt, "Selecting a selection procedure," *Manage. Sci.*, vol. 53, no. 12, pp. 1916–1932, 2007. [Online]. Available: https://doi.org/10.1287/mnsc.1070.0721

[29] C.-H. Chen, J. Lin, E. Yücesan, and S. E. Chick, "Simulation budget allocation for further enhancing the efficiency of ordinal optimization," *Discret. Event Dyn. Syst.*, vol. 10, no. 3, pp. 251–270, Jul 2000. [Online]. Available: https://doi.org/10.1023/A:1008349927281

[30] C.-H. Chen, S. D. Wu, and L. Dai, "Ordinal comparison of heuristic algorithms using stochastic optimization," *IEEE Trans. Robot. Autom.*, vol. 15, no. 1, pp. 44–56, Feb 1999.

[31] H. S. Chang, R. Givan, and E. K. P. Chong, "Parallel rollout for online solution of partially observable markov decision processes," *Discret. Event Dyn. Syst.*, vol. 14, no. 3, pp. 309–341, Jul 2004. [Online]. Available: https://doi.org/10.1023/B:DISC.0000028199.78776.c4

[32] A. Kaveh and N. Soleimani, "CBO and DPSO for optimum design of reinforced concrete cantilever retaining walls," *Asian J. Civ. Eng.*, vol. 16, no. 6, pp. 751–774, 2015.