Simultaneous Optimization And Sampling Of Agent Trajectories Over A Network

Hala Mostafa[1], Akshat Kumar[2], and Hoong Chuin Lau[2]

[1] United Technologies Research Center mostafh@utrc.utc.com
[2] Singapore Management University akshat@smu.edu.sg hclau@smu.edu.sg

Abstract. We study the problem of optimizing the trajectories of agents moving over a network given their preferences over which nodes to visit subject to operational constraints on the network. In our running example, a theme park manager optimizes which attractions to include in a day-pass to maximize the pass's appeal to visitors while keeping operational costs within budget. The first challenge in this combinatorial optimization problem is that it involves quantities (expected visit frequencies of each attraction) that cannot be expressed analytically, for which we use the Sample Average Approximation. The second challenge is that while sampling is typically done prior to optimization, the dependence of our sampling distribution on decision variables couples optimization and sampling. Our main contribution is a mathematical program that *simultaneously* optimizes decision variables and implements inverse transform sampling from the distribution they induce. The third challenge is the limited scalability of the monolithic mathematical program. We present a dual decomposition approach that exploits independence among samples and demonstrate better scalability compared to the monolithic formulation in different settings.

1 Introduction

The diffusion of entities or phenomena over networks, be they social, physical or information networks, has been studied in areas as diverse as disease control [6], targeted advertising [3, 12] and traffic management. Initial efforts were more concerned with building different diffusion models applicable under different assumptions [4, 16, 5, 10, 7, 19, 17]. Armed with a model, the natural next step is optimization in the context of the learned model. The goal is typically to find the set of actions that result in a diffusion process with certain desirable properties subject to operational constraints [9, 7, 15]. Applications include disease control (which nodes to vaccinate to curb the spread of disease) and advertising (which nodes to target to encourage the adoption of a product).

The decision problems addressed in the literature involve actions that either do not change the diffusion process, or change it only by altering the network over which diffusion takes place. For example, in influence maximization the goal is to determine the set of nodes to target (e.g. with free samples of a product) to maximize the spread of a given phenomenon (e.g. purchase of the product)

using a generalization of the Independent Cascade Model [7]. Network design tries to find the set of nodes whose addition to/removal from the network maximizes/minimizes the expected number of cascades that reach/infect a target set of nodes subject to constraints on the cost of adding/removing nodes [15, 11].

We consider a more challenging optimization problem where **actions affect the parameters of the diffusion process**. As a motivating example, we consider the problem of deciding which attractions to include in a theme park day-pass to maximize its appeal while keeping operational costs within budget. Park attractions are modeled as nodes in a network. Park visitors "diffuse" along network edges based on both their inherent preferences and the attractions included in the pass. We model agent diffusion in the network using conditional distributions P_{uv} that specify the probability of visiting attraction v after u is visited. This is a modification of the Independent Cascade model [7] where probabilities of visiting the different neighbors of a given node add up to 1 (unlike the case of a disease spreading to multiple neighbors of a node, for example). Diffusion starts at a start node (e.g., park entrance) and at each time step, each agent samples from the distribution of its current attraction to determine its next attraction. Each trajectory is therefore a random walk through the network.

The first challenge is that our optimization problem is stochastic; it involves quantities (e.g. expected visit frequencies to park attractions) that depend probabilistically on the decision variables and cannot be expressed analytically in closed form. We address this challenge using Sample Average Approximation (SAA) which creates a deterministic version of the problem that approximates expected quantities by evaluating them on a set of samples [8].

The second challenge is that because the decisions of which attractions are in the pass affect diffusion probabilities, we cannot generate samples offline before the optimization step. To address this, our main contribution is a mathematical program that simultaneously optimizes the decision variables and implements inverse transform sampling on-the-fly from the diffusion distributions induced by the decision variables.

Implementing inverse transform sampling introduces a large number of variables which limit the scalability of the monolithic mathematical program. We present a dual decomposition approach that exploits independence among samples and demonstrate better scalability compared to the monolithic formulation in different settings.

2 A MILP For Online Sampling

We present a Mixed Integer Linear Program (MILP) formulation of the combined optimization and sampling problem when *actions affect parameters of the diffusion process*. Specifically, the diffusion of the agents depend on their inherent preferences and the decisions variables. Throughout the section, we will refer to constraints that make up the MILP in Table 1.

We first present the stochastic program formulation and explain how we model the effects of decisions on diffusion. We then use sample average ap-

proximation to determinize the stochastic program. Next, we present our main contribution, a linear formulation of a sampling procedure to plug into the MILP.

2.1 The stochastic program

Although we focus on the theme park problem as an example where the optimization involves expectations taken over a distribution that depends on the decision variables, our formulation is applicable to any diffusion model where flow is conserved. In our setting, a park owner wishes to design a fixed-price pass. Revenue per pass is the price minus the total redemption cost of the attractions actually visited. The pass should include popular attractions to increase sales. However, since the owner pays a redemption cost c_u to the operator of an attraction u when a visitor redeems the ticket for this attraction, the pass includes less popular items with low redemption probability to increase revenue[3].

We use a stochastic program formulation to find the pass with maximum expected "appeal" subject to the expected redemption cost being within a budget B. As a surrogate for appeal, we maximize the expected number of times pass attractions are visited.

$$\max_x \sum_u x_u \mathbb{E}_{\tilde{P}}[I_u] \tag{1}$$

$$s.t \quad \sum_u x_u c_u \mathbb{E}_{\tilde{P}}[I_u] \leq B \tag{2}$$

where x_u is the binary decision variable of whether item u is in the pass and I_u is a random variable denoting whether u is visited/redeemed. I_u follows the diffusion distribution $\tilde{P}$ that depends on x as discussed below.

2.2 Original and modified diffusion models

The original diffusion model: We adopt a variant of the Independent Cascade Model [7] where instead of an activated/visited node activating a subset of its neighbors, it activates only 1 neighbor, since each sequence of activations models the trajectory of 1 visitor.

If there is no pass, the diffusion model is given by P_{uv} specifying the probability of a visitor at node u moving to node v. This model can be learned from instances of trajectories [19] and used for optimization [15]. To generate a sample trajectory, we start at the start node and sample from the distributions P_{uv} until either a designated end node is reached (park exit) or we have generated a trajectory of a given length.

It is important to note that we cannot evaluate the expected number of visits to a given node in the context of the original model P, since P does not reflect the effects of decisions of which attractions are chosen. We therefore need to express

[3] This problem is motivated by the authors' interaction with the management of a large theme park in Singapore.

a modified model $\tilde{P}$ in terms of P and the decision variables. For example, we need to capture the effect of the pass contents on the way visitors move in a park; once a visitor buys a pass, she is more likely to visit attractions included in the pass than those for which she needs to buy additional tickets. However, the visitor's inherent preferences modeled by P will still affect her transition probabilities to some extent.

To highlight the importance of optimizing w.r.t. $\tilde{P}$ rather than P, consider a node v with a high probability of being visited under the original P but is only visited after u. If the cost c_u is too high, the optimal pass may not include u. Optimizing w.r.t. P does not account for the decision of not including u and overestimates the expected number of visits to v. Using $\tilde{P}$ allows us to evaluate a pass in the context of its actual effects on visitor trajectories.

The modified distribution $\tilde{P}$: A simple model for how pass contents affect the trajectory of a visitor is to discount the probability of transitioning to an item not in the pass by a factor of $\alpha \in [0, 1]$ and increase the probability of each item in the pass to maintain a legal transition function. If p_{uv} is the original transition probability from u to v, we define the modified probability $\tilde{p}_{uv}$ as:

$$\tilde{p}_{uv} = p_{uv} + x_v d_u - \alpha(1 - x_v)p_{uv} \tag{3}$$

where d_u is the probability mass added to each neighbor of u that is in the pass. $\tilde{p}_{uv}$ is thus the old probability, to which we add d_u if v is in the pass, or subtract a fraction α if v is not in the pass [4]

Let $\mathcal{N}_u$ be the set of u's neighbors and b_u be the number of neighbors of u that are in the pass. d_u is then given by:

$$d_u = \frac{\alpha \sum_{w \in \mathcal{N}_u} (1 - x_w)p_{uw}}{b_u} \tag{4}$$

The numerator is the total mass removed from neighbors not in the pass. This way of re-distributing probability mass guarantees that $\sum_{w \in \mathcal{N}_u} \tilde{p}_{uw} = 1$.

To linearize the product $x_v * d_u$ in constraint (3), we introduce continuous variables $e_{uv} \in [0, 1]$ to represent the probability mass, if any, to be added to p_{uv}.

$$e_{uv} = x_v * d_u$$

Because x_v is binary, the above can easily be expressed using the linear constraints (11-13) in Table 1.

To linearize constraint (4), we introduce binary variables b_u^m for $m = 1..|\mathcal{N}_u|$ where $b_u^m = 1$ if m of u's neighbors are in the pass. This is enforced by constraint (14). For every value of m, constraints (15) and (16) guarantee that if $b_u^m = 1$, then

$$d_u = \frac{\alpha \sum_{w \in \mathcal{N}_u} (1 - x_w)p_{uw}}{m}$$

[4] According to Eq(3), when $\alpha = 1$, attractions that are not in the pass are removed from a visitor's choice set. We note that our scheme for redistributing probabilities respects the axiom of independence of irrelevant alternatives[14] which says that removing irrelevant alternatives that were not chosen from a set does not change the relative order of other items in the set.

$$\max \frac{1}{N} \sum_i \sum_u S_u^i \quad (5)$$

$$\text{s.t.} \sum_i \sum_u c_u * S_u^i \leq B * N \quad (6)$$

A. Visit frequency constraints $\forall u, i$

$$S_u^i \leq x_u \quad (7)$$

$$S_u^i \leq \sum_t \sum_v I_{vu}^{it} \quad (8)$$

$$S_u^i \geq x_u + \sum_t \sum_v I_{vu}^{it} - 1 \quad (9)$$

B. Prob. modification constraints $\forall u, v$

$$\tilde{p}_{uv} = p_{uv} + e_{uv} - \alpha(1 - x_v)p_{uv} \quad (10)$$

$$e_{uv} \leq x_v \quad (11)$$

$$e_{uv} \leq d_u \quad (12)$$

$$e_{uv} \geq x_v + d_u - 1 \quad (13)$$

$$\sum_{w \in \mathcal{N}_u} x_w = \sum_{m=1}^{|\mathcal{N}_u|} m * b_u^m \quad (14)$$

$\forall u, \forall m = 1..|\mathcal{N}_u|$:

$$d_u \leq 1 + \frac{\alpha \sum_{w \in \mathcal{N}_u}(1 - x_w)p_{uw}}{m} - b_u^m \quad (15)$$

$$d_u \geq \frac{\alpha \sum_{w \in \mathcal{N}_u}(1 - x_w)p_{uw}}{m} + b_u^m - 1 \quad (16)$$

C. ITS constraints $\forall i = 1..N, \quad u, \quad v \in \mathcal{N}_u$

$$g_{uv}^i \geq r_u^i - \sum_{v' \prec v} \tilde{p}_{uv'} \quad (17)$$

$$g_{uv}^i \leq 1 + r_u^i - \sum_{v' \prec v} \tilde{p}_{uv'} \quad (18)$$

D. Edge activiation constraints

$\forall i = 1..N, \quad u, \quad v \in \mathcal{N}_u, \quad t = 1..T$

$$I_{uv}^{it} \leq g_{uv}^i \quad (19)$$

$$I_{uv}^{it} \leq 1 - g_{uv''}^i \quad (20)$$

$$I_{uv}^{it} \leq \sum_{w \neq v} I_{wu}^{i(t-1)} \quad (21)$$

$$I_{uv}^{it} \leq 1 - \sum_{t' < t} \sum_w I_{wv}^{it'} \quad (22)$$

$$I_{uv}^{it} \geq g_{uv}^i + 1 - g_{uv'}^i + \sum_{w \neq v} I_{wu}^{i(t-1)} + 1 - \sum_{t' < t} \sum_w I_{wv_j}^{it'} - 3 \quad (23)$$

$x_u, b_u^m, g_{uv}^i \in \{0, 1\}, I_{uv}^{it} \in \{0, 1\}$ for $t = 1, u = 1$.

Table 1. MILP for optimization and sampling.

Semantics of α: The parameter α models how much p_{uv} depends on the decisions x_u. In our example, α can be a measure of a visitor's unwillingness to purchase additional tickets for items outside the pass. When $\alpha = 0$, $\tilde{p}_{uv} = p_{uv}$ and a visitor's trajectory is independent of the pass (he is willing to purchase additional tickets for attractions not in the pass). At $\alpha = 1$,

$$\tilde{p}_{uv} = x_v \left(p_{uv} + \frac{\sum_{w \in \mathcal{N}_u}(1 - x_w)p_{uw}}{\sum_{w \in \mathcal{N}_u} x_w} \right)$$

So if $x_v = 0$, $\tilde{p}_{uv} = 0$. In this case, the visitor is unwilling to pay extra to visit attractions not in the pass. Defining $\tilde{P}$ in terms of x and α has the desirable property of maintaining the relative ordering among a node's neighbors that are in the pass (the same amount is added to the probability of each).

2.3 Sample Average Approximation

Sample Average Approximation (SAA) [8, 13] is a method for solving stochastic optimization problems by sampling from the underlying distribution to generate a finite number of scenarios and reducing the stochastic optimization problem to a deterministic analogue.

In lieu of evaluating $\mathbb{E}_{\tilde{P}}[I_u]$ analytically, which can be hard or impossible depending on how x affects $\tilde{P}$, we use SAA to approximate the expectation as the average over N sample trajectories $\{\xi^i\}_{i\in 1..N}$ drawn from $\tilde{P}$.

$$\mathbb{E}_{\tilde{P}}[I_u] = \frac{1}{N}\sum_{i=1}^{N}\sum_{v} I^i_{vu} \tag{24}$$

where I^i_{uv} is an indicator variable of whether trajectory ξ^i includes a move from v to u. If $I^i_{vu} = 1$, we say that edge vu is *active* in trajectory ξ^i.

The objective function (1) and budget constraint (2) can now be expressed in terms of edge activation variables and linearized by the introduction of variables S^i_u. If $x_u = 0$, $S^i_u = 0\ \forall i$. If $x_u = 1$, S^i_u indicates whether ξ^i visits u. The definition

$$S^i_u = x_u * \sum_t \sum_v I^{it}_{vu}$$

is linearized in constraints (7)-(9). We explain the need for the time index t below.

2.4 Inverse transform sampling

Our optimization problem has a circularity whereby decisions are evaluated based on samples drawn from $\tilde{P}$ while $\tilde{P}$ itself depends on the decisions. This is different from previous applications of SAA and similar determinization methods (e.g., [15, 19, 18]) where the samples are generated offline prior to solving the optimization problem.

Our main technical contribution is breaking this circularity by formulating linear constraints that implement Inverse Transform Sampling (ITS) [2] and including them in our MILP, thereby allowing simultaneous optimization and sampling. We first explain ITS then show how we use it in our setting.

In the general case, to perform ITS from a distribution $p(X)$, we uniformly generate a number $r \in [0, 1]$. The value of the sample is the largest number x for which $p(-\infty \leq X \leq x) \leq r$. For categorical distributions, where the notion of "largest" number does not hold, an arbitrary order is imposed on the categories which is used to convert the category probabilities into a cumulative distribution function (CDF). For category c, $\mathrm{CDF}(c) = \sum_{c' \prec c} p(c')$, where $\prec$ denotes precedence in the arbitrary order. For a uniformly drawn value r, ITS returns the unique category c for which r is in the interval

$$\left[\sum_{c' \prec c} p(c'),\ \ \sum_{c' \prec c} p(c') + p(c)\right]$$

In other words, r must lie in the probability range of the returned category.

In our setting, a trajectory consists of a linear path of *active edges* starting at the start node. Sampling a trajectory is therefore the process of deciding, for each edge, whether it is active in this trajectory. To apply ITS to sampling

trajectories from our conditional distribution $\tilde{P}$, we uniformly sample a value $r_u^i \in [0,1]$ for every node u and every sample ξ^i. Again, we impose an arbitrary order on the neighbors of u and determine which neighbor's probability range r_u^i falls within. The probability range of a neighbor v of node u is given by:

$$\left[\sum_{v' \prec v} \tilde{p}_{uv'}, \quad \sum_{v' \preceq v} \tilde{p}_{uv'} + \tilde{p}_{uv}\right]$$

Time-indexed variables: Note that if an edge uv is active, then r_u^i falls within the probability range of neighbor v of u. The converse is *not necessarily* true, since we disallow edge activations that result in cyclic trajectories. We disallow cycles by using time-indexed binary activation variables I_{uv}^{it} indicating whether edge uv is traversed at time t in ξ^i.

To show how I_{uw}^{it} depends on r_u^i, consider a node u with neighbors v, w and z and an arbitrary order $\{v, w, z\}$. For node w, we impose constraints that activate edge uw at time t in trajectory ξ^i if 4 conditions hold (i.e., we force I_{uw}^{it} to be the AND of 4 binary conditions): 1) $r_u^i \geq$ the lower limit of w's probability range; 2) $r_u^i <$ the upper limit of w's probability range; 3) u was visited at time $t-1$; and 4) w was not visited at any time $t' < t$.

To enforce the first 2 conditions, we create binary variables g_{uv}^i for each neighbor v of each node u indicating whether r_u^i is greater than the lower limit of v's probability range in sample ξ^i. Constraints (17) and (18) enforce

$$g_{uv}^i = 1 \text{ iff } r_u^i \geq \sum_{v' \prec v} \tilde{p}_{uv'}$$

Constraints (19) and (20) guarantee that r_u^i falls in v's probability range (v'' is the node that directly follows v in the order). Constraint (21) forces I_{uv}^{it} to be 0 if u was not visited at time $t-1$. Constraint (22) forces I_{uv}^{it} to be 0 if v was previously visited. Constraint (23) completes the set of constraints that guarantee that I_{uv}^{it} is the AND (product) of the 4 conditions.

Because by definition of probability ranges, r_u^i will fall within the range of exactly 1 of u's neighbors, it is not possible for 2 edges out of a single node to be activated. And because the constraints disallow cycles, if the uniformly sampled random numbers for a trajectory give rise to a cycle 1-2-3-2, for example, the constraints will only activate edges 1-2 at time 1, and 2-3 at time 2. Some trajectories will therefore be shorter than others.

Extension to non-linear diffusion: We have so far shown how ITS can be used to sample linear trajectories through a network. The same methodology can also be used to generate cascades from an Independent Cascade Model where an active node can infect/activate multiple of its neighbors at multiple points in time. In this case, we need a set of uniformly sampled numbers per time step and activation is defined per node rather than per edge. The probability modification constraints will relax the requirement that $\sum_v p_{uv} = 1$, since activations of different neighbors are independent. We leave a thorough exploration of this extension to future work.

Binary variables: The last line in Table 1 shows the set of binary variables, with all other variables being continuous. It is well known that the difficulty of solving a MILP increases as we increase the number of binary variables. We observe that only the activation variables for edges emanating from the start node u at $t = 1$ need to be specified as binary; the constraints ensure that the rest will only take on 0/1 values.

3 Dual Decomposition Approach

The auxiliary variables introduced by linearization and ITS result in a MILP whose size does not scale well with the number of samples. However, the independence of the sampling process across samples strongly favors a decomposition into a set of subproblems, each responsible for the constraints implementing ITS for a subset of samples. We create local copies of the decision variables x and each subproblem optimizes its local copy. We show how Lagrangian relaxation provides a principled way of exploiting this independence among samples.

3.1 Lagrangian relaxation

For N samples, we create M subproblems each involving N/M samples. For ease of exposition, we assume $M = N$. Subproblem i involves a local copy of the decision variables (denoted x^i) as well as purely local variables $S^i_u, g^i_{uv}, I^{it}_{uv}$. Each subproblem has the form of the MILP in Table 1, but reduced to the sample(s) it is responsible for and augmented with *consistency constraints* $x^i_u = G_u \ \forall u$ which force decision variables of different subproblems to be equal (to a global vector G).

The two constraints that "tie" the subproblems are the budget constraint and the global consistency constraints. We relax both by dualizing them into the objective function with appropriate Lagrange multipliers. The Lagrangian of the the MILP now becomes

$$\mathcal{L} = \frac{1}{N}\sum_{i,u} S^i_u + \sum_{i,u} \mu^i_u (x^i_u - G^i_u) + \gamma(\sum_{i,u} c_u S^i_u - B * N)$$

where μ and γ are Lagrange multipliers of the consistency constraint in subproblem i and the budget constraint, respectively. The dual is

$$q(\mu, \gamma) = \max_{\mathcal{P}} \sum_{i,u} S^i_u + \sum_{i,u} \mu^i_u (x^i_u - G^i_u) + \gamma(\sum_{i,u} c_u S^i_u - B * N)$$

where the maximization is over the set $\mathcal{P}$ of primal variables (x, g, I...etc.). To prevent the dual from being unbounded from below, μ must satisfy

$$\sum_{i=1}^{N} \mu^i_u = 0 \ \ \forall u$$

and γ must be non-negative. Under these conditions, the global variable G disappears from the dual (it is multiplied by 0) to give

$$q(\mu,\gamma) = \max_{\mathcal{P}} \sum_{i,u} S_u^i + \sum_{i,u} \mu_u^i x_u^i + \gamma \sum_{i,u} c_u S_u^i - \gamma B * N$$

The relaxation and dualization of the coupling constraints allows us to decompose the dual into the sum of subproblem duals:

$$\begin{aligned} q(\mu,\gamma) &= \sum_i q^i(\mu^i,\gamma) - \gamma BN \\ q^i(\mu^i,\gamma) &= \max_{\mathcal{P}^i} \sum_u S_u^i + \sum_u \mu_u^i x_u^i + \gamma \sum_u c_u S_u^i \end{aligned} \tag{25}$$

The above decomposition and the separability of the constraints by subproblem allow us to do each minimization independently. The difficulty of solving each subproblem is now independent of the number of the total number of samples. As we show in the experimental results, this addresses the scalability issues of the monolithic MILP.

3.2 Subgradient method

The question now is how to solve the dual optimization problem

$$\min_{\mu,\gamma} q(\mu,\gamma) \qquad \text{s.t.} \sum_{i=1}^{N} \mu_u^i = 0 \;\; \forall u, \quad \gamma \geq 0$$

while obtaining quality bounds on the solution. Duality theory tells us that the value of the dual is an upper bound on the value of the primal, as well as the important fact that the dual is always a convex function. The absence of local optima in the dual allows the use of subgradient descent methods to minimize the dual by iteratively updating the values of the Lagrange multipliers [1]. The update rule for γ at iteration k is given by

$$\gamma^k = \max(0, \gamma^{k-1} - \sigma^k \nabla_\gamma q(\mu^{k-1}, \gamma^{k-1}))$$

where σ^k is the step size at iteration k and

$$\nabla_\gamma q(\mu^{k-1}, \gamma^{k-1}) = \sum_{i,u} c_u \bar{S}_u^{i,k-1} - BN$$

is the subgradient of q wrt γ evaluated at the previous point $(\mu^{k-1}, \gamma^{k-1})$. Each vector $\bar{S}^{i,k-1}$ is the value of S^i in the optimal solution to subproblem i at the previous iteration (this is obtained as part of the solution to the maximization in problem i). The max operator projects γ back to satisfy $\gamma \geq 0$.

For each μ^i, the subgradient is given by

$$\nabla_{\mu^i} q(\mu^{k-1}, \gamma^{k-1}) = \bar{x}^{i,k-1}$$

where vector $\bar{x}^{i,k-1}$ is the value of x^i in the optimal solution to subproblem i at the previous iteration. Projecting μ back to the set $\sum_{i=1}^{N} \mu_u^i = 0 \ \forall u$ involves finding the closest point in this set to the updated μ, which is given by

$$\mu_u^{i,k} = \mu_u^{i,k-1} - \sigma^k(\bar{x}_u^i - \frac{\sum_j \bar{x}_u^j}{N})$$

Intuitively, the updates to the Lagrange multipliers encourage the primal variables to take values that satisfy the relaxed constraints. For example, if the previous iteration's budget exceeds the limit $B * N$, γ will decrease, thus penalizing each q^i into finding a solution with lower cost. Similarly, if a subproblem has a value of x_u^i that is above (resp. below) the average value for this variable across subproblems, the multiplier of this variable will increase (resp.decrease) to promote consistency across subproblems.

Step size: There are a number of recommendations for setting the step size in subgradient method. We use the following rule which has theoretical justifications in [1]:

$$\sigma^k = \lambda \frac{\max_{k'<k} f^{k'} - q(\mu^k, \gamma^k)}{\parallel \nabla q^k \parallel^2}$$

where $\max_{k'<k} f^{k'}$ is the maximum primal value seen so far and is used as an approximation to the optimal dual value. The above rule takes larger steps when the duality gap between the best primal and the current dual is large. The constant multiplier λ is recommended to be in [0,2] and we set it to 0.3. $\parallel \nabla q^k \parallel^2$ is the squared norm of the gradient evaluated in the current iteration.

Primal solutions While the Lagrange multipliers promote consistency across subproblems, in practice converging to complete consistency is often a slow process. After every iteration of the subgradient method, we can use the x^i values to construct a primal solution. A simple scheme used in [11] is to use a majority vote with a consensus threshold θ where for each u, if at least a fraction θ of subproblems agree on the value of x_u, this value is used in the constructed primal. Otherwise, x_u remains a free optimization variable. We then solve the monolithic MILP of Table 1 with the fixed consensus values to obtain a full primal solution. An iteration has no primal solution if: 1) there are too many free variables, so we avoid solving the monolithic MILP; or 2) the fixed variables result in an infeasible MILP if they always violate the budget constraints.

Besides improving scalability, our decomposition offers modeling flexibility where different subproblems can use different values of α. If we know the distribution over the values of α (e.g. ground surveys indicate that 30% of visitors are very willing to pay for additional tickets), we can set a low value for α in 30% of the subproblems and a high value for the rest.

4 Experimental Results

We now compare the performance of the monolithic and decomposed formulations in terms of 1) quality of the best solution found; 2) time the best solution

was first found and 3) the duality gap (difference between the maximum primal and the minimum dual solutions). We also investigate how our formulations perform as we change 1) the number of samples; 2) the parameter α and 3) the tightness of the budget constraint [5].

Instead of 1 subproblem per sample, we group every 2 samples in a subproblem. This increases the size of each subproblem, but is more conducive to convergence and finding good intermediate primal solutions. We solve the subproblems in parallel. In solving the monolithic MILP, we allow CPLEX to use all available cores.

We use 5 randomly generated instances of a network with 24 nodes. The instances differ in the redemption costs of nodes, the distribution P, and the connectivity (each node is connected to half of the remaining nodes at random). For every instance, we generate the uniformly sampled numbers $r_u^{i=1..N}$ for samples sizes $N \in \{10, 16, 22, 36\}$. We set the maximum trajectory length to 5, so for N samples, the upper bound on the objective function is $N * (5 - 1)$, since the start node does not count towards the objective. Note that because some trajectories are cut short to avoid cycles, this value is not always attainable.

To vary the tightness of the budget, we calculate the average redemption cost per trajectory over 1000 trajectories drawn from the original distribution P. We then set the budget to either 75% or 100% of the calculated average to experiment with tight and relaxed budget settings.

Table 2 compares the performance of the two formulations. Boldface entries indicate the decomposed formulation doing as well as or better than the monolithic formulation. Times are in seconds and the final duality gap is calculated as $\frac{min(q)-max(f)}{max(f)} * 100\%$ where q and f are the dual and primal solution values. All runs were terminated after 3600 seconds except for experiments with 36 samples and $\alpha = 0.3$ at a budget 75% of the average (shown in the starred row in Table 2) which was run for 5000 seconds.

At $\boldsymbol{\alpha = 0}$ As explained above, α is a parameter controlling how much the distribution $\tilde{P}$ depends on the decision variables. When $\alpha = 0$, $\tilde{P} = P$ and the size of the MILP is greatly reduced because trajectory sampling can be done offline, eliminating the constraints and variables implementing the inverse transform sampling. Because it is very easy to obtain, we use the solution from setting $\alpha = 0$ as the initial solution (denoted $x_{\alpha=0}$) when solving for larger values of α in both the monolithic and decomposed formulations. This approach can be helpful in finding good solutions early on. However, as the value of α increases, $x_{\alpha=0}$ becomes less valid and indeed may be infeasible. For example, trajectories from the distribution $\tilde{P}$ induced by $x = x_{\alpha=0}$ and $\alpha = 0.9$ have a high redemption cost because $x_{\alpha=0}$ was obtained assuming no increase in the probability of visiting an attraction in the pass, which is not true at $\alpha = 0.9$. "Deploying" the solution $x_{\alpha=0}$ when $\alpha = 0.9$ will therefore typically violate the

[5] All MILPs solved using IBM CPLEX 12.6 on a 16-core 2.6GHz machine under a quota of 200G RAM and 24 threads.

Table 2. Comparison of monolithic and decomposed formulations.

	Monolithic			Decomposed			Result on 3000 samples	
N	Best sol.	1^{st} time	Final gap	Best sol.	1^{st} time	Final gap	% over estimate	% budget violation
α=0.3 Budget = 75% of avg.								
10	34	566	0	33.4	**233**	13.4	30.8%	13.6%
16	48	753	25.2	**50.2**	1560	**14**	12.1%	19.4%
22	66.8	361	31.8	66.6	1588	**24.1**	8.44%	15.3%
36$\star$	106	540	36.3	104	2012	**30.7**	8.85%	14.3%
α=0.9 Budget = 75% of avg.								
10	38.4	1721	2.7	37.4	**354**	**2.7**	10.0%	12.9%
16	45.6	603	43.7	**60**	608	**3**	6.23%	25.4%
22	63.2	536	42.2	**80.6**	1519	**5.5**	4.98%	19%
36	N/A	N/A	N/A	**129**	**1731**	**11**	4.78%	6.4%
α=0.3 Budget = 100% of avg.								
10	37	806	3.3	36.2	**110**	5.7	23.3%	12.4%
16	53.2	62.7	20.4	**55.4**	2115	**6.1**	15.4%	13.5%
22	72.2	100	22	**72.6**	1435	**16.7**	6.49%	10.8%
36	120	113	20	115.8	958	20.5	7.53%	10.2%
α=0.9 Budget = 100% of avg.								
10	39.4	890	0.5	39	**119**	1.54	11.19%	8.44%
16	56.2	1893	15	**60.6**	**127**	**2.33**	5.00%	13.4%
22	78	287	12.9	**83.2**	336	**2.6**	5.05%	13.5%
36	129	603.6	12.1	**136.2**	**556**	**3.6**	5.98%	8%

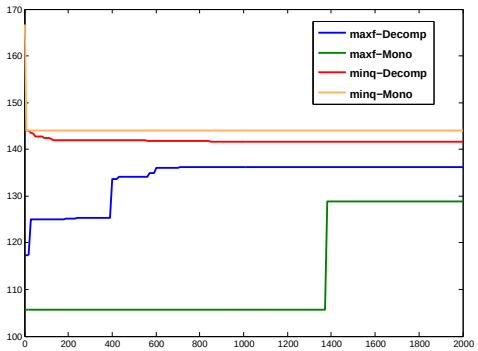

Fig. 1. Average maximum primal and minimum dual values of decomposed and monolithic formulations over time (in seconds). $\alpha = 0.9$, $N = 36$ and relaxed budget.

budget constraint because at $\alpha = 0.9$ visitors flock to pass attractions much more than anticipated.

At larger α, we notice that the pre-processing done by CPLEX is less effective in pruning the monolithic MILP, which makes runs with $\alpha = 0.9$ the most challenging for the monolithic formulation. Combined with a tight budget that makes finding a feasible solution difficult, the setting $\alpha = 0.9$ with $N = 36$ resulted in only 1 instance that could be solved using the monolithic formulation (corresponding average reported as N/A in Table 2).

However, instances with $\alpha = 0.9$ should be the least challenging, since under this setting most solution will achieve a high objective value. In the theme park example, this situation is like having visitors that are completely "malleable", in which case any day-pass will be very popular, achieving an objective value close to the upper bound. Because each subproblem is responsible for only 2 samples, the decomposed formulation is far less affected by CPLEX's inability to prune the MILP during pre-processing, and can find good solutions early on and terminate with a low duality gap. This is clearly reflected in Table 2 where at a high α and relaxed budget, the decomposed formulation scales well with the number of samples.

Figure 1 shows an example of how the value of the best primal solution and the minimum dual value evolve over time in both formulations. Note that in the monolithic formulation, CPLEX is unable to tighten the upper bound on the objective function beyond the theoretical upper bound of $N(T-1)$ which for $N = 36$ and maximum trajectory length of 4 is 144. The value of the dual function in our decomposed formulation, on the other hand, falls below that bound. The sharp increase in the best primal value of the monolithic formulation is due to the fact that in some instances, CPLEX was unable to find any feasible solution other than the trivially feasible solution with all $x_u = 0$ with objective value 0. When the first non-trivially feasible solution is found, the average primal value rises sharply.

Effect of sample size: Intuitively, the accuracy of the SAA approximation should improve as we use more samples, i.e., the approximate value of a solution should approach the solution's true value. Note that this is different from stating that the value of the optimal solution x^*_N of the determinized version of the stochastic problem will necessarily *improve* as N increases. Rather, the *exact* value $v(x)$ of a solution x will be better approximated by the value $\hat{v}_N(x)$ obtained using N samples as N increases.

In our setting, we do not have access to the exact value $v(x)$ of a solution x; we cannot evaluate x on the set of all trajectories. As a proxy for $v(x)$, we evaluate x on 3000 test trajectories sampled from $\tilde{P}_x$, giving a (better) approximation $\hat{v}_{3000}(x)$. We then compare $\hat{v}_{3000}(x)$ to $\hat{v}_N(x)$ (the optimal value of the objective function in Eq (5) using N samples) for different values of N. Naturally, $\hat{v}_N(x)$ will tend to be an over-estimate of $\hat{v}_{3000}(x)$. The column "% over estimate" in Table 2 shows the average percentage of this over-estimation for different N. As expected, approximate values obtained using smaller sample sizes tend to be

overly optimistic regarding the quality of the solution when applied to the test data. Another notable trend is that this over-estimation is more pronounced at the smaller value of α. The reason is that for larger α, it is not too difficult to score well on the test data, since the modified $\tilde{P}$ will be such that there is a very large probability of visiting items in the pass.

Respecting the budget constraint: In addition to performance in terms of visit frequencies of items in the pass, we also calculate how well a solution respects the budget constraint. Any feasible solution for either of our formulations will have an average redemption cost within the specified budget B. However, the cost of an *individual* sample trajectory may be more than B. The last column in Table 2 shows the average percentage budget violation *per sample* when the optimal solution is tested on 3000 sample trajectories. Increasing the number of samples used in the SAA approximation produces solutions that result in lower budget violations when applied to the 3000 test trajectories.

5 Conclusion

In this paper we addressed decision making in settings where movement of agents in a network is affected by both the agents inherent preferences and the actions of the network manager. Using such a notion of simultaneous decision making and trajectory sampling, we addressed the theme park pass design problem in which the objective is to include popular attractions that increase sales while respecting the redemption cost budget. Our key contributions lie in solving the underlying stochastic program that describes the combinatorial optimization problem using sample average approximation. We implement the logic for inverse transform sampling in the language of mixed integer programming, a formulation which can be leveraged in other optimization settings that need to sample from a distribution online. Our mathematical program simultaneously optimizes the decision variable and generates samples on-the-fly from modified probability distribution. To improve scalability, we developed a Lagrangian relaxation-based decomposition that exploits independence among samples. Experimental results comparing the monolithic and decomposed MILP formulations in different settings show better scalability of the latter and more accurate decision evaluation as sample size increases.

References

1. D. P. Bertsekas. *Nonlinear Programming.* Athena Scientific, 2nd edition, 1999.
2. L. Devroye. *Non-uniform random variate generation.* Springer Verlag, 1986.
3. P. Domingos and M. Richardson. Mining the network value of customers. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 57–66, 2001.
4. J. Du, A. Kumar, and P. Varakantham. On understanding diffusion dynamics of patrons at a theme park. In *AAMAS*, 2014.

5. M. Gomez Rodriguez, J. Leskovec, and A. Krause. Inferring networks of diffusion and influence. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1019–1028, 2010.
6. M. E. Halloran, I. Longini, and C. Struchiner. Binomial and stochastic transmission models. In *Design and Analysis of Vaccine Studies*, Statistics for Biology and Health, pages 63–84. Springer New York, 2010.
7. D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. In *ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 137–146, 2003.
8. A. J. Kleywegt, A. Shapiro, and T. Homem-de-Mello. The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization*, 12:479–502, 2002.
9. A. Krause, J. Leskovec, C. Guestrin, J. VanBriesen, and C. Faloutsos. Efficient sensor placement optimization for securing large water distribution networks. *Journal of Water Resources Planning and Management*, 134(6):516–526, 2008.
10. A. Kumar, D. Sheldon, and B. Srivastava. Collective diffusion over networks: Models and inference. In *International Conference on Uncertainty in Artificial Intelligence*, 2013.
11. A. Kumar, X. Wu, and S. Zilberstein. Lagrangian relaxation techniques for scalable spatial conservation planning. In *AAAI Conference on Artificial Intelligence*, pages 309–315, 2012.
12. J. Leskovec, L. A. Adamic, and B. A. Huberman. The dynamics of viral marketing. *ACM Trans. Web*, 1(1), May 2007.
13. B. K. Pagnoncelli, S. Ahmed, and A. Shapiro. Sample Average Approximation Method for Chance Constrained Programming: Theory and Applications. *Journal of Optimization Theory and Applications*, 142(2):399–416, 2009.
14. P. Ray. Independence of irrelevant alternatives. *Econometrica*, 41(5):pp. 987–991, 1973.
15. D. Sheldon, B. Dilkina, A. Elmachtoub, R. Finseth, A. Sabharwal, and J. Conrad et al. Maximizing the spread of cascades using network design. In *Conference on Uncertainty in Artificial Intelligence*, pages 517–526, 2010.
16. D. Sheldon, T. Sun, A. Kumar, and T. Dietterich. Approximate inference in collective graphical models. In *International Conference on Machine Learning*, 2013.
17. D. R. Sheldon and T. G. Dietterich. Collective graphical models. In *Advances in Neural Information Processing Systems*, pages 1161–1169, 2011.
18. P. Varakantham and A. Kumar. Optimization approaches for solving chance constrained stochastic orienteering problems. In *Algorithmic Decision Theory*, volume 8176 of *Lecture Notes in Computer Science*, pages 387–398. Springer, 2013.
19. X. Wu, A. Kumar, D. Sheldon, and S. Zilberstein. Parameter learning for latent network diffusion. In *International Joint Conference on Artificial Intelligence*, pages 2923–2930, Beijing, China, 2013.