# Optimization Approaches for Solving Chance Constrained Stochastic Orienteering Problems

Pradeep Varakantham and Akshat Kumar*

School of Information Systems, Singapore Management University
* IBM Research India

**Abstract.** Orienteering problems (OPs) are typically used to model routing and trip planning problems. OP is a variant of the well known traveling salesman problem where the goal is to compute the highest reward path that includes a subset of nodes and has an overall travel time less than the specified deadline. Stochastic orienteering problems (SOPs) extend OPs to account for uncertain travel times and are significantly harder to solve than deterministic OPs. In this paper, we contribute a scalable mixed integer LP formulation for solving *risk aware* SOPs, which is a principled approximation of the underlying stochastic optimization problem. Empirically, our approach provides significantly better solution quality than the previous best approach over a range of synthetic benchmarks and on a real-world theme park trip planning problem.

## 1 Introduction

Motivated by competitive orienteering sports, Orienteering Problems (OPs) [15] represent the problem of path selection, where the reward accumulated by visiting a subset of nodes in the path is the maximum for the condition that overall travel time to traverse the path does not violate the deadline. While OPs have been used to represent problems like vehicle routing [7] and production scheduling [2], in this work, we are motivated by the problem of tourist trip design problems similar to the one described in Vansteenwegen *et al.* [16, 1]. Specifically, we address the problem of providing risk sensitive route guidance to visitors at theme parks [10], where the presence of queues at nodes lead to stochastic travel time between nodes.

As OP assumes deterministic edge lengths, they are insufficient to represent the route guidance problem at theme parks. Thus, researchers have extended OPs to stochastic OPs (SOPs), where edge lengths are now random variables that follow a given distribution. The goal is to find a sequence that maximizes the sum of utilities from vertices in the sequence [4]. In this paper, we consider the risk aware SOP [10], where the goal is to compute a path that maximizes the overall reward while enforcing a risk aware deadline constraint. That is, we compute paths where the probability of violating the deadline is less than a given risk parameter, $\alpha$.

Lau *et al.* [10] introduced a local search approach for solving such risk aware SOPs. While such an approach is scalable, it is adhoc and does not provide any a priori or posteriori guarantees with respect to optimal solution. To address these limitations, we provide a principled optimization based approach that employs ideas from the sample average approximation technique to solve stochastic optimization problems [12].

In order to illustrate the utility of our approach, we provide comparisons with the local search approach on a synthetic benchmark set introduced in the literature [4] and also on a real theme park navigation problem, where the travel times are computed from a year-long data set of travel times at a popular theme park in Singapore. The results are quite encouraging—our approach provides significant and consistent increase in solution quality (more than 50% for some synthetic benchmarks and more than 100% for real-world problems) when compared against the local search approach of [10].

## 2 Background: OPs and SOPs with Chance Constraints

The *orienteering problem* (OP) [15] is defined by a tuple $\langle V, E, T, R, v_1, v_n, H \rangle$, where $V$ and $E$ denote the vertices and edges respectively of the underlying graph. $T : v_i \times v_j \rightarrow \mathbb{R}^+ \cup \{0, \infty\}$ specifies a finite non-negative travel time between vertices $v_i$ and $v_j$ if $e_{ij} \in E$ and $\infty$ otherwise; and $R : v_i \rightarrow \mathbb{R}^+ \cup \{0\}$ specifies a finite non-negative reward for each vertex $v_i \in V$. A solution to an OP is a Hamiltonian path over a subset of vertices including the start vertex $v_1$ and the end vertex $v_n$ such that the total travel time is no larger than $H$. Solving OPs optimally means finding a solution that maximizes the sum of rewards of vertices in its path. Researchers have shown that solving OPs optimally is NP-hard [7]. In this paper, we assume that the end vertex can be any arbitrary vertex. The start and end vertices in OPs are typically distinct vertices.

Researchers have proposed several exact branch-and-bound methods to solve OPs [9] including optimizations with cutting plane methods [11, 6]. However, since OPs are NP-hard, exact algorithms often suffer from scalability issues. Thus, constant-factor approximation algorithms [3] are necessary for scalability. Researchers also proposed a wide variety of heuristics to address this issue including sampling-based algorithms [15], local search algorithms [7, 5], neural network-based algorithms [17] and genetic algorithms [14]. More recently, Schilde et al. developed an ant colony optimization algorithm to solve a bi-objective variant of OPs [13].

The assumption of deterministic travel times is not a valid one in many real-world settings and thus researchers have extended OPs to *Stochastic OPs* (SOPs) [4], where travel times become random variables that follow a given distribution. The goal is to find a path that maximizes the sum of expected utilities from vertices in the path. The random variables are assumed to be independent of each other.

Existing research has focussed on two different objectives in obtaining solutions for a SOP. The first objective by Campbell *et al.* [4] is to maximize sum of expected utilities of visited nodes. The expected utility of a vertex is the difference between the expected reward and expected penalty of the vertex. The expected reward (or penalty) of a node is the reward (or penalty) of the vertex times the probability that the travel time along the path thus far is no larger (or larger) than $H$. More formally, the expected utility $U(v_i)$ of a vertex $v_i$ is

$$U(v_i) = P(a_i \leq H)\, R(v_i) - P(a_i > H)\, C(v_i)$$

where the random variable $a_i$ is the arrival time at vertex $v_i$ (that is, the travel time from $v_1$ to $v_i$), $R(v_i)$ is the reward of arriving at vertex $v_i$ before or at $H$ and $C(v_i)$ is the penalty of arriving at vertex $v_i$ after $H$. Campbell et al. have extended OP algorithms to solve SOPs including an exact branch-and-bound method and a local search method based on variable neighborhood search [4]. Gupta et al. introduced a constant-factor approximation algorithm for a special case of SOPs, where there is no penalty for arriving at a vertex after $H$ [8].

The approach by [4] suffers from many limitations. Firstly, it is a point estimate solution which does not consider the "risk" attitude with respect to violating the deadline. By "risk", we refer to probability of completing the path within the deadline. In other words, a risk-seeking user will be prepared to choose a sequence of nodes that have a large utility, but with a higher probability of not completing the path within the deadline, compared to a risk-averse user who might choose a more "relaxed" path with lower utility. Secondly, the underlying measurement of expected utility is not intuitive in the sense that a utility value accrued at each node does not usually depend on the probability that the user arrives at the node by a certain time; but rather, the utility is accrued when the node is visited.

Given the above consideration, Lau *et al.* [10] proposed a second objective where we maximize accumulated reward while satisfying a chance constraint to account for the risk of exceeding the horizon. This allows the user to tradeoff risk against total utility. More precisely, given a value $0 \leq \alpha \leq 1$, we are interested in obtaining a path, where the probability of failing to complete the entire path within a deadline $H$ is less than $\alpha$. Formally,

$$prob(a_n > H) \leq \alpha \tag{1}$$

where $a_n$ is the arrival time at the last vertex of the path.

## 3 Deterministic Approximation for Chance Constrained Optimization

In this section, we provide a brief overview of the *sample average approximation* (SAA) technique for solving stochastic optimization problems [12]. The stochastic orienteering problem is an instance of the *stochastic optimization* problem, where the risk sensitive behavior is often encoded in the form of chance constraints. An example of such an optimization problem is given below:

$$\min_{x \in X} \{ g(x) := \mathbb{E}_P \big[ G(x, W) \big] \} \tag{2}$$

$$\text{s.t. } \operatorname{prob}\big\{ F(x, W) \leq 0 \big\} \geq 1 - \alpha \tag{3}$$

where $X$ is the feasible parameter space, $W$ is a random vector with probability distribution $P$ and $\alpha \in (0,1)$. The above stochastic optimization problem is called a chance constrained problem [12]. Notice that the objective function is an expectation due to the unobserved random variable $W$. Similarly, the constraint function $F(\cdot)$ is also a random variable due to its dependence on $W$. The parameter $\alpha$ can be interpreted as the parameter to tune the risk seeking or risk averse behavior.

It may seem that such an optimization problem is too unwieldy to solve. Fortunately, a number of techniques do exist that transform such stochastic optimization problem into a deterministic problem in a principled manner. One such technique is called *sample average approximation* [12] or *SAA*. We describe a brief outline below; further details can be found in [12]. Interestingly, the SAA technique can also provide stochastic bounds on the solution quality and thus, provides a principled approximation.

The main idea behind the SAA is to generate a number of samples for the random vector $W$. Let us denote these samples as $W^i$. First, we define the following indicator-like function that returns 1 if the argument is positive and 0 otherwise.

$$\mathbb{I}(t) = \begin{cases} 1 & if\ t > 0 \\ 0 & if\ t \leq 0 \end{cases} \tag{4}$$

We generate $N$ samples for the random variable $W$. Based on these samples, we define the approximate probability of constraint violation for a particular point $x$ as follows:

$$\hat{p}_N(x) = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}\big(F(x, W^i)\big) \tag{5}$$

Now the stochastic optimization is reformulated (approximately) as the following deterministic optimization problem:

$$\min_{x \in X} \frac{1}{N} \sum_{i=1}^{N} G(x, W^i) \tag{6}$$

$$\text{s.t. } \hat{p}_N(x) \leq \alpha' \tag{7}$$

The parameter $\alpha'$ plays the role of $\alpha$ in the above optimization problem. Typically, we set $\alpha' < \alpha$ to get a feasible solution. Often, the above optimization problem can be formulated as a mixed-integer program and thus, can be solved using CPLEX. Based on the number of samples and the parameter $\alpha'$, several bounds for the solution quality and feasibility can be derived [12].

## 4 Solving SOP With Chance Constraints

In this section, we first formulate the SOP problem with chance constraints (see Section 2) as an optimization problem. We then employ the SAA scheme to get a

$$\max_{\boldsymbol{\pi}} \sum_{i,j} \pi_{ij} R_i \tag{8}$$

s.t.

$$\pi_{i,j} \in \{0,1\}, \qquad \forall v_i, v_j \in V \tag{9}$$

$$\sum_j \pi_{ji} \leq 1, \qquad \forall v_i \in V \tag{10}$$

$$\sum_j \pi_{ij} \leq 1, \qquad \forall v_i \in V \tag{11}$$

$$\sum_j \pi_{1j} = 1; \sum_j \pi_{jn} = 1 \tag{12}$$

$$\sum_j \pi_{ij} - \sum_j \pi_{ji} = \begin{cases} 1 & \text{if } i = 1; \\ -1 & \text{if } i = n; \\ 0 & \text{otherwise;} \end{cases}, \qquad \forall v_i \in V \tag{13}$$

$$r_i \leq r_j - 1 + (1 - \pi_{ij}) * M \qquad \forall v_i, v_j \in V \tag{14}$$

$$r_1 = 1,\ r_n = n, r_i \in [1, n] \qquad \forall v_i \in V \tag{15}$$

$$\Pr\big(\sum_{i,j} \pi_{ij} T_{ij} \geq H\big) \leq \alpha \tag{16}$$

Table 1: Formulation of chance constrained SOP as a mathematical program.

deterministic approximation. For each directed edge $(v_i, v_j)$, the binary variable $\pi_{ij}$ denotes whether the edge $(v_i, v_j)$ is in the final path. The random variable $T_{ij}$ denotes the travel time for traversing the directed edge $(v_i, v_j)$. We assume that the underlying distribution for each variable $T_{ij}$ is provided as input. The parameter $R_i$ represents the reward obtained on visiting the node $v_i$.

Table 1 shows the mathematical program for chance constrained SOPs. We next describe its structure. We designate the start node with id 1 and the destination node with $n$. The objective function seeks to maximize the overall reward obtained based on nodes visited. Constraints (10)-(11) specify that there is a single incoming and outgoing active edge for each node. Constraint (13) denotes the flow conservation.

To ensure that there are no cycles in the path, we introduce a new set of variables $r_i$ for each node $v_i$ to denote its rank in the final path. For instance, if the rank of the source node is 1, then any node connected immediately from source will be ranked greater than 1 and so on. Such monotonically increasing ranking of nodes will enforce that no cycles are generated. The constraint (14) models this ranking scheme. The parameter $M$ is a large constant used to maintain the consistency of the constraint.

Constraint (16) denotes the chance constraint. The total duration of the SOP is denoted as $\sum_{i,j} \pi_{ij} T_{ij}$, which is a random variable as each $T_{ij}$ is a random variable. The parameter $H$ denotes the input deadline. The chance constraint states that the probability of violating the deadline should be no greater than $\alpha \in (0, 1)$, another input parameter. This constraint is not linear and in general, a closed form expression is not readily available. We next show how to determinize this constraint using SAA in the MIP framework.

For each edge of the graph, we generate $Q$ samples for the duration random variable $T_{ij}$, denoted by $t^q_{ij}$. We represent the function $\mathbb{I}(\cdot)$ of Eq. (4) using the following linear constraints:

$$z^q \geq \frac{\sum_{ij} \pi_{ij} t^q_{ij} - H}{H} \qquad \forall q \in Q \tag{17}$$

$$z^q \in \{0, 1\} \qquad \forall q \in Q \tag{18}$$

where we have introduced auxiliary integer variables $z^q$ for each sample $q$. Using these auxiliary variables, the constraint (7) is represented as:

$$\frac{\sum_q z^q}{Q} \leq \alpha' \tag{19}$$

where $\alpha'$ is a parameter that is set by the user and is generally smaller than the parameter $\alpha$ as used in constraint (16). The setting of $\alpha'$ is critical and we will provide a detailed discussion about the same in our experimental results section. To summarize, we get a deterministic mixed-integer program corresponding to the stochastic program of Table 1 by replacing the stochastic constraint (16) using $Q$ samples for each random variable corresponding an edge, introducing auxiliary integer variables $z^q$ for each SAA sample and using linear constraints (17), (18) and (19). The following theoretical results establish the convergence guarantees for the SAA technique:

**Theorem 1 ([12]).** *Let $v^\star$ be the optimal solution quality, $\hat{v}_N$ be the quality of the SAA problem, $x^\star$ be the optimal solution, $\hat{x}_N$ be the SAA solution and the parameter $\alpha' = \alpha$, then $\hat{v}_N \to v^\star$ and $\hat{x}_N \to x^\star$ as $N \to \infty$.*

The next theorem provides convergence results regarding the feasibility of the solution w.r.t. the chance constraint.

**Theorem 2 ([12]).** *If $\hat{x}_N$ be the feasible solution of the SAA problem and $\alpha' < \alpha$, then the probability that $\hat{x}_N$ is a feasible solution of the true problem approaches one exponentially fast with the increasing number of samples $N$.*

## 5 Experimental Results

To illustrate the effectiveness of our approach, referred to as MILP-SAA, for solving the SOP, we provide experimental results on a synthetic benchmark set employed in the literature [4] and a real world theme park decision support problem introduced by Lau *et al.* [10]. We measure the performance of our approach with respect to the solution quality and the probability of constraint violation by varying problem parameters. Our results are quite encouraging—the MILP-SAA approach provides significant increase in solution quality (more than 50% for some synthetic benchmarks and more than 100% for real-world problems) when compared against the local search approach of [10], all the while keeping the probability of constraint violation within the specified limit ($= \alpha$).

## 5.1 Synthetic Benchmark Set

Firstly, we provide the comparison on the benchmark set introduced by Lau *et al.* [10]. In this set, we vary the following key problem parameters:

- The graph structures are taken from existing work [4] and the number of nodes ($|V|$) in these graphs vary in the following range: $\langle 20, 32, 63\rangle$. The reward obtained by visiting a node is chosen randomly between 1 and 10.
- The probability of constraint violation or the $\alpha$ parameter of Eq. 16 is varied as: $\langle 0.3, 0.25, 0.2, 0.15, 0.11\rangle$. Corresponding to each setting of $\alpha$, we use the parameter $\alpha'$ ($\leq \alpha$, see Eq. (19)) from the values $\langle 0.2, 0.15, 0.1, 0.05, 0.01\rangle$.
- As in the previous work [10], we employ a gamma distribution, $f(x; k, \theta)$, for modeling the travel time of an edge or the random variable $T_{ij}$.

$$f(x; k, \theta) = \frac{1}{\theta^k} \frac{1}{\Gamma(k)} x^{k-1} e^{\frac{x}{\theta}}, \;\; x > 0, \; k, \theta > 0$$

  The $k$ parameter is randomly selected for different edges, the theta ($\theta$) parameter is varied as: $\langle 1, 2, 3\rangle$.
- Finally, we also test for a range of the deadlines $H$. For each instance, we calculate approximately the total time required to visit all the nodes and then set the deadline $H$ to be the following fraction of the total time: $\langle 20\%, 25\%, 30\%, 35\%\rangle$.

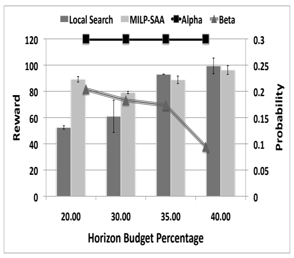
We do not modify the parameters of the local search algorithm provided in Lau *et al.* [10], as it was shown to work across a wide variety of problems. With the MILP-SAA, we compute a 90% optimal solution to ensure easy scalability to larger problems. Also, to understand the performance of our approach better, we employ the following settings for the algorithm:

- The number of samples ($Q$) used by MILP-SAA is varied as: $\langle 25, 30, 35, 40\rangle$
- The number of sample sets generated for each problem is 15. This corresponds to the initial random seeds used to sample the travel time from the gamma distribution.

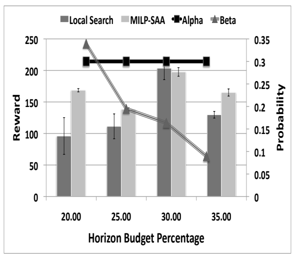
While we obtained results for all the combination of parameters, we only show a representative set of results due to space constraints. We show results where one parameter is modified while keeping other parameters set to their default value. The default values for different parameters are as:

$$\theta = 1; \alpha = 0.3; \alpha' = 0.1; H = 25\%; Q = 40; \tag{20}$$

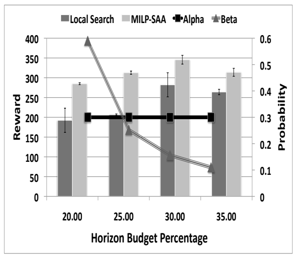
The local search approach always provides a solution with the specified limit $\alpha$. For the MILP-SAA, we empirically determine the actual probability of constraint violation for a particular solution $\boldsymbol{\pi}$, say $\beta$, by generating 1000 complete samples for edge duration and computing the fraction of samples for which the solution violated the deadline $H$. Ideally, the probability $\beta$ should be less than $\alpha$ for the solution to be valid, which is indeed the case for most problem instances for MILP-SAA.



(a) $|V|$=20



(b) $|V|$=32



(c) $|V|$=63

Fig. 1: Effect on reward as the horizon budget is varied

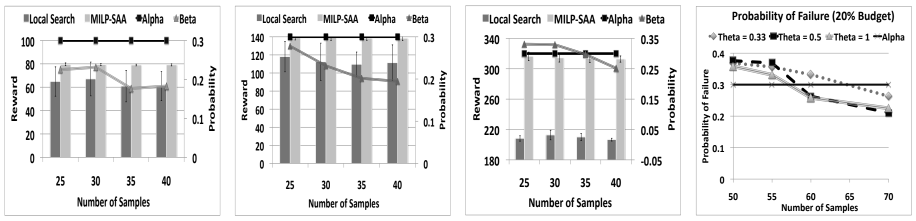
**Runtime:** In this paper, we do not provide detailed results on run-time[1] because both approaches were able to solve all the problems within less than 10 minutes. Local search was able to obtain solutions on the most difficult of problems within a few seconds. On the other hand, we were able to compute solutions by using MILP-SAA approach within 10 minutes on the most difficult problem (63 nodes, $H = 20\%, Q = 70, \alpha' = 0.01, \theta = 3$). When number of samples is less than or equal to 40, we obtain solutions within 2 minutes. There exist expected patterns, such as run-time increasing with decreasing horizon budget and increasing number of samples, however, due to space constraints, we would not be going through those in this paper.

**Horizon Budget:** Figure 1 shows the effect of varying horizon on the overall reward for the three graph configurations. The $X$-axis shows the horizon as the percentage of total time required to visit all the nodes. A 20% horizon budget indicates that on an average, only about 20% of all the nodes can be traversed. The primary $Y$-axis (left side) indicates the reward obtained and the secondary $Y$-axis (right side) indicates the probability of constraint violation. The bars indicate the reward obtained by local search and MILP-SAA. In addition, the two lines represent the probability of constraint violation. The legend 'Alpha' denotes the $\alpha$ parameter and 'Beta' denotes the empirically computed probability of constraint violation for the MILP-SAA solution using 1000 samples. We make the following observations:

- MILP-SAA outperforms the local search in terms of reward consistently and significantly for several cases. In addition, this difference in performance is significant and consistent in the 63 node case. For instance, for the 25% horizon budget case for 63 nodes in Figure 1(c), the reward difference is close to 100, indicating about 50% improvement over the local search. This also implies the traversal of an additional 10 nodes[2] in the worst case and 20 nodes in the average case.

[1] We conduct our experiments on an Intel Core i5 machine with 1.8 GHz processor speed and 8 GB RAM.

[2] Reward for nodes is drawn from a uniform distribution with minimum value of 1 and maximum value of 10.



(a) $|V| = 20$ (b) $|V| = 32$ (c) $|V| = 63$ (d) Higher Samples

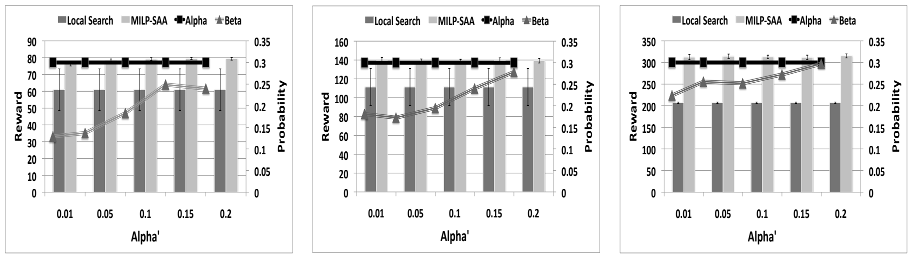
Fig. 2: Effect on reward as the number of samples is varied

- In most of the cases, the variance in performance of local search is much higher than the variance of MILP-SAA. This is an important observation, especially for the few cases where local search dominates MILP-SAA. Thus, MILP-SAA was highly consistent in providing good solution quality.
- As the horizon budget is increased, the problem becomes less constrained and the difference in the reward values between the two approaches reduces. This is as expected.
- As the horizon budget is decreased, the problem is more constrained and hence the actual probability of constraint violation ($\beta$) increases. Specifically, 20% horizon budget is a difficult problem to solve for the 32 and 63 node problems when MILP-SAA employs 40 samples only. This is reflected in the $\beta$ values, which are greater than the $\alpha = 0.3$ threshold. As we show later in this section, this can be addressed by increasing the number of samples ($> 40$) or reducing the $\alpha'$ value employed ($< 0.1$).

**Number of Samples:** We now show the effect of increasing the number of SAA samples on reward in Figure 2, while setting all the other parameters to their default value as in Eq. (20). We make the following key observations:

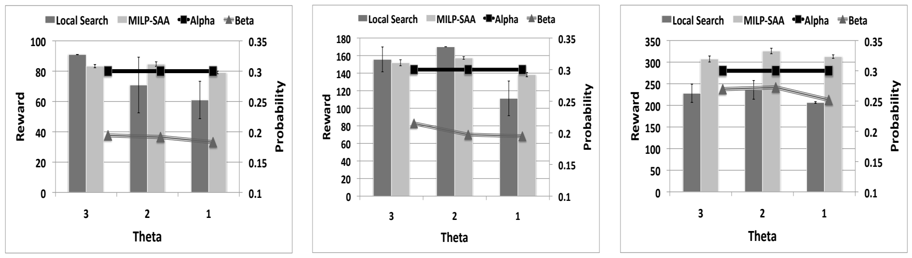
- As the number of SAA samples is increased, the $\beta$ value reduces. This is as expected as with the increasing number of samples, the SAA approximation becomes tighter.
- There is only a minor reduction in reward values obtained by MILP-SAA with the increasing number of samples. This shows that MILP-SAA can find a good solution that minimizes the probability of constraint violation even with increased problem complexity with the higher number of samples.
- Figure 2(d) shows the effect of increasing the number of samples for 20% horizon budget setting. We see that with 60 SAA samples, the probability of constraint violation $\beta$ is smaller than $\alpha$. Thus, increasing the number of samples can provide a feasible solution.

**Alpha$'$ and Theta:** Figures 3 and 4 indicate the impact of changing $\alpha'$ and the $\theta$ parameter of the gamma distribution on the performance of MILP-SAA in comparison to local search. The remaining parameters are fixed to their default setting (20).



(a) $|V| = 20$ (b) $|V| = 32$ (c) $|V| = 63$

Fig. 3: Effect on reward as $\alpha'$ is varied



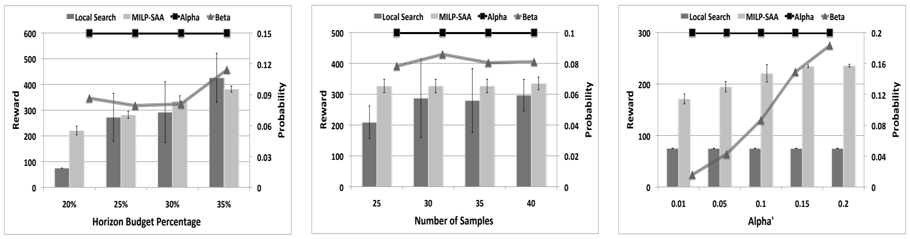
(a) $|V| = 20$ (b) $|V| = 32$ (c) $|V| = 63$

Fig. 4: Effect on reward as $\theta$ is varied

- As expected, with the increase in $\alpha'$, the empirical probability of constraint violation, $\beta$, increases. However, the increase in accumulated reward is minimal for increased $\alpha'$ values. This shows that a smaller value of $\alpha'$ is preferable to limit the probability of constraint violation.
- For a fixed budget percentage, as we increase the $\theta$ parameter of the gamma distribution, local search on average performs slightly better (albeit with higher standard deviation) than MILP-SAA in smaller problems (20 and 32 nodes). However, on the 63 node problems, we see that MILP-SAA is significantly better over all the values of $\theta$.

## 5.2 Real World Theme Park Problem

Lau *et al.* [10] introduced the route guidance problem for experience management at theme parks. Based on a year long data set of wait times at attractions in the theme park, they constructed best fit gamma distributions for travel times between attractions. In their work, the problem was formulated as a dynamic SOP and hence, there was a different travel time distribution for different time interval of the day. In contrast, we model the problem as a SOP and based on the same data set, we compute best fit gamma distributions for travel times between nodes over the entire time horizon. Extending our approach for dynamic SOPs remains an important area for future work.



(a) Horizon Budget (b) Number of Samples (c) Alpha′ ($\alpha'$)

Fig. 5: Solution quality comparisons on real-world theme park SOP

Figure 5 provides the results on the real world data set when horizon budget, number of samples and $\alpha'$ parameters are varied. Due to space constraints, we are unable to show the results where only one parameter is varied in each set of graphs. In fact, we show results with different values of $\alpha$ (0.1,0.15, 0.2) to indicate that, unlike in the synthetic data set, we do not get cases where $\beta$ exceeds $\alpha$.

- MILP-SAA consistently obtains higher average reward solutions in comparison to local search. In some cases, the reward improvement in using MILP-SAA was more than 100%. For instance, if we consider the case with 20% horizon budget in Figure 5 (a), the actual reward improvement is more than 125 and the simulated probability of constraint violation, $\beta$, is well below the $\alpha$. Similarly, in Figure 5(c), we obtain more than 100% improvement in solution quality over the local search approach, all the while keeping the $\beta$ within the limit.
- In most cases, the standard deviation in the solutions obtained with local search is significantly higher in comparison with MILP-SAA.
- Even with 25 samples, we obtain sufficiently stable solutions where the empirical probability of failure, $\beta$, is less than the $\alpha$.
- As the parameter $\alpha'$ employed by MILP-SAA approach is increased in Figure 5(c), as expected, the overall reward accumulated and probability of constraint violation increases.

To summarize, using extensive experiments, we analyzed a number of important properties of the MILP-SAA approach, such as the number of samples required and the effect of parameter $\alpha'$ on the feasibility and quality of the solution. Our approach provided significantly better results than the local search technique for both synthetic and real-world benchmarks.

## 6 Summary

In this paper, we have presented a new optimization based approach for solving risk aware Stochastic Orienteering Problems, where chance constraints represent risk attitude towards violating the given deadline. By approximating chance constraints with deterministic linear constraints, we provide a scalable approach that provides confidence based guarantees on solution quality. In addition, we show that our approach provides significantly superior strategies in comparison to an existing local search approach over a wide range of real world and synthetic problems.

## References

1. C. Archetti, D. Feillet, A. Hertz, and M. Speranza. The capacitated team orienteering and profitable tour problems. *Journal of the Operational Research Society*, 13 (1):49–76, 2008.
2. E. Balas. The prize collecting traveling salesman problem. *Networks*, 19:621–636, 1989.
3. A. Blum, S. Chawla, D. Karger, T. Lane, A. Meyerson, and M. Minkoff. Approximation algorithms for orienteering and discounted-reward TSP. *SIAM Journal on Computing*, 37(2):653–670, 2007.
4. A. Campbell, M. Gendreau, and B. Thomas. The orienteering problem with stochastic travel and service times. *Annals of Operations Research*, 186(1):61–81, 2011.
5. I.-M. Chao, B. Golden, and E. Wasil. Theory and methodology – the team orienteering problem. *European Journal of Operational Research*, 88:464–474, 1996.
6. M. Fischetti, J. J. S. Gonzlez, and P. Toth. Solving the orienteering problem through branch-and-cut. *INFORMS Journal on Computing*, 10:133–148, 1998.
7. B. Golden, L. Levy, and R. Vohra. The orienteering problem. *Naval Research Logistics*, 34(3):307–318, 1987.
8. A. Gupta, R. Krishnaswamy, V. Nagarajan, and R. Ravi. Approximation algorithms for stochastic orienteering. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2012.
9. G. Laporte and S. Martello. The selective traveling salesman problem. *Discrete Applied Mathematics*, 26:193–207, 1990.
10. H. C. Lau, W. Yeoh, P. Varakantham, D. T. Nguyen, and H. Chen. Dynamic stochastic orienteering problems for risk-aware applications. In *Proceedings of the International Conference on Uncertainty in Artificial Intelligence (UAI)*, 2012.
11. A. Leifer and M. Rosenwein. Strong linear programming relaxations for the orienteering problem. *European Journal of Operational Research*, 73:517–523, 1994.
12. B. Pagnoncelli, S. Ahmed, and A. Shapiro. Sample average approximation method for chance constrained programming: theory and applications. *Journal of Optimization Theory and Applications*, 142:399–416, 2009.
13. M. Schilde, K. Doerner, R. Hartl, and G. Kiechle. Metaheuristics for the biobjective orienteering problem. *Swarm Intelligence*, 3(3):179–201, 2009.
14. M. F. Tasgetiren. A genetic algorithm with an adaptive penalty function for the orienteering problem. *Journal of Economic and Social Research*, 4(2):1–26, 2001.
15. T. Tsiligrides. Heuristic methods applied to orienteering. *Journal of Operation Research Society*, 35(9):797–809, 1984.
16. P. Vansteenwegen and D. V. Oudheusden. The mobile tourist guide: An OR opportunity. *OR Insights*, 20(3):21–27, 2007.
17. Q. Wang, X. Sun, B. L. Golden, and J. Jia. Using artificial neural networks to solve the orienteering problem. *Annals of Operations Research*, 61:111–120, 1995.