Hybrid Sampling-Based Evaluators for the Orienteering Problem with Stochastic Travel and Service Times

Vassilis Papapanagiotou, Roberto Montemanni, and Luca M. Gambardella
Istituto Dalle Molle di studisull' Intelligenza Artificiale (IDSIA), Scuolauniversitaria Professionaledella Svizzeraitaliana (SUPSI), Università della Svizzeraitaliana (USI), Lugano, Switzerland
Email: {vassilis, roberto, luca}@idsia.ch

***Abstract*—Stochastic Combinatorial Optimization Problems (SCOPs) are many times used to model more accurately realistic situations. However, the stochasticity introduced also perplexes the computation of the objective function making it either difficult to solve or in our case very time-consuming. In this paper, we present different techniques of evaluating the objective function of the Orienteering Problem with Stochastic Travel and Service Times, that combine analytical, sampling and deterministic parts. We then compare these methods experimentally on well-known datasets.**

***Index Terms*—orienteering problem; stochastic optimization; objective function evaluation; monte carlo sampling**

I. Introduction

There has been an augmenting interest in the study of Stochastic Combinatorial Optimization Problems (SCOPs) in recent years. SCOPs are especially well suited as a model for realistic situations where some quantities are uncertain, such as travel and service times in real world applications. However, modeling stochasticity introduces complexities in the problem one of which is making objective functions very time consuming to compute. In this paper, we present an experimental comparison of different techniques to approximate the objective function of the Orienteering Problem with Stochastic Travel and Service Times (OPSTS).

Metaheuristics based on Monte Carlo Sampling have become very popular in the state-of-the-art approaches for dealing with different SCOPs such as OPSTS and the Probabilistic Traveling Salesman Problem with Deadlines [1], [2]. Monte Carlo Sampling is usually used to compute the objective function of these problems. One of the disadvantages observed in its use is that in the problems that incorporate a notion of deadline such as OPSTS, Monte Carlo Sampling is more error-prone near the deadline. The intuition behind this disadvantage is that Monte Carlo sampling rewards or penalizes each node with the whole reward or penalty based on the arrival time at each node. An error in the arrival time will propagate to subsequent nodes causing an error in the objective value returned by the objective function. This can have negative consequences when used in a metaheuristic because sometimes the error introduced is large enough to misguide the metaheuristic towards worse solution spaces than the ones it could have explored and exploited.

The OPSTS that is the focus of this paper, to our knowledge was first introduced in [3]. In this paper, the authors define the problem and develop two exact methods to solve a simplified version of the problem and a metaheuristic method to solve the original problem. The metaheuristic method described is a Variable Neighborhood Search (VNS) [4]. The objective function used is an analytical approximation of the probabilistic cost of the travel and service times, according to the Gamma distribution [5]. In this paper, this method is referred to as "ANALYTICAL" and it is used as a base function to compare our new approaches.

In [6]-[8] some alternative Monte Carlo sampling techniques for computing the objective function of OPSTS were presented and compared to the 'ANALYTICAL' approach of [1] in terms of time gain and error. In [7], even further techniques were presented in order to approximate the objective function of OPSTS more accurately. In [8], some tuning concerns about the proposed evaluators were addressed. Additionally, some evidence about the usefulness of the methods in a metaheuristic was presented.

Using sampling-based objective function evaluators has been studied in the past for different problems. Two of these problems, which are related to OPSTS, are the Probabilistic Traveling Salesman Problem and the Probabilistic Traveling Salesman Problem with Deadlines. A Monte Carlo sampling based Local Search algorithm for PTSP and PTSPD was presented in [9] and an improvement in [10]. The goal of this Local Search was to approximate the objective function faster than the traditional methods.

In this paper, we present and compare experimentally different hybrid sampling based objective function evaluators for the OPSTS. We extend previous works [6]-[8] by presenting a new objective function evaluator and new results and insights about the behavior and the usefulness of the methods.

Manuscript received February 1, 2015; revised April 17, 2015.

doi: 10.12720/jtle.3.2.108-114

II. PROBLEM DEFINITION

Let $N = \{1, ..., n\}$, be a set of n nodes-customers with the depot being node 0. We let D be a global deadline before which customers can be serviced without penalty. In other words, a customer i served before D, gives a reward $r_i$otherwise a penalty $e_i$is incurred. Part of the decision process of OPSTS is to select $M \subseteq N$. We assume that there is an edge (i,j) $\forall i, j \in N$. We assign at each edge a non-negative random variable $X_{i,j}$ denoting the travel time for node i to node j. Also, we define $S_i$as a random variable representing the service time of customer i. The probability distribution of $X_{i,j}$ is known $\forall\, i$, $j$and the service time $S_i$for the i^{th} customer follows the same distribution as $X_{i-1,i}$.For this reason $S_i$and $X_{i-1,i}$ can be added together and need not be computed separately. Therefore, in this paper travel and service times are simplified to travel times. The probability distribution of the random variables in this paper, is the Γ distribution [5], [11]. To configure the Γ distribution, we need 2 parameters, a shape parameter k (travel times in our case) and a scale parameter θ.In this paper, the second parameter $\theta$is evaluated to 1. The travel times $X_{i,j}$ are assumed independent from each other. We used theΓ distribution in order to have a direct comparison with the first OPSTS paper [1]. The Monte Carlo evaluator mentioned in this paper is independent of the distribution used, while the evaluators using an “ANALYTICAL” part introduced later in this paper need to satisfy a property that is examined in the next section.

We let the random variable $A_i$be the arrival time at customer i and $\bar{A}_i$ a realization of the variable A_i. We then define $R(\bar{A}_i)$ as a function representing the reward earned at customer i with arrival time $\bar{A}_i$. Therefore, $R(\bar{A}_i) = r_i$for $\bar{A}_i \leq D$, otherwise $R(\bar{A}_i) = -e_i$(for $\bar{A}_i$).

Let a tour τ be a sequence of customers of M. The objective function of the problem is defined as the maximization of the expected profit of the tour:

$$u(\tau) = \sum_{i \in \tau} [P(A_i \leq D) r_i - (1 - P(A_i \leq D)) e_i] \qquad (1)$$

III. HYBRID EVALUATORS

A. Analytical Evaluation (ANALYTICAL Method)

This evaluator is our reference evaluator and it is the same as the one used in [1]. In this section, we describe how this evaluator is derived analytically making the assumption that our travel times $X_{i,j}$ are Γ distributed as in [1].

A_i, the arrival time ata node is the sum of the travel times $X_{j,k}$ of all the edges (j, k) in the path from the depot to i. Therefore,A_i is a sum of $X_{j,k}$s which are Γ distributed variables. We know that if Y_i follows a $\Gamma(k_i, \theta)$ distribution for i = 1,2,...,N and all Y_i are independent then $\sum_{i=1}^{N} Y_i \sim \Gamma(\sum_{i=1}^{N} k_i, \theta)$ [4]. Therefore, when this property holds (it can hold for other distributions too like the normal) A_i can be approximated by the Γ function of the sum of all ks (travel times) of the relevant $X_{j,k}$ variables (travel times of edges in the path from the depot to node i in the solution under investigation).

Now, we need to come up with a closed-form expression for the computation of the objective function (1). To do this, we have to compute the probability $P(A_i \leq D)$. This probability can be calculated using the Cumulative Distribution Function (CDF) of A_i. We let $F_{k_i}(D)$ be the probability (Γ distributed) that customer i is visited before the deadline D.k_i is the k parameter for the i^{th} customer, that can be obtained by summing up all the k parameters (arrival times) of the edges on the path depot-i:

$$u(\tau) = \sum_{i \in \tau} [F_{k_i}(D) r_i - (1 - F_{k_i}(D)) e_i] \qquad (2)$$

B. Monte Carlo Evaluation (MC Method)

The objective function (1) can be evaluated if we first evaluate $P(A_i \leq D)$. To do this in the MC method, we use Monte Carlo sampling. First of all, many scenarios (fully-connected graphs) are generated with different arrival times ($\bar{A}_i$) for every node, obtained according to the probability distributions of each edge. Given a solution to evaluate, for each scenario we compute its deterministic objective function value. The objective function value (estimation of (1)) that the MC method returns is the mean value of the objective value of all scenarios generated.

The speed of MC method depends on the speed of generating random numbers of a specific distribution. We can greatly speed up this procedure by precomputing a large number of such random numbers. More specifically, we precompute samples of travel times from every node to every other node. The great benefit of the precomputation is that it is done once and the samples generated are reused throughout the computations multiple times. To do the precomputation we create amatrix for each sample. Each cell in each matrix represents a realization of a random variable $X_{i,j}$.

After we generate the samples we compute the objective function as follows. We use each sample as a graph distance matrix and we calculate the travel and service time and the total score of the path. If the travel time exceeds the deadline we subtract the respective penalty from our score, otherwise we add the respective reward.

C. Hybrid EvaluatorI (MC-ANALYTICAL-MCmethod, M-A-M)

In [8], it was first described that the ANALYTICAL method can be accelerated, while keeping the error low by partitioning the solution and evaluating each part with a different evaluator.

In OPSTS, we have stochastic travel times and thus some nodes are in positions that are considered critical. These nodes in each solution are the ones visited at a time close to the deadline D. These nodes will be visited on time in some scenarios and too late in others. We define a ‘deadline area’, a critical interval of time around the deadline D. In order to avoid big errors, we evaluate the nodes in this area using the ANALYTICAL method. To define the interval of the deadline area first we select a factor α ($0 < \alpha < 1$) and then we consider all the nodes that have deterministic travel times between $[(1-\alpha)\ D,$ $(1+\alpha)D]$. The rest of the nodes, outside the 'deadline area' are evaluated using Monte Carlo. An optimizer will avoid generating solutions with many "penalized" nodes; therefore, inside an optimizer there are usually very few nodes after the deadline area.

D. Hybrid EvaluatorII (REWARD-MC-ANALYTICAL-MC-PENALTYmethod, R-M-A-M-P)

MC-ANALYTICAL-MC was the first attempt to combine the speed of Monte Carlo sampling while keeping a very low error. One can make the additional observation that nodes at the beginning of the tour with deterministic arrival times much lower than the deadline *D*, have very high probability to gain a reward. Symmetrically, nodes towards the end of the tour with arrival times much higher than *D*, have high probability to incur a penalty. Analogously to M-A-M we define the reward-penalty area by selecting a factor ρ ($\alpha \leq \rho \leq 1$) and considering the nodes in the intervals [0, (1-ρ)D) and [(1+ρ)D, Inf] (where Inf is the last node) to get a reward and penalty respectively without any further computation.

Therefore, this method is parameterized by factors α and ρ. To sum up, it consists of the following areas: REWARD area: nodes in the interval [0, (1-ρ)D), they get instantly a reward, MC area: [(1-ρ)D, (1-α)D) they are a evaluated using Monte Carlo, ANALTYICAL area:[(1-α)D, (1+α)D), they are evaluated using analytical evaluator, MC area: [(1+α)D, 1-ρ)D), they are evaluated using Monte Carlo and finally PENALTY area: [(1+ρ)D, Inf] (where Inf is the last node) they incur a penalty without any further computation. When used inside an optimizer the second MC area and the PENALTY area tend to be very small to non-existent.

This method can provide fine-grained control and most of the times better performance than M-A-M with very small overhead.

E. Hybrid Evaluator III (REWARD-ANALYTICAL-PENALTYmethod, R-A-P)

This evaluator is first introduced in this paper and is a simplified version of the R-M-A-M-P evaluator without using any sampling part. As in M-A-M we define the deadline area by selecting a factor α ($0 < \alpha < 1$) and then adding in the area all the nodes that have deterministic travel times in the interval between $[(1-\alpha)\cdot D, (1+\alpha)\cdot D]$. Therefore, given the factor α the result of this evaluator is deterministic. This evaluator can act as a baseline for the usefulness of Monte Carlo sampling in the evaluators and is of practical importance in the cases where the deadline occurs rarely and a small analytical part can keep the error very low.

IV. EXPERIMENTS

In this section, we describe the experimental procedure used to compare the evaluators and we discuss the results.

A. Experimental Procedure

In order to compare the usefulness and the performance of the evaluators, we use 2 main metrics: speedup and relative error. Both metrics relate to the performance of the evaluator relatively to the analytical evaluator. We define speedup as follows:

$$Speedup = \frac{Runtime\_of\_Analytical}{Runtime\_of\_Evaluator}$$

If for example the result of speedup is 1.5, it means that the evaluator we examine was 1.5 times faster than the analytical method while evaluating the same solution.

We define the relative error as follows:

$$Relerror = \left|\frac{v_{Analytical} - v_{Evaluator}}{v_{Analytical}}\right|$$

where$v_{\text{Analytical}}$ is the value from the analytical objective function evaluator and $v_{Evaluator}$ is the value returned by the examined objective function evaluator.

B. Parameter Selection

We get measurements for different parameter configurations. First we specify the deadline area ratio range (DA range) and the reward-penalty area ratio range (RP range). For this paper the DA range is [0.1, 0.7] with step 0.1 and the RP range is [0.7725, 0.99] with step 0.0725. Then we run the VNS metaheuristic using the analytical evaluator and save 15000 different solutions produced. Each of these solutions is then evaluated by each evaluator 30 times and we obtain the mean value of the 2 metrics Speedup and Relerror for each evaluator.

We repeat this evaluation procedure for each combination (configuration) of deadline area ratio and reward-penalty ratio in the DA and RP range. For each method we select the configuration with the best average speedup with the constraint that the average relative error over all deadlines is below a threshold ("best avg methodology"); or for each method we eliminate all entries that do not comply with a given error threshold and for each deadline we select the configuration that yields the best speedup ("best methodology"). Both methodologies are explained in more detail in the next section.

C. Methods Comparison

In our experiments, we use the same datasets used in [1], with 21, 32and 64 customers named 221, 432, 664 respectively. Here we present only the most interesting results. To see the full results of these and other 3 datasets please visit [12]. In the experiments that follow, we want to compare the performance of the different hybrid evaluators.

As it was discussed in a previous section the metrics that are of importance are speedup and relative error. The desirable outcome is to select an evaluator with minimum relative error and maximum speedup. Optimizing for the 2 objectives simultaneously can be difficult since to reduce the relative error some speed is usually traded-off. However, considering that we want the objective function to be used inside a metaheuristic, it is often more important to have a faster metaheuristic so that more comparisons can be made for a given unit of time than having the minimum error possible. Some evidence of that can be found in [8]. Thus, in our experiments we fix a threshold for error and then compare which evaluator reaches the best speedup for this threshold. We use 2 ways to fix the threshold for error. The first method called "*best avg*" in the graphs selects the configuration of which the average speedup over all examined deadlines is maximum and the average relative error over all examined deadlines is less than the error threshold. The second method named "*best*" in the graphs selects the best speedup by deadline with the constraint that the relative error is less than the error threshold.

We present 2 types of graphs, one that shows the best performance of the method (speedup) given the above by deadline and a boxplot that shows the speedup performance of each method over all deadlines. For 4 error thresholds (0.5%, 1%, 2%, 5%) we plot both types of graphs for both methodologies (*best avg and best)* and present the best results for each threshold. We prefer presenting *best avg* if the results are not much different than *best* since it imposes smaller overhead in its computation. As it is clarified in a previous section, both metrics are relative to the ANALYTICAL method and one useful observation is that all the methods are statistically significantly faster than the ANALYTICAL even with the strictest error limitations (0.5% error).

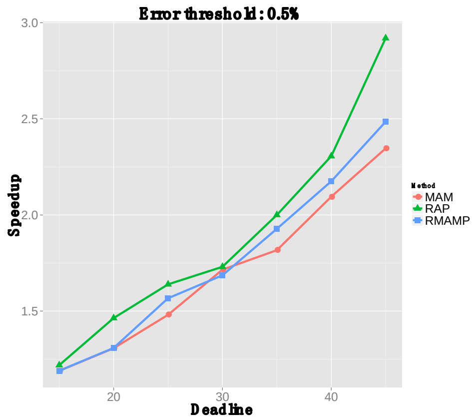

Figure 1. Deadline vs Speedup for dataset 221 (21 customers), tuning by each deadline separately (best). Error threshold 0.5%

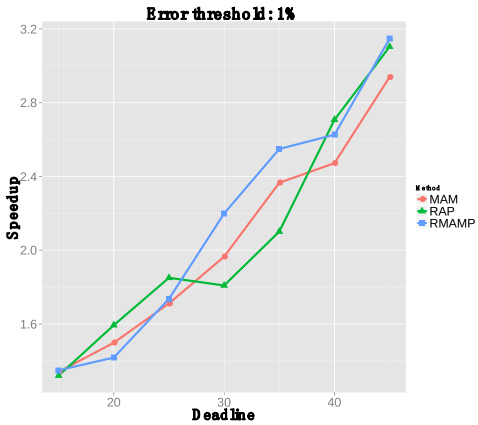

Figure 2. Deadline vs Speedup for dataset 221 (21 customers), tuning by each deadline separately (best). Error threshold 1%

Dataset 221:

For an error threshold of 0.5% (Fig. 1) for the *best* methodology we can observe that the R-A-P method is more performant than the others for every deadline.

For an error of 1% (Fig. 2), R-M-A-M-P is best on average for the best *methodology*

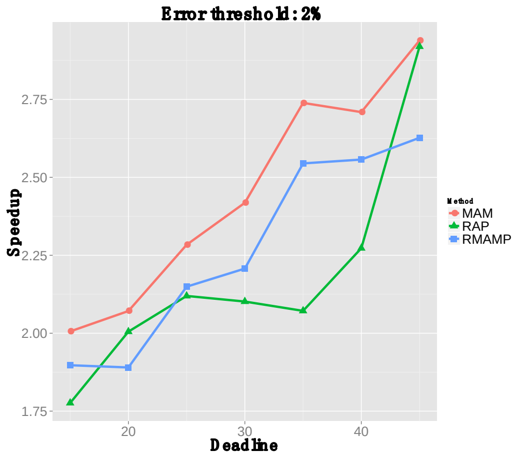

Figure 3. Deadline vs Speedup for dataset 221 (21 customers), tuning by according to average relative error and speedup (*best avg)*. Error threshold 2%

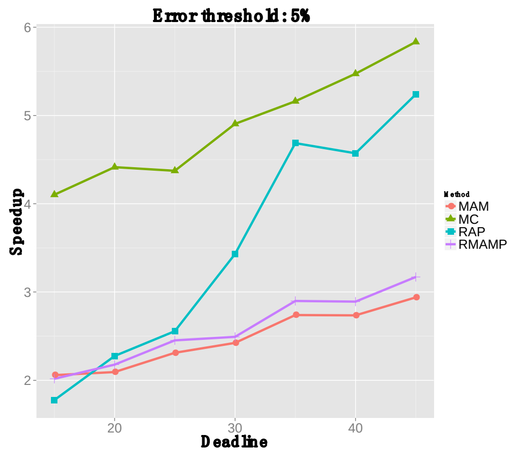

Figure 4. Deadline vs Speedup for dataset 221 (21 customers), tuning by each deadline separately (best). Error threshold 5%

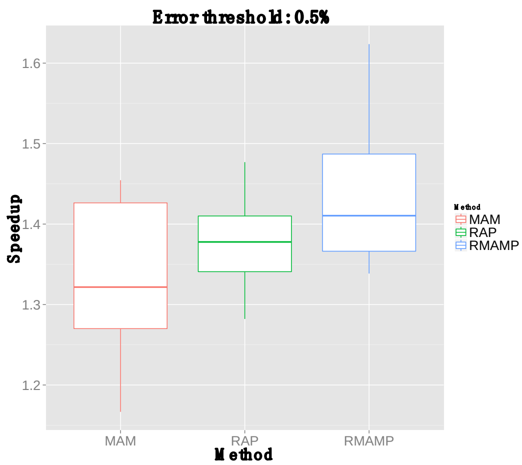

Figure 5. Methodvs Speedup for dataset 432 (32 customers), tuning by according to average relative error and speedup (*best avg)*. Error threshold 0.5%

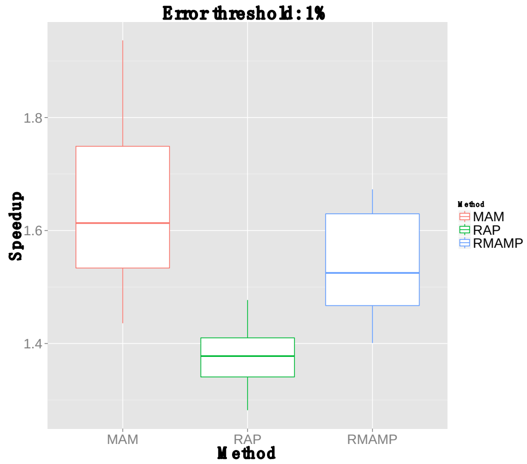

Figure 6. Methodvs Speedup for dataset 432 (32 customers), tuning by according to average relative error and speedup (*best avg)*. Error threshold 1%

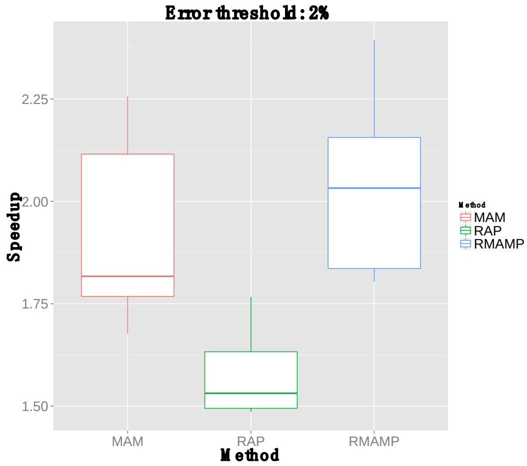

Figure 7. Methodvs Speedup for dataset 432 (32 customers), tuning by according to average relative error and speedup (*best avg)*. Error threshold 2%

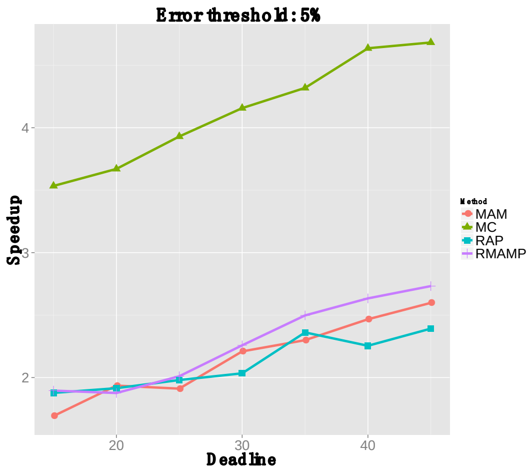

Figure 8. Deadline vs Speedup for dataset 432 (32 customers), tuning by according to average relative error and speedup (*best avg)*. Error threshold 5%

For an error of 2% (Fig. 3) and the *best avg* methodology, the M-A-M method is has the best speedup on average.

For an error of 5% (Fig. 4) MC is significantly better than the other methods but an error of 5% usually is too high and can impact the metaheuristic in many negative ways.

Dataset 432:

In this dataset,we observe that for error thresholds of 0.5% and 2% R-M-A-M-P is better on average (Fig. 5 and Fig.7).

For 2% error threshold both M-A-M and R-M-A-M-P are significantly better than R-A-P (Fig.7).

For error threshold of 1% M-A-M is better on average and significantly faster than R-A-P (Fig. 6).

For error threshold of 5% MC is statistically significantly better than any other method (Fig. 8).

Dataset 664:

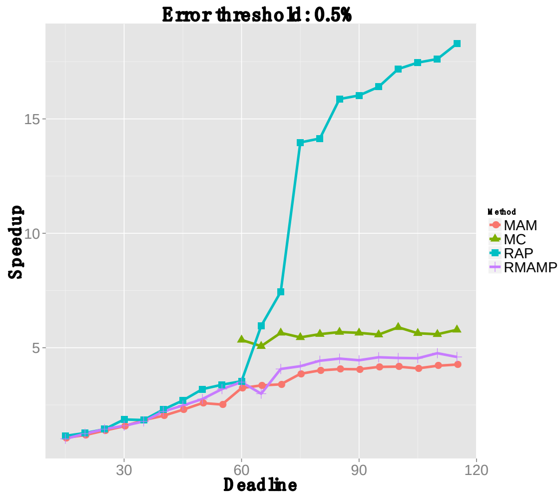

Figure 9. Deadlinevs Speedup for dataset 664 (64 customers), tuning by each deadline separately (best). Error threshold 0.5%

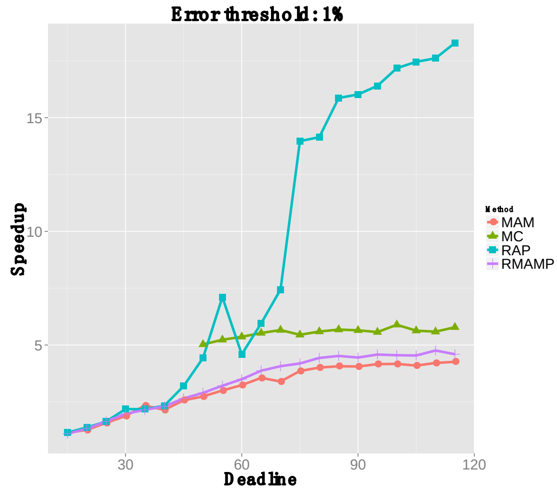

Figure 10. Deadlinevs Speedup for dataset 664 (64 customers), tuning by each deadline separately (best). Error threshold 1%

In this dataset we observe that from a certain deadline onwards, R-A-P is much faster than the other methods. This occurs because in this instance, the analytical part of R-A-P is enough to keep the error very low and from a certain deadline upwards, the deadline is rarely occurring making the analytical part faster and needing a very small analytical part to achieve good results. This dataset is a good use case for the utility of a method like R-A-P which does not include any sampling. We also observe that in some graphs the MC method is present but with missing points (Fig. 9, Fig. 10, Fig. 11, Fig. 12). This means that for the specific deadlines there was no result below the selected error threshold.

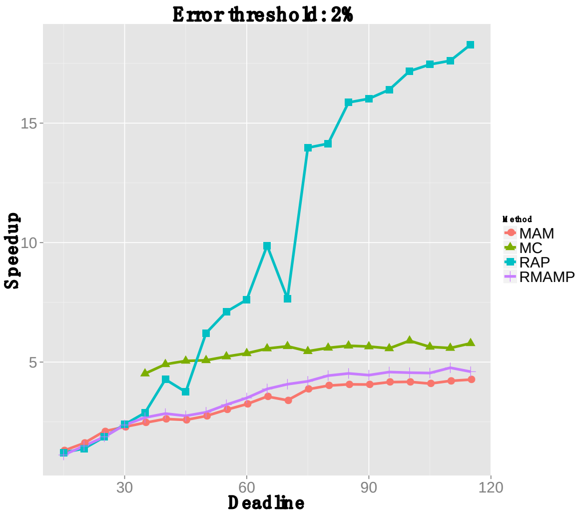

Figure 11. Deadlinevs Speedup for dataset 664 (64 customers), tuning by each deadline separately (best). Error threshold 2%

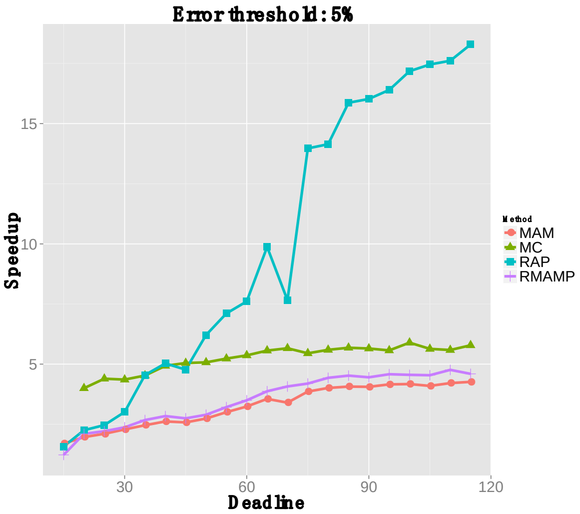

Figure 12. Deadlinevs Speedup for dataset 664 (64 customers), tuning by each deadline separately (best). Error threshold 5%

D. General Remarks& Guidelines

Using the representative results presented and also the full results [11], one can see that all methods are useful and significantly faster than the ANALYTICAL one. According to the needs of the application we showed two methodologies on how to select the best-suited method.

R-M-A-M-P usually yields consistent results and because it provides fine-grained control for balancing error and speedup, it is in many cases the method of choice for high performance with low error.

M-A-M is more accurate but slower than MC and it beats R-M-A-M-P in limited cases where the M-A-M part of the solution is enough to yield good results and the overhead to divide the solution in more parts (5 in the case of R-M-A-M-P, 3 in the case of M-A-M) makes R-M-A-M-P less efficient.

R-A-P is very efficient when tuned by each deadline separately in datasets where a small ANALYTICAL part is enough to reduce the error of evaluation to minimal levels. Usually, this happens in datasets when the deadline is large enough and occurs rarely, In this case, most of the solution is evaluated using the deterministic REWARD part, which is very fast.

MC is usually the method of choice when large error is acceptable and the deadline occurs often. In most of these cases MC is significantly better than the other methods, however, usually the error threshold in these cases is unacceptable for use in a metaheuristic.

V. Conclusions & Future Work

In this paper, we presented and compared different hybrid objective function evaluators for the Orienteering Problem with Stochastic Travel and Service Times. All of these techniques accelerate significantly the evaluation of the objective functionwhich is most times the performance bottleneck when trying to find solution in Stochastic Combinatorial Optimization Methods. We presented different ways to compare the evaluators, provided guidelines how to select and tune the evaluators and showed the usefulness of each one. In the future, we intend to explore the behavior of these methodswhen embedded inmetaheuristics and when used in different problems with similar structure.

Acknowledgment

Vassilis Papapanagiotou was supported by the Swiss National Science Foundation through project 200020-156259/1:" Hybrid Sampling-based Metaheuristics for Stochastic Optimization Problems with Deadlines".

References

[1] P. Brucker, A. Drexl, R. Möhring, K. Neumann, and E. Pesch, "Resource-constrained project scheduling: Notation, classification, models, and methods," *European Journal of Operational Research*, vol. 112, no. 1, pp. 3–41, 1999.

[2] D. Weyland, R. Montemanni, and L. M. Gambardella, "Hardness results for the probabilistic traveling salesman problem with deadlines," in *Combinatorial Optimization, ser. Lecture Notes in Computer Science*, Springer Berlin Heidelberg, 2012, vol. 7422, pp. 392–403.

[3] A. M. Campbell, M. Gendreau, and B. W. Thomas, "The orienteering problem with stochastic travel and service times," *Annals of Operations Research*, vol. 186, pp. 61–81, 2011.

[4] Z. Sevkli and F. E. Sevilgen, "Variable neighborhood search for the orienteering problem," in *Computer and Information Sciences–ISCIS 2006*. Springer, 2006, pp. 134–143.

[5] WIKIPEDIA: Gamma Distribution. [Online]. Available: http://en.wikipedia.org/wiki/Gamma_distribution

[6] V. Papapanagiotou, R. Montemanni, and L. M. Gambardella, "Objective function evaluation methods for the orienteering problem with stochastic travel and service times," *Journal of Applied Operational Research*, vol. 6, no. 1, pp. 16 – 29, 2014.

[7] V. Papapanagiotou, R. Montemanni, and L. M. Gambardella, "The orienteering problem with stochastic travel and service times, new approaches to sampling-based objective function evaluation," IN *Proc. 4th Annual Conference on Computational Mathematics, Computational Geometry & Statistics*, 2105.

[8] V. Papapanagiotou, R. Montemanni, and L. M. Gambardella, “Sampling-based objective function evaluation techniques for the orienteering problem with stochastic travel and service times,” in *Proc. International Conference on Operations Research*, Aachen, 2014.
[9] M. Birattari, P. Balaprakash, T. Stützle, and M. Dorigo, “Estimation-based local search for stochastic combinatorial optimization,” Technical Report, IRIDIA, Université Libre de Bruxelles, 2007.
[10] M. Birattari, P. Balaprakash, T. Stützle, and M. Dorigo, “Estimation-based local search for stochastic combinatorial optimization using delta evaluations: A case study on the probabilistic traveling salesman problem,” *INFORMS Journal on Computing*, vol. 20, no. 4, pp. 644–658, 2008.
[11] D. Wackerly, W. Mendenhall, and R. Scheaffer. “Mathematical statistics with applications,” *Cengage Learning*, 2007. pp. 185-193
[12] [Online]. Available: http://people.idsia.ch/~papapanagio/

Vassilis Papapanagiotou is currently a Ph.D student at the DalleMolle Institute for Artificial Intelligence (IDSIA) and at the University of Lugano, Switzerland. He obtained a Diploma of Engineering and a Master’s degree in Computer Engineering and Informatics at the University of Patras, Greece. His main research interests are in stochastic optimization problems, metaheuristics and mathematical programming modeling.

Roberto Montemanni is Professor of Basic and Robust Algorithms at the University of Applied Sciences of Southern Switzerland. He is active as a Senior Researcher at IDSIA and is an External research advisor for PhD students at the University of Lugano. He obtained a Laurea degree in Computer Science from the University of Bologna, Italy in 1999 and a Ph.D. in Applied Mathematics from the University of Glamorgan, UK in 2002. His main research interests are in mathematical programming modeling and heuristic algorithms development for applied optimization problems.

Luca Maria Gambardella is Co-Director of IDSIA, DalleMolle Institute for Artificial Intelligence in Manno, Lugano. He has proposed and published several state of the art bio-inspired distributed optimisation algorithms for the travelling salesman problem, the vehicle routing problem with time windows, the quadratic assignment problem, the sequential ordering problems, and the flexible job shop problem. He is leading projects at the Swiss level (Swiss National Science Foundation, CTI) and the international level (EU Commission).