Proceedings of the 2002 Winter Simulation Conference
E. Yücesan, C.-H. Chen, J. L. Snowdon, and J. M. Charnes, eds.

TWO-STAGE NP METHOD WITH INHERITANCE

Jumi Kim
Sigurdur Ólafsson

Department of Industrial and Manufacturing Systems Engineering
Iowa State University
2019 Black Engineering
Ames, IA 50011, U.S.A.

ABSTRACT

The nested partitions method is a flexible and effective framework of optimizing large-scale problems with combinatorial structure. In this paper we consider the nested partitions method for simulation optimization and propose a new variant that uses inheritance to speed convergence. The new nested partitions method with inheritance algorithm performs well for when applied to test problems but it also calls for new analysis of convergence.

1 INTRODUCTION

Simulation optimization has become a very popular and in particular interest in simulation optimization of discrete problems has grown rapidly over the past decade (Andradóttir, 1998; Fu, 1994; Jacobson and Schruben, 1989). Many methods that are designed for combinatorial optimization have been applied to simulation, including simulated annealing, genetic algorithms, tabu search, and neural networks (April et al., 2001). Such randomized metaheuristics have indeed to be found to be the most effective for practical problems and have been incorporated into numerous commercial simulation packages.

This paper deals with the nested partitions (NP) framework that has been found to be effective for numerous applications and is specifically designed for simulation optimization and hence accounts explicitly for the inherent randomness (Ólafsson, 1999; Ólafsson and Kim, 2001; Ólafsson and Shi, 2000; Shi and Ólafsson, 1997; 1998; 2000). The NP method uses iterative partitioning of the feasible region to narrow the focus of the optimization search and concentrate the computational effort where good solutions are likely to be found. In the NP variants that have been proposed in the literature to date, these iterations are done independently, which results in the NP search generating a simple homogeneous Markov chain. Thus, a simple but eloquent Markov chain analysis can be used to prove asymptotic convergence almost surely, to calculate the expected number of iterations it takes to converge, and to provide probabilistic statements regarding finite time performance (Ólafsson, 1998; 1999). However, this also means that some information is lost between iterations. The selection of a move is based on the good performance of certain solutions that are randomly sampled. When independence is enforced between iterations, these good solutions are necessarily discarded (or at least not used for the search). Intuitively, however, it has certain appeal to keep a given fraction of the best solutions from one iteration to the next. The downside is that the simple Markov chain analysis is no longer possible.

The NP framework can also be combined with local search heuristics, such as genetic algorithms to speed the search (Ólafsson and Kim, 2001). When this is done, the motivation for keeping good solution is even more compelling. For example, when using genetic algorithms, the issue is if the final population in each iteration should be discarded or if the fittest individuals from each iteration should be allowed to survive to the next iteration. In this paper we present a new framework for NP that used genetic algorithms and inheritance and address both practical performance and convergence analysis issues related to this new NP variant.

The remainder of this paper is organized as follows. In Section 2 we present the basis of the NP framework and in Section 3 we show how to use inheritance to extend the NP method and improve its computational efficiency. In Section 4 we address the critical issue of how to determine the amount of computational effort in each iteration of the method. Some preliminary results from simulation experiments are reported in Section 5, and Section 6 contains some concluding remarks.

2 THE NP ALGORITHM

In mathematical notation, we want to solve the problem

$$\min_{\theta \in \Theta} J(\theta), \tag{1}$$

where Θ is a finite feasible region, and $J : \Theta \rightarrow \mathbf{R}$ is a performance function that is subject to noise. In other words, for any feasible point $\theta \in \Theta$, $J(\theta)$ cannot be evaluated analytically. Often $f(\theta)$ is an expectation of some random estimate of the performance of a complex stochastic system given a parameter θ, that is, $J(\theta) = E[L(\theta)]$. Here $L(\theta)$ is a random variable which depends on the parameter $\theta \in \Theta$. We assume that $L(\theta)$ is a discrete event simulation estimate of the true performance, and refer to it as the sample performance.

In a recent paper, Shi and Ólafsson (2000) introduced an optimization method, the *nested partitions* (NP) method, for global optimization when the objective function is deterministic. In this context, the method has been found to be quite efficient for combinatorial optimization (Ólafsson and Shi 2000). Furthermore, as was first suggested by Shi and Ólafsson (1997), this method can also be applied to stochastic problems, where no analytical expression exists for the objective function and it must be evaluated using simulation. In Ólafsson (1999), this method is further improved by drawing on ideas from statistical sampling techniques that have proven useful in simulation in the past, namely ranking-and-selection methods. Thus, the resulting algorithm combines statistical sampling techniques traditionally used for comparing a few alternatives with a global optimization framework aimed at large-scale optimization problems.

The basic idea of the method is to systematically partition the feasible region into subsets and focus the computational effort in those subsets that are considered promising. The main components of the method are:

- **Partitioning**: at each iteration the feasible region is partitioned into subsets that cover the feasible region but concentrate the search in what is believed to be the most promising region.
- **Random sampling**: to evaluate each of the subsets, a random sample of solutions are obtained from each subset and used to estimate the performance of the region as a whole.
- **Local improvement**: to improve the estimate of each of the subset, the sample points can be used as starting points for a local search procedure that is constrained within the region.

This method can be understood as an optimization framework that combines adaptive global sampling with local heuristic search. It uses a flexible partitioning method to divide the design space into regions that can be analyzed individually and then aggregates the results from each region to determine how to continue the search, that is, to concentrate the computational effort. Thus, the NP method adaptively samples from the entire design space and concentrates the sampling effort by systematic partitioning of the design space.

The key features in determining how to implement the method is developing a partitioning method, deciding how much sample effort to use in each region, and how much local search effort to use in each iteration. These factors are of course interconnected. A high quality partition will lessen the need for sampling and local search, and in general increased effort along one of these dimensions decreases the need for the other two. Implementing the NP method can therefore be quite problem dependent, in particular, partitioning schemes that have been devised in the past have drawn heavily on specific structure related to the application itself. This, however, requires substantial effort on part of the practitioner using the method, and in this paper we present a new framework for automating these decision, namely an intelligent partitioning method, guided random sampling, and guided local search.

In each iteration of the NP method it maintains what is called the most promising region, that is, a sub-region that is considered the most likely to contain the best solution. This most promising region is partitioned into a given number of sub-regions, these sub-regions and the surrounding region are sampled using random sampling, and the sampling information used to determine which region should be the most promising region in the next iteration.

As opposed to purely heuristic optimization methods, the NP method guarantees that the optimum solution is eventually found (Shi and Olafsson, 2000). Furthermore, Olafsson (2000) uses standard ranking and selection procedures develop and algorithm that allows us to pre-specify a probability, say 90% or 95%, and terminate the algorithm when the probability that the correct solution has been selected exceeds this value. Here the correct subset of best features is defined as a subset of features that has a performance that is within a certain distance, that is an indifference zone, of the optimal performance. The key to this result is to guarantee in each iteration of the algorithm that the correct move is made with a minimum probability P^*, which can be calculated numerically from the following equation The termination of the algorithm when solution with an indifference zone has been found with a certain probability should be appealing to many practitioners, as this is widely known in the comparison of two or more systems.

3 INCORPORATING INHERITANCE

As stated in the previous section, the NP method can use local search to improve the estimates of the best solution in each region. The search uses the randomly generated sample points as starting points and continues until some predetermine criteria is satisfied. In other words, an initial set of sample points $\mathcal{D}^{(k,j)}$ is obtained from each region

$\sigma_j(k)$ in the k-th iteration and then some local search method is used to transform this set into a new set $\mathcal{L}(\mathcal{D}^{(k,j)})$ that is presumably an improvement and in particular one would expect that $\min_{\theta\in\mathcal{L}(\mathcal{D}^{(k,j)})} f(\theta) < \min_{\theta\in\mathcal{D}^{(k,j)}} f(\theta)$.

In general, any local search method can be used, for example taking each of the points in the sample point set $\mathcal{D}^{(k,j)}$ as a starting point, that is,

$$\mathcal{L}(\mathcal{D}^{(k,j)}) = \{\mathcal{L}(\theta)\}_{\theta\in\mathcal{D}^{(k,j)}},$$

where $\mathcal{L}(\theta)$ is the final solution after starting the local search at an arbitrary initial solution $\theta \in \mathcal{D}^{(k,j)}$. However, as the algorithm already starts with a set of points, using a genetic algorithm (GA) is a natural companion and we will do so in our framework.

In the context of using GA within the NP framework, the initial set of random samples, $\mathcal{D}^{(k,j)}$ is the initial population of the genetic algorithm, and the GA search then progresses through a sequence of populations $\mathcal{D}_1^{(k,j)}$, ..., $\mathcal{D}_n^{(k,j)}$ until some stopping criterion is satisfied. The final population $\mathcal{D}_n^{(k,j)}$ of the genetic algorithm is the set of points used by the NP method to calculate the performance measures for each region.

The critical issue of inheritance is now what to do with the final population $\mathcal{D}_n^{(k,j^*)}$ that was used to select the next most promising region $\sigma(k+1) = \sigma_{j^*}(k)$. Now clearly, $\mathcal{D}_n^{(k,j^*)} \subset \sigma(k+1)$ and the best point in $\mathcal{D}_n^{(k,j^*)}$ is the best solution that was found during this iteration. Discarding such good solutions appears somewhat inefficient and thus we want to investigate if it is beneficial to inherit all or a fraction of $\mathcal{D}_n^{(k,j^*)}$ to the next iteration. Note that in the next iteration, all the solutions in $\mathcal{D}_n^{(k,j^*)}$ fall into one of the subregions of $\sigma(k+1)$. Thus, by only looking at $\mathcal{D}_n^{(k,j^*)}$ no solutions are inherited to the surrounding region which may induce a bias that makes it difficult to backtrack. However, high quality solutions also exist that will fall into the next surrounding region as

$$\bigcup_{j\neq j^*} \mathcal{D}_n^{(k,j)} \subset \Theta \setminus \sigma(k+1)$$

and many good solutions should be contained in this set and can be inherited to the next iteration.

Summarizing the above thoughts, we now have the following NP algorithm with inheritance and GA search:

3.1 NP with Inheritance

Step 0 Initialize $\sigma(0) = \Theta$, $k = 1$, $\mathcal{D}_n^{(0,j)} = \emptyset$, $j = 1, 2, ..., m$. Let p be the fraction of the initial population to be inherited.

Step 1 Partition $\sigma(k)$ into $m-1$ subregions $\sigma_j(k)$ and aggregate the surrounding region into one $\sigma_m(k) = \Theta \setminus \sigma(k)$.

Step 2 Let $\widetilde{\mathcal{D}}_n^{(k-1,j^*(k))}$ be p% best points in $\mathcal{D}_n^{(k-1,j^*(k))}$. Inherit these points from the previous iteration:

$$\mathcal{D}_0^{(k,j)} = \left\{\theta \in \widetilde{\mathcal{D}}_n^{(k-1,j^*(k))} | \theta \in \sigma_j(k)\right\}, \quad (2)$$

$j = 1, 2, ..., m-1$. Similarly, inherit point from $\bigcup_{j\neq j^*} \mathcal{D}_n^{(k,j)}$ to $\sigma_m(k)$.

Step 3 Randomly sample

$$n_0 - \left|\mathcal{D}_0^{(k,j)}\right| \quad (3)$$

points from $\sigma_j(k)$ and add to the set $\mathcal{D}_0^{(k,j)}$, $j = 1, 2, ..., m$.

Step 4 Apply n steps of a GA search to the set of points from each region, generating a final population

$$\mathcal{D}_n^{(k,j)} = \mathcal{L}\left(\mathcal{D}_0^{(k,j)}\right), \quad (4)$$

$j = 1, 2, ..., m$.

Step 5 Obtain simulation estimates $L(\theta)$ of the performance of each of the solutions in each of the final sets, $\theta \in \bigcup_j \mathcal{D}_n^{(k,j)}$.

Step 6 Find the region that looks the most promising after the GA search:

$$j^*(k) = \arg\min_j \min_{\theta\in\mathcal{D}_n^{(k,j)}} L(\theta). \quad (5)$$

Step 7 Select the next most promising region by either backtracking if the surrounding region is best ($j^*(k) = m$) or moving to the appropriate subregion if one of those is the best:

$$\sigma(k+1) = \begin{cases} \sigma(k-1), & \text{if } j^*(k) = m, \\ \sigma_{j^*(k)}(k), & \text{otherwise.} \end{cases} \quad (6)$$

Step 8 If $\sigma(k+1)$ is a singleton terminate the search and let the estimate of the best solution be

$$\hat{\sigma} = \sigma(k+1). \quad (7)$$

Otherwise let $k = k+1$ and go back to Step 1.

Of course there is a great deal of detail that is omitted from Step 4 as the parameters of the GA search must be specified, that is, the crossover and mutation operators should be defined, as well as probabilities of survival, etc. However, this should be done as for any GA search and is

application specific. It will thus not be discussed further here. The issue of when to terminate the GA search in Step 4 is, however, of special interest and requires different considerations. This will be addressed in the next section.

4 TERMINATING THE GA SEARCH

Within the NP framework a critical issue is how much effort should be put into each iteration, that is, how well each region should be explored before deciding which region should become the most promising region in the next iteration.

This calculation of the amount of random sampling is a key issue in the NP methodology in general and as its determination is somewhat difficult. Also note that when applying the NP method to a simulation optimization problem there are two sources of randomness that complicate the selection of the correct subset. First, there is a sampling error due to a relatively small sample being used to estimate the performance of an often large set. Secondly, the performance of each sample points is estimated using simulation and is hence noisy. It is important to observe that the former of these elements implies that the variation within a subset differs greatly from one region to the next. As an extreme case consider a singleton region that is being compared to the entire surrounding region. That is, a region containing only one solution being compared to a region containing all of the other solutions. Clearly the first source of randomness has been completely eliminated in the singleton region, whereas it probably accounts for almost all of the randomness in the surrounding region. This implies that to make better use of the sampling effort the number of sample points from each region should be variable and dependent on the variation within the region.

In the case of NP without inheritance this can be addressed by assuming that enough exploration is done to make the correct selection with a fixed probability $P^* > \frac{1}{2}$. With this assumption, Ólafsson (1999) uses a random walk analysis to show that the probability of terminating correctly when first encountering a singleton, that is the first singleton $\hat{\sigma}$ is really the optimal solution σ^*_{opt}, is given by the following equation:

$$P\left[\hat{\sigma} = \sigma^*_{opt}\right] = (P^*)^{d^*} \frac{(1-P^*)^{d^*} - (P^*)^{d^*}}{(1-P^*)^{2d^*} - (P^*)^{2d^*}}, \qquad (8)$$

where d^* is the depth of the tree generated by the iterative partitions, that is, the maximum number of partitions needed until the regions become singletons.

Equation (8) assumes that the correct selection is made with a fixed probability and this must therefore be somehow assured. Ólafsson (1999) suggests to use a ranking-and-selection procedure (Goldman and Nelson, 1998) to prescribe the number of sample points that are needed to make the correct selection, within an indifference zone, with at least probability P^*. Thus, the left hand side of (8) becomes a lower bound on the probability of terminating correctly within an indifference zone.

We suggest a similar approach for the NP with Inheritance algorithm. Note that this is equivalent to determine how much effort should be devoted to the GA search, that is, how many populations the GA should generate. Also note that the usual stopping rules for GA may not be appropriate. When implementing GA as an independent optimization algorithm it may be appropriate to continue the search while better solutions can be found by the algorithm, and for example use a stopping rule that terminates the search when no improvement has been found for a fixed number of iterations. However, when using GA to select regions within the NP framework we may want to terminate the GA search even it could still make improvements. Specifically, as soon as the correct region can be selected with sufficiently high confidence (or probability) then the GA search should be terminated even if it could make further progress.

Given the above discussion, we can now elaborate on Step 4 in the NP with Inheritance algorithm:

Step 4 Given an initial population $\mathcal{D}_0^{(k,j)}$ for each region $j = 1, 2, ..., m$.
Set $l = 0$.
Loop
Set $l = l + 1$.
For every $j = 1, 2, ..., m$
Apply one GA step to $\mathcal{D}_l^{(k,j)}$.
Simulate the performance of all $\theta \in \mathcal{D}_l^{(k,j)}$.
Calculate the mean $\mu_l^{(k,j)}$ and variance $Var_l^{(k,j)}$ of this population.
End for
For every $j = 1, 2, ..., m$
Use a statistical selection procedure and $\mu_l^{(k,j)}$, $Var_l^{(k,j)}$, $j = 1, 2, ..., m$ to determine if more exploration is needed from $\sigma_j(k)$.
End for
Until no more exploration is needed.
Return the final populations for each region.

As before we have left out the details of the GA search itself (crossover, mutation, etc) as it is problem specific. We have also not specified which statistical selection procedure should be used.

In past NP variants (without inheritance), two-stage procedures such as Rinott's procedure (Rinott, 1978; Ólafsson, 1999) and the Nelson-Matejcik procedure (Matejcik and Nelson, 1995; Ólafsson and Kim, 2001) have been

used. As these procedure guarantee correct selection within an indifference zone with a predetermined probability, they fit nicely with the analysis of NP without inheritance and equation (8) can be used. Here, however, the random walk analysis that lead to this equation can no longer be applied as the inheritance causes a dependency between iterations that means that there is no longer an underlying homogeneous Markov chain structure to be used for the analysis.

Due to the lack of independence, previously use ranking and selection methods may not be the best choice now for controlling the search, and another approach to making the correct selection is to use Bayesian analysis (Chick, 1997; Chick and Inoue, 1999; 2000). This approach does not guarantee a correct selection probability within an indifference zone, but may be more consistent with NP with Inheritance as inheriting points simply means that there is an informative prior before more effort is put into each iteration to obtain a new posterior distribution.

5 PRELIMINARY NUMERICAL RESULTS

We have conducted some simulation experiments for optimizing simple queuing systems to indicate the promising of the proposed inheritance feature in the NP algorithm. The preliminary results indicate the following:

- Inheritance can significantly speed convergence with common speedup factor being 20% to 30%.
- The quality of the solution is not affected by the inheritance feature.
- The optimal level of inheritance seems to be surprisingly low. For the small test problems, just inheriting one or two of the best solutions tends to have the best performance.

Detailed results of the simulation experiments will be reported later.

6 CONCLUSIONS

In this paper we have looked at the NP method for simulation optimization and proposed a new variant that incorporates inheritance between iterations to speed convergence. A detailed algorithm is presented for the new method, which also incorporates genetic algorithm search, and the primary theoretical and practical issues involved in adding the inheritance feature are discussed.

From convergence analysis standpoint the main issue is that independence from one iteration to the next is needed for all of the prior analysis for the NP method, and this independence no longer holds for the NP with Inheritance algorithm. Thus a new approach must be investigated and we propose that this could be based on using Bayesian analysis to guide the search, that is, to determine when a new region should be selected. This topic requires further research.

The empirical performance of using inheritance is promising with significant speedups in the running time of the algorithm. One of the most interesting elements of our initial experiments, however, is that the best levels of inheritance are very low, perhaps one one or two of the best solutions. The explanation for this may be that the NP method uses the best point from each region to determine which region becomes the most promising in the next iteration, and this is the critical decision in each iteration. However, more experimental results are need to confirm this behavior and more analysis is needed to obtain a satisfactory explanation of why small amount of inheritance are preferable.

REFERENCES

Andradóttir, S. 1998. "A Review of Simulation Optimization Techniques," in *Proceedings of the 1998 Winter Simulation Conference,* 151-158.

April, J., F. Glover, J. Kelly, and M. Laguna. 2001. "Simulation/Optimization using "Real-World" Applications," in *Proceedings of the 2001 Winter Simulation Conference,* 134-138.

Chick, S.E. 1997. "Selecting the Best System: A Decision Theoretic Approach," in *Proceedings of the 1997 Winter Simulation Conference,* 326-333.

Chick, S.E. and K. Inoue. 1999. "A Decision-Theoretic Approach to Screening and Selection with Common Random Numbers," in *Proceedings of the Winter Simulation Conference,* 603-610.

Chick, S.E. and K. Inoue. 2000. "New Results on Procedures that Select the Best System using CRN," in *Proceedings of the Winter Simulation Conference,* 603-610.

Fu, M.C. 1994. "Stochastic Optimization using Simulation: A review," *Annals of Operations Research*, 53, 199-248.

Goldsman, D. and B.L. Nelson. 1998. "Statistical Screening, Selection, and Multiple Comparison Procedures in Computer Simulation," in *Proceedings of the 1998 Winter Simulation Conference,* 159-166.

Jacobson, S.H. and L.W. Schruben. 1989. "A Review of Techniques for Simulation Optimization," *Operations Research Letters*, 8, 1-9.

Matejcik, F.J. and B.L. Nelson. 1995. "Two-Stage Multiple Comparisons with the Best for Computer Simulation," *Operations Research*, 43, 633-640.

Ólafsson, S. 1999. "Iterative Ranking-and-Selection for Large-Scale Optimization," in P.A. Farrington, H.B. Nembhard, D.T. Sturrock, and G.W. Evans (eds.), *Proceedings of the 1999 Winter Simulation Conference,* 479-485.

Ólafsson, S. and J. Kim. 2001. "Towards a Framework for Black-Box Optimization," in *Proceedings of the 2001 Winter Simulation Conference.*

Ólafsson, S. and L. Shi. 2000. "A Method for Scheduling in Parallel Manufacturing Systems with Flexible Resources," *IIE Transactions on Scheduling and Logistics*, 32: 135-146.

Rinott, Y. 1978. "On Two-Stage Selection Procedures and Related Probability-in-Equalities," *Communications in Statistics*, A7, 799-811.

Shi, L. and S. Ólafsson. 1997. "An Integrated Framework for Deterministic and Stochastic Optimization," in *Proceedings of the 1997 Winter Simulation Conference*, 358-365.

Shi, L. and S. Ólafsson. 1998. "Nested Partitions Method for Stochastic Optimization," Working Paper Series, 98-116, Department of Industrial and Manufacturing Systems Engineering, Iowa State University.

Shi, L. and S. Ólafsson. 2000. "Nested Partitions Method for Global Optimization," *Operations Research*, 48(3).

AUTHOR BIOGRAPHIES

JUMI KIM is a Ph.D. candidate in the Department of Industrial and Manufacturing Systems Engineering at Iowa State University. She received a B.S. in Industrial Engineering from the Kangnung National University of Korea, and a M.S. in Industrial Engineering from the Seoul National University of Korea. Her research interests are in simulation optimization.

SIGURDUR ÓLAFSSON is an assistant professor in the Department of Industrial and Manufacturing Systems Engineering at Iowa State University. He received a B.S. in Mathematics from the University of Iceland in 1995, and an M.S. and a Ph.D. in Industrial Engineering from the University of Wisconsin - Madison in 1996 and 1998, respectively. His research interests include applied probability, stochastic optimization, and simulation. He is a member of IIE and INFORMS.