Proceedings of the 2005 Winter Simulation Conference
M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, eds.

A NEW OPTIMIZATION HEURISTIC FOR CONTINUOUS AND INTEGER DECISIONS WITH CONSTRAINTS IN SIMULATION

Mufit Ozden

Department of Computer Science and Systems Analysis
Miami University
Oxford, OH 45056 U.S.A.

ABSTRACT

In this paper, a new metaheuristic optimization approach is developed for the mixed integer decisions with constraints within a simulation model. Each decision variable is handled by an optimizer that uses a machine learning technique. At the beginning of each iteration, the decisions are selected randomly from their decision distributions. The performance evaluation is estimated during a short simulation run. The optimizers modify their selection-distributions for the decisions that prove to be "good" performance judged against an advancing threshold value. Then, a new set of decisions is generated for the next run. When the average performance reaches a good competency, the threshold value is advanced to a higher level. Thus, the optimizers are forced to learn toward the optimal solution. In this paper, after brief explanation of the approach, we present an application to a challenging engineering problem dealing with pressure-vessel design.

1 INTRODUCTION

We can often consider a simulation model as a complex random system in which a number decision variables or system parameters are to be set to their "optimal" values after obtaining the performance measures through a series of simulation runs. In simulation models, the optimization of the decisions is usually based on a metaheuristic using a search technique with a machine learning capability, such as Evolutionary Algorithms (Michalewicz, 1994), Simulation Annealing (Gutjahr and Pflug, 1996), and Tabu Search (Glover, et al., 1996).

In this paper, we propose a new machine learning technique called "optimizers with learning distributions" as the decision-making/learning components of simulation models. These optimizers employ an adaptive probabilistic search in the continuous domain guided by the performance evaluations and they learn the pattern of good decisions in their selection distributions. This learning technique is inspired from of the learning automata theory (LA) (Narendra and Thathachar, 1989). In formal mathematical terms, the decision making problem in simulated environments can be stated as follows:

$$\begin{aligned} &\min J(\theta)\,, \\ &g_i(\theta) \le 0, \quad i = 1,\ldots,m \\ &L \le \theta \le U \\ &\text{where } J(\theta) = E[\,P(\theta,\omega)] \end{aligned} \tag{1}$$

and P is a real-valued performance function, which depends on the decisions θ and a random variable ω, and $J(\theta)$ is the expected value of this performance function. Note that an equality constraint can be represented by two inequality constraints equivalently. Hence, the model (1) is general.

Typically, the random performance function in (1) is not known explicitly and can only be estimated with running a simulation program. Formally, the optimizer learning technique can be characterized with the triple $\{\theta, \beta, A\}$ for designs θ, performance estimation β, and a learning algorithm A. The search for the optimal decisions is conducted with the decision selection distribution P defined on the decision set θ. Initially, the distribution is usually constructed as a uniform distribution giving equal probability to every decision value. When an optimizer selects a decision θ_i randomly and applies it to a random simulation model with all the other optimizers, it receives a (normalized) performance estimation β_i from the model after a short run. On the basis of this estimation, the automaton modifies its decision selection distribution according to the learning algorithm A. We adapted a particular learning algorithm known as the "reward and inaction" (a $L_{R\text{-}I}$ automaton in the nomenclature of the LA theory) in conjunction with a new moving threshold level to simulation models.

2 GLOBAL OPTIMIZATION OF OBJECTIVE FUNCTION

First, we will consider the objective function and the boundary box constraints on the variables. In our implementation of optimizers to the continuous decision domains, every continuous decision component is first scaled over [0,1) interval and each digit after the decimal-point is represented by a discrete optimizer that has a selection distribution defined on 10 discrete values (0,1,…,9) to the desired level of accuracy. Hence, several optimizers represent a continuous decision component. The simulation model provides a binary feedback (success or failure) for the randomly selected decisions after comparing to a current threshold value. When found successful, the optimizers modify their respective selection-distributions as follows: At the simulation run n, if β_i is 1,

$$p_i(n+1)=p_i(n)+a[1-p_i(n)]$$

$$p_j(n+1) = (1-a)\,p_j(n) \qquad \forall j \text{ except } i$$

$$\text{otherwise,} \quad p_j(n+1) = p_j(n)\,, \qquad \forall j \tag{2}$$

where a is small positive learning constant and $P(n)$ is a distribution vector whose i[th] component is the selection-distribution $p_i(n)$ for the optimizer i. For the integer variables, there is a single selection-distribution defined over the discrete values of the decision. Formally, the process $\{P(n) : n \geq 0\}$ describes a Markov process. The long-term convergence properties of the learning algorithm can be derived using this Markov property. The development and implementations to the simulation models of various forms of this machine learning technique can be found in (Ozden, 1994; and Ozden and Ho 2003) for the discrete decision domain in (Ozden, 2005) for the continuous domain.

Considering a minimization problem, we define a decision set θ as successful if

$$J(\theta) < T\,, \quad \forall \theta \in \Theta \tag{3}$$

The estimation of performance for a selected decision set at the end of a run may deviate from the expected value due to the random nature of a simulation model. Therefore, sometimes it is possible to reward a mediocre decision set erroneously. However, the probabilistic nature of the decision-making is expected to correct these mistakes in the long run. Hence, if the estimated performance of a decision set θ_i is less than the threshold value T, the value β is set to 1; otherwise to 0. Then, the optimizers update the selection distributions according to the formula (2). If the optimizers achieve a certain rate of success, with the current threshold value after a fixed number of runs, the threshold value itself is lowered to a new level as follows:

$$T \leftarrow \gamma T, \text{ where } 0 < \gamma < 1. \tag{4}$$

When the optimizers meet a certain convergence criterion, the search is terminated. For some simulation problems, it is possible to continue until some decision set achieves very high probability close to 1 as the optimal solution. However, this may require many expensive simulation runs and therefore it may not be feasible for large simulation problems. In this case, a less stringent convergence criterion may be adopted. For example, the probability of any decision set reaches a high level, such as 0.90. Alternately, the termination criteria may be based on the success rate of the decisions or the quality of the best solution discovered.

3 HANDLING CONSTRAINTS

There are few methods to deal with the constraints in (1) implicitly. The metaheuristics (Coello, 2002) frequently use these methods which are usually based on some form of the penalty and augmented lagrangean methods that incorporate the constraints in the objective function. Here, we will introduce another method that takes the advantage of the adaptive nature of our method in dealing with the objective function and the constraints. The method will promote feasible decision making gradually by advancing the frontiers of the inequality constraints toward the actual values along with the decreasing threshold value for the objective value as the approach learns the behavior of the problem at hand with experimentation.

In the beginning of our approach, the right-hand sides values (RHS) of the constraints are computed as the maximum or the average values using random sampling of the variables for a few times using the box-constraints on the decision variables. Whether the maximum or the average values are used actually depends how restrictive the boundary constraints are. In any case, these relaxed values b_i's of the RHS are used for the inequality constraints until the threshold value of the objective value is to be advanced with the improved performance as explained above. Along with the threshold value, the current RHS of one of constraints is reduced toward zero in order according to the following formula: for the constraint i,

$$b_{i,new} = r(\gamma)\, b_i \tag{5}$$

where $r(\gamma)$ is a reduction coefficient which is a function of the threshold reduction coefficient and is greater than 0 and less than 1.

4 LOCAL OPTIMIZATION OF CONTINUOUS VARIABLES

When the termination criterion is met by the global optimization, the best feasible solution is used to build the restricted solution space around it by the local optimizer (LO). The LO also carries out a random search in this space that redefined after a certain probabilistic convergence. More specifically, for each variable there are three possible values with some selection probabilities. Initially, one of these values is the best value obtained in the global search, say x_i' for the variable θ_i and the other two values are defined around it as ($x_i' \pm \alpha x_i'$) where α is the step size as a percentage which is usually less than 0.1. Initially, all three points get equal selection probability of 1/3. The local search progresses in the same fashion as the global search by gradually increasing the probabilities of a feasible solution that beats the current threshold value that is itself advanced intermittently. When any local point reaches an aspiration probability level, say 0.9, then the search subspace is redefined again around this point with the equal probabilities. The step size α is also reduced intermittently when no progress is achieved for an extended period of time. The final result is produced when a termination criterion is reached in terms of number of iterations or the magnitude of the step size, or no progress for an extended period of time.

5 APPLICATION TO AN ENGINEERING PROBLEM

We have applied our probabilistic optimization approach to a number of random benchmark problems, but here we will go into details of one challenging problem that have also been used by many other researcher in the literature. The problem is an engineering problem of designing a pressure-vessel with a cylindrical shape and a hemispherical end. The performance reports of the other metaheuristics are easily available for this same problem, (Coello, 2002). Here, we will apply approach to the deterministic as well as probabilistic version of this problem. The deterministic version of the problem can be stated as follows:

$$\begin{aligned}
&\text{Pressure-vessel Problem:}\\
&Min\, f(X) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 +\\
&\qquad\qquad 3.1661x_1^2x_4 + 19.84x_1^2x_3\\
&st:\quad g_1(X) = -x_1 + 0.0193x_3 \le 0\\
&\qquad g_2(X) = -x_2 + 0.00954x_3 \le 0\\
&\qquad g_3(X) = -\pi x_3^2x_4 - 4/3\pi x_3^3 + 1296000 \le 0\\
&\qquad g_4(X) = x_4 - 240 \le 0
\end{aligned}$$

As with the other papers, the boundary conditions used on the variables in the application are $0 \le x_1 \le 99$, $0 \le x_2 \le 99$, $10 \le x_3 \le 200$, and $10 \le x_4 \le 200$. The variables x_1 and x_2 take on values only in the multiples of 0.0625, which is the thickness of a steel plate used. That is, they are integer variables. The variables x_3 and x_4 are continuous decisions. The probabilistic version of the problem is constructed with a random objective function as follows:

$$\hat{f}(X) = (1 + r\,w)\,f(X) \qquad (6)$$

where r is a uniformly distributed variable over the interval [-1, 1] ans w is a weight shich is set to 0.1 in our application.

In application of the approach of this paper to the deterministic pressure-vessel problem, we used a learning coefficient value of 0.0005 in (2), 0.85 for the parameter γ in (4), and $r(\gamma) = \gamma/1.75$ in (5). The objective function of the pressure-vessel was evaluated 100,000 times before the algorithm was terminated. In application of the approach to the probabilistic pressure-vessel problem with w=0.1 in (6), we used a learning coefficient value of 0.00001 in (2), with the same values of other parameters as in the deterministic case. The objective function of the pressure-vessel was evaluated 200,000 times before the algorithm was terminated. In both cases, the problem was solved 10 times to obtain the performance statistics. The results are displayed in Table 1.

First, the problem was solved as a completely continuous problem. The rows 10 and 11 in Table 1
Show that the results were better than all previous solutions in the literature. Some of the approaches solved the problem as a continuous problem and rounded the first two variables to the closest integer values. Here, the problem was also solved as a mixed integer programming problem. The rows 12 and 13 display the results for the deterministic and random version of the problem, respectively. The results of row 12 are more appropriate to compared to the results of the other approaches in literature. As it can be observed in the table, the performance of the approach of this paper turned out to be superior to the other approaches in all respects.

For the random version of the pressure-vessel problem, the result was also good with respect to the other techniques considering that 10% of the performance evaluation might be in error at every objective function evaluation; and especially, the low standard deviation is noteworthy to mention here. This version of the problem is actually the most significant for the simulation problems since we are proposing to use our approach in the random simulation environment for short batch-runs.

6 CONCLUSION

The probabilistic learning approach of this paper found very good solution for the pressure-vessel problem. For the future research, we will present application to a variety of the benchmark mathematical problems as well simulation models where the number of variables and constraints of different dimensions.

Table 1.1 Performance of a number of different Approaches to the Problem

Approach	Best	Mean	Worst	S. D.
1) Coello Evol. M. Obj	6069.3	6263.8	6403.5	97.9
2) Deb GenAS	6410.4	NA	NA	NA
3) Kannan Aug. Lagrange	7198.0	NA	NA	NA
4) Sandgren Branch&Boun	8129.1	NA	NA	NA
5) Coello CoEvol Pen.	6288.7	6293.8	6308.1	7.4
6) Homaifar GA stat. Pen	6110.8	6656.3	7242.2	320.8
7) Joines GA dyn. Pen.	6213.7	6691.6	7445.7	322.8
8)Michalewicz Annealing	6127.4	6660.9	7380.5	330.8
9) Bean Dual GA	6110.8	6689.6	7411.3	330.5
10)Ozden (con.; ss =10)	5891.8	6021.4	6222.2	114.4
11)Ozden(con. w=0.1;ss= 10)	5910.7	6015.5	6315.1	112.9
12) Ozden(mix ; ss=10)	6067.0	6185.2	6259.6	60.5
13)Ozden(mix. ;w=0.1; ss=10)	6134.7	6308.9	6422.3	91.4

REFERENCES

Bean, J.C. and A. B. Hadj-Alouane. 1992. A dual genetic algorithm for bounded integer programs, Technical Report TR 92-53, Dept. of Industrial and Operations Engineering, The University of Michigan.

Coello, C.A.C. 2000. Constraint handling using an evolutionary multiobjective optimization technique. *Civil Engr. Environ. Syst.* 17: 319-346.

Coello, C.A.C. 2000. Use of a self-adaptivepenalty approach for engineering optimization problems. *Comput. Ind*. 41(2):113-127.

Coello, C.A.C. 2002. Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. Computer Methods in Applied Mechanics and Engineering, Elsevier, (191): 1245-1287.

Deb, K. 1997. GeneAS: A robust optimal design technique for mechanical component design. In: D. Dasgupta and Z Michalewicz (eds). 1997. *Evolutionary algorithms in Engineering Applications*. Stringer, Berlin.

Glover, F., Kelly, J. P., Laguna, M. 1996. New advances and applications of combining simulation and optimization. *Proceedings of Winter Simulation Conference*.

Gutjahr,W.J. and G.Ch Pflug. 1996. "Simulated annealing for noisy cost functions." *Journal of Global Optimization* 8.

Homaifar, A., S.H.Y. Lai, and X. Qi. 1994. Constrained optimization via genetic algorithms. *Simulation*. 62(4): 242-254.

Joines, J. and C. Houck. 1994. On the use of nonstationary penalty functions to solve nonlinear optimization problems with GA. In*: Proceedings of the First IEEE Conference on Evolutionary Computation*, ed. D. Fogel, 1994. IEEE press, Orlando: 579-584.

Kannan, B.K. and S.N. Kramer. 1994. An augmented lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design*, J. Mech. Des. Trans*. ASME (116): 318-320.

Michalewicz, Z. and N.F. Attia. 1994. Evolutionary optimization of constrained problems. In*: Proceedings of the 3rd Annual Conference on Evolutionary Programming*. 1994. World Scientific, Singapore: 98-108.

Michalewicz, Z. 1994. Genetic Algorithms + Data Structures = Evolution Programs, 2nd Ed, Springer-Verlag.

Narendra, K., Thathachar, M. A. L. 1989. Learning automata an Introduction: Prentice Hall.

Ozden, M. 1994. Intelligent objects in simulation. *ORSA Journal on Computing* 6(4): 329-342.

Ozden, M. and Y-C. Ho. 2003. A Probabilistic Solution Generator for Simulation. *European Journal Operational Research,* 146(1): 35-51.

Sandgren, E. 1988. Nonlinear integer and discrete programming in mechanical design. In: *Proceedings of the ASME Design Technology Conference*. 95-105.