A Fast Multi-Objective Evolutionary Algorithm for Expensive Simulation Optimization Problems

Shin-Ming Guo
National Kaohsiung First University of Science and Technology
smguo@ccms.nkfust.edu.tw

Abstract

This paper describes a multi-objective evolutionary algorithm which targets primarily on "expensive" simulation-based optimization problems. The idea is to approximate the Pareto optimal front using Response Surface Methodology and screen out less promising offspring solutions before they are evaluated via simulation runs. Numerical examples suggest that the algorithm can save computational efforts without degrading the quality of final solutions.

1. Background

Optimization problems often have conflicting objectives such as inventory cost versus customer service in designing a supply chain. Various algorithms are available for solving multi-objective optimization problems. However, few algorithms deal with problems in which the objective functions must be evaluated via simulation runs. Optimization via simulation is challenging because it requires balancing the tradeoff between the simulation time used in estimating the objective function and the computational effort used for searching new solutions. If too much effort is spent on long simulation runs, the algorithm may evaluate only a few solutions in the time available and may fail to identify an optimal solution, or even a good solution. On the other hand, if we cannot control variability in the simulation experiment, the search can be misled or fail to recognize good solutions when they are encountered.

2. Literature Review

Evolutionary Algorithms (EA) are widely applied to solve multi-objective optimization problems. These algorithms initialize a set of solutions randomly and then evolve from generation to generation by operations of selection, crossover, and mutation. They mimic nature's evolutionary principles to move toward optimal solutions. Using a population of solutions simultaneously, instead of a single solution, gives EA an advantage for its use in solving multi-objective optimization problem. In terms of solution quality, NSGA-II proposed by [4] appears to be the benchmark of multi-objective evolutionary algorithms (MOEAs).

Some researchers focus more on the speed of the algorithms. Salami and Hendtlass [11] suggest that EAs are popular and robust for solving optimization problems but may require huge computation power for solving real problems. They introduce a fast evolutionary algorithm that does not evaluate all new solutions, thus operating faster. Each new solution is assigned a reliability value and is only evaluated using the true fitness function if the reliability value is below a threshold. Knowles [8] concerns multi-objective optimization where each solution evaluation is financially and/or temporally expensive. His ParEGO algorithm uses a design-of-experiments inspired initialization procedure and learns a Gaussian processes model of the search landscape, which is updated after every function evaluation. Test results suggest that ParEGO outperforms NSGA-II after just 100 and 250 function evaluations.

Another important issue is how to maintain both diversity and proximity during the evolutionary process. Bosman and Thierens [2] point out that an MOEA must approximate the Pareto optimal frontier such that solutions in the approximation set are close to Pareto optimal solutions and are as diverse as possible. They suggest the use of elitism to achieve exploitation of proximity, exploitation of diversity, exploration of proximity, and exploration of diversity. Prudius and Andradóttir [9] develop an algorithm which conducts global search for promising solutions within the entire feasible region (exploration) and local search of promising regions (exploitation). The algorithm switches between local search and global search based on most recent search results.

0-7695-2882-1/07 $25.00 ©2007 IEEE

Simulation is a very popular tool for performance evaluation due to its great modeling flexibility. In terms of optimization, the traditional viewpoint is that simulation provides no concrete solutions to problems, but only possible ones. In recent years, simulation researchers began to apply deterministic optimization procedures to search for a combination of inputs to a simulation program that will improve the system performance. An optimum-seeking tool called OptQuest based on tabu search and scatter search has been integrated into simulation commercial software.

However, marrying simulation and optimization algorithms is a challenging task. In addition to huge computational effort, simulation errors also hinder the effectiveness of the search process [13]. Boesel *et al.* [1] warn that commercial simulation optimization software is racing ahead of the supporting theory. They claim that commercial applications employing deterministic optimization procedures fail to account for randomness in the simulation results, meaning that their progress may be no better than a random search if the variability is high.

One way to overcome random errors and computational inefficiency is to develop effective procedures for simulation optimization. Pichitlamken and Nelson [9] construct an algorithm featuring a global guidance system, selection of the best, and local improvement. The global guidance system ensures that the search not only advances toward optimal solutions, but it also reaches one of them, if there is enough simulation effort. Another idea to improve the optimization process is to carefully select solutions to simulate in order to save computation time and/or maximize the improvement. One can use auxiliary information to perform such selection. Studies combining EAs with Response surface methodology (RSM) has received increasing attention in the literature. See [8], [11] and [12]. This is also the main direction of this research.

3. Research Design

Key assumptions on the optimization problem are:

(1) The problem has multiple objectives. Some objectives conflict with others.

(2) Each objective must be evaluated via simulation. Simulation errors can be controlled to a certain level.

(3) The dimensionality of the decision space is low.

(4) Total number of evaluations to be performed is limited by financial, time or resource constraints.

The proposed algorithm is an evolutionary algorithm which applies knowledge about the shape of the Pareto optimal front to avoid simulating inferior solutions and thus save computational effort. We first describe major steps of the algorithm, followed by discussion of key features that fine-tune the algorithm.

Step 0. Construct the initial population. Simulate all systems in the initial population.
Step 1. Use ranking and selection or multiple comparison procedures to identify the non-dominated solution set S_0. Set i=1.
Step 2. Estimate the response surface of the initial Pareto optimal front, denoted by R_0.
Step 3. At iteration i, construct high quality reference sets and diversity reference sets from the current population based on performance measures.
Step 4. Crossover members from different reference sets to create new system configurations.
Step 5. Screen out unpromising systems based on the distance to the estimated Pareto optimal front R_{i-1}.
Step 6. Simulate all system configurations in the remaining offspring population.
Step 7. Identify new non-dominated solutions and update the non-dominance set as S_i.
Step 8. Estimate the response surface of the updated Pareto optimal front R_i.
Step 9. Stop if the current result meets specified criteria. Otherwise, let $i \leftarrow i+1$ and go to step 3.

First of all, since the nature of the problem does not favor a large population size or a large number of generations, we propose to use Latin-Hypercube sampling to select an initial population in order to get a better covering of the decision space. Secondly, we construct a diversity set consisting of solutions that are farthest from the estimated Pareto optimal front. Solutions from this set will be mated with solutions from high quality reference sets to create a diverse offspring population. Two types of crossover operations based on scatter search [5] are used in order to explore the entire feasible region (global search) or search promising neighborhood (local search). Let $(x_1, x_2 \ldots, x_m)$ and $(y_1, y_2 \ldots, y_m)$ denote configurations selected from different high-quality sets. The first type of crossover operation chooses a random number w from [-1, 2] and create the offspring as

$$w\cdot(x_1, x_2 \ldots, x_m)+(1-w)\cdot(y_1, y_2 \ldots, y_m) \qquad (1)$$

The second type of operation uses three reference points, one of them chosen from the diversity set, as shown by (2) and figure 1,

$$w_2[w_1\vec{x} + (1-w_1)\vec{y} + (1-w_2)\vec{z} \qquad (2)$$

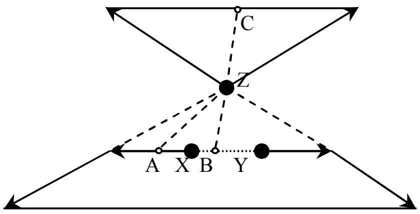

Figure 1: Region spanned by three reference points

The resulting offspring solution maybe modified or truncated if any of the parameters is out of the feasible range. It maybe abandoned or recreated if it cannot meet the problem constraints. Test results show that combining two points from high quality sets and one point from the diversity set has a good chance of producing new non-dominated solutions.

The most important feature of the algorithm is to screen out unpromising offspring solutions. We propose that solutions with small crowding distances or with large Euclidean distances from the current Pareto optimal front should be screened out to save simulation time. This approach requires knowledge about the shape of the true Pareto optimal front, which seems to be contradictory. Our approach is to apply RSM to model the landscape based on non-dominated solutions after five or more generations. This is similar to the concept of warming up for steady state simulation. The accuracy of the estimated Pareto set is less critical because we are not using it to search for solutions that maximize the "expected improvement". The shape of the Pareto set is updated after each generation using the new non-dominated set.

After simulation runs, the algorithm updates the non-dominated set and begins the next iteration if necessary. Comparing solutions may not be efficient if simulation errors are significant. We use the crowding distance metric which calculates the distance to the nearest solution in the current non-dominance set. We prefer the solution with the largest crowding distance because it is located in a lesser crowded region.

4. Experimental Results

Due to the limitation of space, we show only two examples in this section. The first test problem is the Minimization Example Problem (Min-Ex) used by [3]:

$$Min-Ex: \begin{cases} Minimize \ f_1(x) = x_1, \\ Minimize \ f_2(x) = \dfrac{1+x_2}{x_1}, \\ subject\ to \ \ 0.1 \le x_1 \le 1, \text{and} \\ \qquad 0 \le x_2 \le 5. \end{cases} \tag{3}$$

The Pareto-optimal front is known to be $0.1 \le x_1 \le 0$ and $x_2=0$. The algorithm is executed for 10 replications. Each replication applies LHS to select an initial population and runs for 20 generations. The population size is fixed at 16. We first conduct the experiment with all 16 offspring solutions evaluated. Then we repeat the experiment with the condition that only 10 "promising" offspring solutions are chosen for evaluation. Figure 2 displays the estimated Pareto set after 20 generations in one replication.

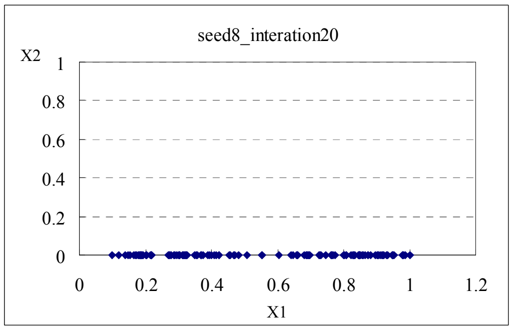

Figure 2: Estimated Pareto set of (3).

Table 1 lists replication averages and standard deviations of convergence metric, diversity metric, and recall rate. We find that screening out unattractive solutions does not affect the performance significantly.

Table 1: The performance of the algorithm for Min-Ex.

	No screening	Screening
Convergence	0.199	0.195
std deviation	0.333	0.306
Diversity	0.937	0.938
std deviation	0.123	0.132
Recall rate	0.507	0.471
std deviation	0.339	0.327

The second problem is an option replenishment inventory simulation model. The inventory is reviewed periodically but an order is placed only when the inventory drops below the reorder point. The objective is to determine values of reorder point, review period, and safety stock such that total daily cost is minimized and the fill rate is maximized. All simulation errors are within 1% of the performance measures. The algorithm keeps solutions with no significant difference. Figure 3 displays one estimated Pareto set after 10 generations. The curve is the "true" Pareto set estimated using a brute-force approach. Table 2 also suggests that screening out unattractive offspring solutions does not degrade the algorithm significantly.

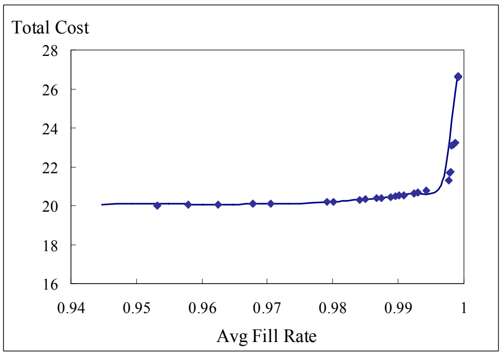

Figure 3: Pareto set of inventory simulation problem.

Table 2: The performance of the algorithm.

	No screening	Screening
Convergence std deviation	15.99 5.72	15.69 6.43
Diversity std deviation	0.773 0.137	0.764 0.128

5. Summary

Evolutionary algorithms have become a popular tool for solving multi-objective optimization problems. If the objective functions must be evaluated via simulation, then it maybe too expensive to perform thousands of simulation runs using regular algorithms. Therefore, this research develops a multi-objective evolutionary algorithm which does not evaluate each offspring solution but selects only promising ones for simulation evaluation. Numerical results show no degradation in the performance of the algorithm while saving nearly 30% of simulation runs.

Future research will include testing large scale problems and the application of Kriging to better estimate the landscape of the Pareto optimal front. Kriging is an interpolation technique originally developed to determine true ore-grades based on samples and does not require formal experimental designs. Some researchers [7], [14] have applied Kriging to simulations and report that Kriging is good technique for prediction and worth further exploration.

Acknowledgement: This study is supported by National Science Council under grant NSC 95-2221-E327-016.

6. References

[1] Boesel, J., Nelson B. L., and Ishii, N. (2003), A Framework for Simulation Optimization Software, *IIE Transaction*, 35:221-229.

[2] Bosman, P. A. N. and Thierens, D.(2003), The Balance Between Proximity and Diversity in Multi-objective Evolutionary Algorithms, *IEEE Transactions on Evolutionary Computation*, Vol. 7, no. 2, pp.174

[3] Deb, K. (2001). *Multi-Objective Optimization Using Evolutionary Algorithms*, John Wiley & Sons, England.

[4] Deb, K., Pratap, A., Agarwal S., and Meyarivan, T.(2002), A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II, *IEEE Transactions on Evolutionary Computation*, Vol. 6, no. 2, pp.182-197.

[5] Glover, F., Laguna, M., and Marti R. (2000), Fundamentals of Scatter Search and path Relinking, *Control and Cybernetics*, 29, no.3, 653-684.

[6] Joines, J.A., Gupta,D. Gokce, M.A., King, R.E. and Kay, M.G. (2002). Supply Chain Multiobjective Simulation Optimization, *Proceeding of the 2002 Winter Simulation Conference*, E. Yücesan, C.-H. Chen, J.L., Snowdon, and J.M. Charnes, eds, pp. 1306-1314.

[7] Jones, D.R., Schonlau, M., and Welch, W.J. (1998), Efficient Global Optimization of Expensive Black-Box Functions, *Journal of Global Optimization* 13: 455–492.

[8] Knowles, J.(2005), ParEGO: A Hybrid Algorithm with On-Line Landscape Approximation for Expensive Multiobjective Optimization Problems, *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 1

[9] Pichitlamken, J. and Nelson, B. L. (2002), A Combined Procedure for Optimization via Simulation. *Proceeding of the 2002 Winter Simulation Conference*, E. Yücesan, C.-H. Chen, J.L., Snowdon, and J.M. Charnes, eds, pp. 292-300.

[10] Prudius, Andrei A. and Andradóttir, S.(2004), Simulation Optimization Using Balanced Explorative and Exploitative Search, *Proceedings of the 2004 Winter Simulation Conference*, R. G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, eds., p.545

[11]. Salami, M. and Hendtlass, T. (2003), A Fast Evaluation Strategy for Evolutionary Algorithms, *Applied Soft Computing* 2/3F, 156–173.

[12] Shang, J. S., Li, S. and Tadikamalla, P. (2004), Operational Design of a Supply Chain System Using the Taguchi Method, Response Surface Methodology, Simulation, and Optimization, *International Journal of Production Research*, vol. 42, no. 18, 3823–3849

[13] Tekin, E., and Sabuncuoglu, I. (2004), Simulation Optimization: A Comprehensive Review on Theory and Applications. *IIE Transactions*, 36, pp.1067-1081.

[14] van Beers, W.C.M. and Kleijnen, J.P.C.(2003), Kriging for interpolation in random simulation, *Journal of the Operational Research Society*, 54, 255–262.