On How Solution Populations Can Guide Revision of Model Parameters

Steven Orla Kimbrough
University of Pennsylvania
Operations and Information Management
Philadelphia, PA USA
kimbrough@wharton.upenn.edu

David Harlan Wood
University of Delaware
Computer and Information Science
Newark, DE USA
wood@cis.udel.edu

ABSTRACT

Post-evaluation analysis of the model of a constrained optimization problem is conducted after obtaining preliminary optimal or heuristically good solutions. The primary goal of post-evaluation analysis is to reconsider assumptions made in the model in the light of information generated while finding the good solutions as well as information not previously detailed in the model. We seek extensions of the techniques presently available for the special case of linear programming problems because these special problems allow excellent post-evaluation analysis as a side-effect of seeking solutions. Unfortunately, more general problem solvers presently provide little if any information for post-evaluation analysis.

We consider general metaheuristic methods that evolve populations of settings of the decision variables. These methods can contribute greatly to reconsideration of modeling assumptions. This is because the evolving populations taken in total provide a great number of samples for conducting post-evaluation analysis in a data-driven fashion. This is a very general claim. It is illustrated in this paper by a single, rather simple, constrained optimization problem.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search—*heuristic methods*

General Terms

evolutionary computation, metaheuristics, parametric analysis, post-evaluation analysis

1. INTRODUCTION

In this note we make and expand upon five observations pertaining to the use of metaheuristics on constrained optimization models. In brief, the observations are as follows.

1. Constrained optimization models are used extensively in practice, where it is generally acknowledged that obtaining preliminary solutions typically precedes the bulk of the modeling of the decision problem at hand. The remaining modeling work is standardly called *post-evaluation* analysis. It largely focuses on *parametric analysis,* the study of the effect of systematic changes to the model's parameters.
2. There exists a need to find and develop concepts to support post-evaluation analysis for general constrained optimization models. We are inspired by the special case of linear programming models, where post-evaluation analysis is greatly facilitated by the simplex algorithm. The simplex algorithm incidentally provides valuable information for post-evaluation analysis. This type of information is generally very limited for other kinds of constrained optimization problems.
3. The decision space of a constrained optimization model consists of variable settings, objective function value, and feasibility criteria. If the decision space could be adequately sampled, the samples could be used to obtain estimates for the key questions of parametric analysis. The decision spaces, even for routine constrained optimization models, are too large for unbiased random sampling to be useful for purposes of parametric analysis. Sampling, to be effective, must be usefully biased.
4. The boundary separating feasible and infeasible solutions is typically a highly relevant part of the decision space. Generally, it will be desirable to bias samples toward the boundary of the feasible region, which is determined by the constraints on the problem. In particular, it will be desirable to sample the infeasible solutions, as well as the feasible.
5. The Feasible-Infeasible Two-Population Genetic Algorithm (FI2PopGA) [3, 4, 5, 6, 7] retains infeasible solutions whose evolutionary selection is based on fitness measured as distance from feasibility. Thus, the FI2PopGA is a method with a credibly biased sampling mechanism advantageous to post-evaluation analysis. Other population-based metaheuristics may also be useful for this purpose. What approach might be best is entirely an open question.

2. AN ILLUSTRATIVE PROBLEM

In expanding upon the five observations of the previous section we will, for the sake of illustration, demonstrate the use of the FI2PopGA on a single knapsack (0-1 integer programming) problem. This will help to demonstrate concepts clearly. More extensive evaluation of concepts and algorithms must await future investigation.

The following class of problems is used for our illustration. Knapsack problems with a single constraint are a special case of constrained optimization problems. In words, such a problem is to select various objects that will fit into a given 'knapsack' so as to maximize their total value, subject to a constraint on their total 'weight.' The problem has the following form.

$$\max z = \sum_{i=0}^{n} p_i x_i \tag{1}$$

subject to the constraint

$$\sum_{i=0}^{n} w_i x_i \leq c \tag{2}$$

by selecting

$$x_i \in \{0,1\}, \quad i = 0,1,2,\ldots,n. \tag{3}$$

Our illustration is based on Knap101. A specific instance of this problem detailed in Appendix A.

3. CONSTRAINED OPTIMIZATION

By way of context and background, optimization problems may usefully be distinguished as either constrained or unconstrained. Our focus in this paper is on constrained optimization problems,[1] which have the following general form:

$$\max z = d(\vec{x}) \tag{4}$$

subject to

$$\begin{aligned} f_i(\vec{x}) &\leq a_i, \quad i = 1,2,\ldots,n_f & (5)\\ g_j(\vec{x}) &\geq b_j, \quad j = 1,2,\ldots,n_g & (6)\\ h_k(\vec{x}) &= c_k, \quad k = 1,2,\ldots,n_h & (7)\\ x_l &\in S, \quad l = 0,2,\ldots,n_l. & (8) \end{aligned}$$

$d(\vec{x})$ in expression (4) is called the *objective function* for the problem. Its value, z, is what we seek to maximize (or minimize) by finding values of, or settings for, the *decision variables*, the x_ls, that yield the highest (or lowest if minimizing) value for z among the settings that satisfy the *constraints,* namely the expressions (5)–(8). Such a setting of values for the decision variables is said to be optimal.

Any particular choice of the values for the decision variables is called a *solution* to the problem, regardless of whether it is optimal or whether it satisfies the constraints. A solution that satisfies all of the constraints is said to be *feasible,* otherwise it is *infeasible.* Optimal solutions must be feasible, but need not be unique; other feasible solutions may yield equally good values of z.

The constraints, as we have just noted, serve to classify solutions as either feasible or infeasible. The *right-hand side* (RHS) values of the inequality constraints, the a_is and the b_js, are said to define *boundaries* between the feasible and infeasible regions for the problem. A given solution, $\vec{x}$, is said to be *near to the boundary* (for a particular constraint) if the *left-hand side* of the constraint is close (pragmatically defined for the problem to hand) to the right-hand side. The solution is said to be *on* the boundary if the left-hand side equals the right-hand side. More generally, we say that a solution is on or near the boundary of the feasible region if it is on or near the boundary of at least one constraint.

In typical constrained optimization problems encountered in practice, the optimal solutions, as well as the good (near optimal) solutions, are on or near the boundary of the feasible region. (As a special example, linear programming problems always achieve their optimum on the boundary.)

4. POST-EVALUATION ANALYSIS

The term post-evaluation analysis refers to investigations for the purpose of decision making that happen *after* a model has been formulated and solved by an optimizing or heuristic evaluator. A presumably good, or even optimal, solution *to this model* is at hand. Call it z^+. Before actual decisions are taken, however, it is normally prudent to ask various kinds of *post-evaluation questions.* There are two main types of questions.

1. Sensitivity analysis questions: Do small changes in the model's parameters have large effects on either (a) the value realized for the objective function, or (b) the accepted solution?

2. Candle-lighting analysis questions: Are there advantageous opportunities to change the assumptions of the model? For example, would a change in the RHS of a constraint yield a significantly improved objective value? If so, does the cost of changing the RHS net out to a profit? Conversely, are there good solutions for which the left-hand side (LHS) value of a constraint is far from the RHS value? If so, can the slack RHS resource be profitably sold or used for some other purpose? See [2, 8, 9, 10, 11] for elaboration of the candle-lighting concept.

Both sorts of questions are quite important in practice. For present purposes it suffices to conflate them and to focus on post-evaluation analysis of the *parameters* of a constrained optimization model. This is called *parametric analysis* and we focus on it in what follows. Parameters fall into three categories, the RHS value or values (c in expression 2), objective function parameters (the p_i in expression 1), and the LHS parameters (the w_i in expression 2).[2] We shall discuss each in turn.

[1]The distinction is perhaps not absolute, since there are cases in which constraints may be eliminated by an alternative encoding of the problem. These special cases, however, need not distract us.

[2]Strictly speaking the concept of a parameter only applies for certain functional forms, e.g., linear and multiplicative. We say parameters because that will be correct for most models of practical import, but our remarks generally apply to parameters, whether or not they are parameters.

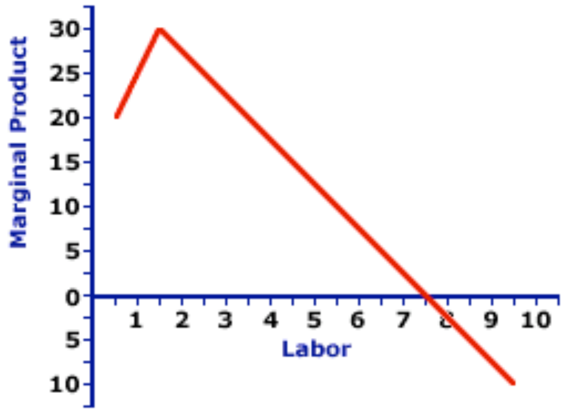

Figure 1: Marginal Product Curve

4.1 RHS Parametric Analysis

What economists call the marginal product curve is a staple of economic analysis.[3]

> MARGINAL PRODUCT CURVE:
>
> A curve that graphically illustrates the relation between marginal product and the quantity of the variable input, holding all other inputs fixed. This curve indicates the incremental change in output at each level of a variable input. [1]

An example of such a curve, also from [1], is shown in Figure 1.

A similar curve can be drawn for each constraint in a constrained optimization problem. Each such curve is called either *the optimal value function for the constraint* or *the objective function operating curve for the constraint*.

We return to our illustrative knapsack problem Knap101. We insert into a database table, `poplog`, *each and every* solution created by the evolutionary process of the FI2PopGA, along with its LHS value and its objective function value. Using these data, Figure 2 shows a plot of pairs of points (LHS value, objective value) with the knapsack's constraint RHS value, c, on the abscissa, and the corresponding optimal value z of the objective function on the ordinate.

In Figure 2 we see a frontier on one side of the plotted points. The frontier has a positive slope because as the RHS values get larger, larger objective function values are discovered. That is, increasing the knapsack capacity can only improve the value of its contents. Without loss of generality, we assume a maximization problem and a $\leq$ constraint. Thus, solutions to the left of an abscissa value are feasible for that constraint. If, as in the present case, the constraint RHS value is 200, then any plotted solution to the left of ($\leq$) 200 on the abscissa is feasible. Then, among the feasible points, those scoring highest on the ordinate axis are best. The black (dark) points in Figure 2 are just such an example.

Figure 2 also contains a second collection of points, roughly running along the frontier of the plot of sampled solutions. This second collection of points, shown in red (or a shade of gray in grayscale display), plots points (LHS value, objective value) obtained from a knapsack heuristic called "bang-per-buck," which is known to perform very well. See Toth and Martello [12]. (The bang-per-buck heuristic[4] only applies to knapsack problems with a single constraint.)

Again, we want to emphasize that our purpose is not to discuss effective heuristics for the knapsack problem (bang-per-buck is hard to beat). Instead, we wish to illustrate how population-based metaheuristics—the FI2PopGA is the present example—may be used for parametric analysis of optimization problems.

We now consider specific questions pertaining to RHS parametric post-evaluation analysis. As in expression (2), let the RHS parameter be labeled c. Perhaps the two most important examples of questions for parametric post-evaluation analysis of a constraint's RHS value are these:[5]

1. As knapsack capacity c *increases* how does this affect the results? At what point does the currently best solution, z^+ change and what is its new value? Generally, how does z^+ change as c increases and what are the associated solutions?

2. As knapsack capacity c *decreases* how does this affect the results? At what point does the currently best solution, z^+ change and what is its new value? Generally, how does z^+ change as c decreases and what are the associated solutions?

These questions in effect ask for the *objective function operating curve* as a function of c. (In linear programming, these questions pertain to what is called the *shadow price* of the RHS value.)

As we saw in Figure 2, the objective function operating curve can be *estimated* as the frontier of the solutions obtained during the execution of an evolutionary solver, in our particular case by FI2PopGA. More focused estimates may be obtained by querying the database of these solutions, as we now demonstrate.

In practice it will often be the case that opportunities of certain types prove profitable. For example, Table 1 shows the results of querying on a relaxation of knapsack capacity in

[3] Thanks to Jack Hershey for alerting us to an analog in economics of what we here are calling the optimal value function or the objective function operating curve for the constraint.

[4] After sorting the objective function-constraint parameter ratios p_i/w_i in descending order, finding the bang-per-buck solution for a given RHS value requires one pass through the decision variables. The knapsack items are considered in descending order of p_i/w_i. If adding item i to the knapsack does not violate the constraint, it is added; otherwise it is skipped and the next item is considered. So it is very fast indeed. This makes it ideal as a benchmark for other heuristics and in particular for the FI2PopGA.

[5] Again, without loss of generality, we are assuming a maximization problem and a $\leq$ constraint.

the Knap101 problem. The query asks for distinct objective function values for feasible solutions when $200 < c \leq 210$. Only the top 20 are listed. Even so, the results are quite intriguing. Recall that when $c = 200$, the optimal solution has an objective function value of 1119.984. As is easily read off from Table 1, if we permit c to be increased to 208.22, the FL2PopGA has already found a solution whose objective value is considerably larger, namely 1153.92. The decision maker will likely want to investigate whether it is possible to increase c and if so at what cost.

objval	lhsval
1153.92	208.022
1140.253	209.401
1136.569	207.438
1136.012	205.986
1135.114	205.092
1132.789	202.797
1131.042	205.001
1130.354	208.516
1127.296	205.859
1126.113	205.101
1125.384	207.531
1125.056	205.803
1124.619	209.369
1124.229	207.004
1124.225	201.669
1123.76	205.717
1123.364	205.512
1122.987	208.366
1122.212	206.138
1121.263	208.47

Table 1: All solution samples in the database selected by the SQL query "`select distinct objval, lhsval from poplog where lhsval > 200.0 and lhsval <= 210 order by objval desc limit 20;`"

On the other hand, Table 2 reports on reducing knapsack capacity to less than $c = 200$. Here it is notable that there is a known solution with objective value 1103.516 at $c = 192.628$. Now the decision maker may want to consider whether, say, 7 units of c are worth more than, say 17 units of z. This may well be the case. In any event, this information is valuable for reconsidering the model of the problem at hand.

4.2 Objective Function Analysis

As in expression (1), let the objective function parameters be labeled p_i. Perhaps the most important examples of questions for objective function post-evaluation analysis are these:

1. Given a solution corresponding to z^+ with $x_i = 0$, what is the best solution available for which $x_i > 0$ and what is its value, z? (In linear programming the difference $z^+ - z \geq 0$ is called the *reduced cost* for the decision variable x_i. Linear programming solvers compute reduced costs as a side-effect, but do not provide the associated solution, the setting of the decision variables. Nothing very similar is available from standard solvers for optimization problems that are not linear programs.)

objval	lhsval
1119.086	197.36
1115.014	197.269
1114.116	196.375
1109.028	198.071
1107.296	197.269
1106.859	195.149
1104.058	197.086
1103.516	192.628
1102.326	196.284
1100.437	195.871
1099.539	194.977
1099.478	198.036
1098.58	197.142
1097.955	192.135
1097.858	195.158
1097.397	196.384
1095.509	192.952
1095.467	194.886
1094.8	192.501
1094.569	193.992

Table 2: All solution samples in the database selected by the SQL query "`select distinct objval, lhsval from poplog where lhsval >= 185.0 and lhsval <= 198.253 order by objval desc limit 20;`"

Conversely, we can ask: Given a solution corresponding to z^+ with $x_i > 0$, what is the best solution available for which $x_i = 0$ and what is its value, z? (Linear programming solvers do not provide this sort of information as a side-effect from solving the original problem. The user can, of course, add a constraint and resolve the problem.)

2. How do z^+ and its corresponding solution change as p_i changes?

Table 3 illustrates how to answer questions of the first type by querying the database of solutions generated by an evolutionary solver. Note that in the optimal solution to Knap101 $x_0 = 0$. What about $x_0 = 1$? What is then the best feasible solution and what is its value? In Table 3 we see that the best solution in the database in which $x_0 = 1$ has an objective functional value of 1007.516, a large reduction from 1119.984.

The converse type 1 question, "What if we remove an item that is in the optimal solution?," is as easily handled. See Table 4. There we find that the best (discovered) feasible solution with $x_1 = 0$ has an objective value of 1092.472. We also see that there is a solution at 1116.049 if we allow the RHS to increase as far as 206.66.

We now consider an example of the second type of objective function question. The objective function parameter on x_{49} is 97.366. Suppose that its value falls to 80.0. What is the value now of the best solution, what is it and does it include x_{49}? Two database queries will answer the question. First, in Table 5, we consider solutions for which $x_{49} = 1$. These will have their objective function values reduced by 97.366

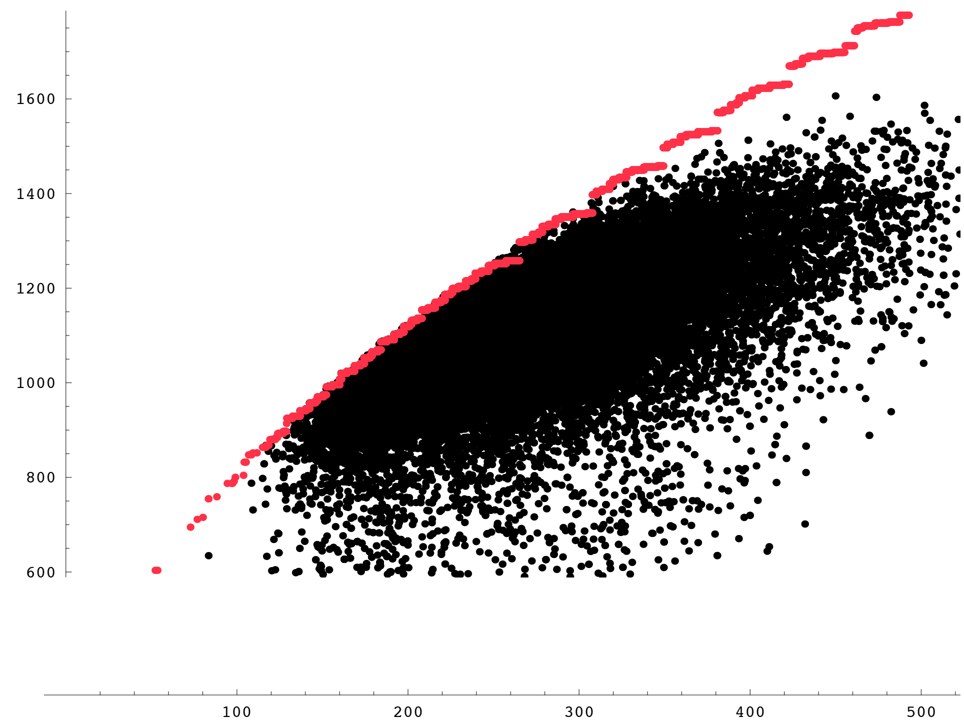

Figure 2: RHS = 200.00. FI2PopGA. knap101-20060430-60-60-500.pdf. The optimal solution, 1119.984, was found in generation 293. Mutation rate = 0.025. Crossover rate = 0.4.

- 80 = 17.366. The table reports the top solution values for solutions with $x_{49} = 1$.

Solutions with $x_{49} = 0$ will be unaffected by the objective function parameter change. Table 6 shows these. Comparison of the two tables indicates that the optimal (or at least the best so far discovered) solution, z^+, would not change.

4.3 Constraint Parameters Analysis

As in expression (2), let the constraint parameters be labeled w_i and assume that there is a decision variable x_i corresponding to each w_i. Perhaps the two most important examples of questions for parametric post-evaluation analysis of a model's constraints are these:

1. If w_i changes to w'_i, how does this affect the results?
2. How much can w_i change without changing the currently best solution? (In the case of optimality: How much can w_i change without changing what is the optimal solution?)

On the first question, suppose that $w'_i = w_i + \theta$. For a given constraint, we can think of its left-hand side value as a function of w_i: $LHS(w_i)$. What we're after is

$$\Delta LHS = LHS(w_i + \theta) - LHS(w_i) \tag{9}$$

Given ΔLHS we can subtract it from the RHS value of the constraint and hope to reduce the problem to the case of RHS analysis. The value of ΔLHS will, however, depend on particular solutions. Without knowing the functional form of the constraint, no general approach is available other than recalculating the constraint LHS values for each solution under consideration.

All this is much simpler in the special case of the (one-constraint) knapsack problem, since the constraint is linear with parameters w_i. See expression (2). Also, the decision variables are 0-1. In consequence, this case can be handled much as in the case of Table 5 and Table 6. First, query for solutions with $x_i = 1$ and the constraint's RHS reduced by θ. Then query for solutions with $x_i = 0$ (these are not affected by changes in w_i). Compare the highest feasible objective values from the two queries. If the $x_i = 0$ query has the best results, then adding θ to w_i changes (what appears to be) the best solution; and otherwise leaves it unchanged.

The second question is also easily handled in our special case. Consider $w_{49} = 4.992$. Compare Table 6 ($w_{49} = 0$) and Table 7 ($w_{49} = 1$). It is clear that so long as $\theta < 1.746$ the currently best solution keeps its title. If $1.746 < \theta < 2.639$, the the bang-per-buck solution is best (1119.086). Note that when $\theta = 11.689$ (or RHS = 200 - 11.689 = 188.311) we still have a feasible solution with $x_{49} = 1$ and objective value 1091.729. In Table 6, however, even with a RHS of 189.894 we only have an objective value of 998.101. It appears that x_{49}'s presence in good solutions is quite robust to changes in w_{49}. (We have only considered increases in the value of w_{49}. Decreases in its value can only enhance the value of x_{49} in a solution. Similarly, however, we may consider decreases in a w_i for an x_i that is 0 in good solutions. This will indicate how much w_i has to decrease in order to make x_i attractive in a good solution.)

objval	lhsval
1007.516	197.709
976.757	188.88
971.787	187.895
971.697	195.624
968.582	191.725
957.506	198.304
947.282	185.088
945.839	199.194
945.356	188.534
939.389	197.829
930.814	179.462
930.128	174.4
929.75	180.413
922.85	196.731
912.527	192.483
910.823	185.513
909.141	183.729
904.29	190.773
887.572	190.292
884.762	180.882

Table 3: The best solution found has $x_0 = 0$. What are the best feasible solutions in which $x_0 = 1$? All solution samples in the database selected by the SQL query "`select distinct objval, lhsval from poplog where x0=1 and feasibility = 1 order by objval desc limit 20;`"

objval	lhsval
1116.049	206.66
1104.142	203.011
1099.172	202.026
1094.243	202.126
1092.472	202.252
1092.355	198.694
1091.457	197.8
1090.567	209.89
1090.342	203.163
1090.2	208.294
1088.114	195.279
1087.877	206.208
1087.385	197.709
1087.216	194.385
1086.749	209.287
1083.697	199.734
1083.144	194.294
1079.667	197.709
1079.302	206.807
1075.426	194.294

Table 4: The best solution found has $x_1 = 1$. What are the best solutions in which $x_1 = 0$? All solution samples in the database selected by the SQL query "`select distinct objval, lhsval from poplog where x1=0 and lhsval <= 210.0 order by objval desc limit 20;`"

5. COMBINATIONS OF PARAMETERS

The data-driven approach illustrated in this paper is able to analyze joint changes of more than one type of parameter at a time. In fact we have illustrated this in Table 4, which presents results for both $x_1 = 0$ and for $c \leq 210.0$.

Methods for linear programming problems are not able to do this despite their other advantages for post-evaluation analysis. With existing methods for linear programming it is possible to analyze changes in RHS values, and it is possible to analyze reduced costs (for example), but it is not possible (without re-execution of the model) to examine both kinds of changes jointly.

6. CONCLUSIONS

Evolutionary solvers, and population metaheuristics generally, may contribute greatly to decision making involving constrained optimization problems. They may do this by finding excellent solutions at reasonable cost, as well as by providing valuable information for conducting post-evaluation analysis in a data-driven fashion. Our aim in this paper has been to support the post-evaluation claim by demonstration on a single, rather simple, constrained optimization problem. We expect these findings to be general.[6]

Very much remains to be investigated. We note in particular three points. First, the solutions found by any metaheuristic in processing a constrained optimization problem can only be a sample of the solution space, presumably with a helpful bias. It will be important to investigate how well different metaheuristics perform *with respect to post-evaluation analysis* on various types of constrained optimization problems. This would appear to be an unavoidably empirical matter, but one that potentially will pay great rewards in practice. Second, there will be scale problems (of a computational as well as cognitive sort) as the numbers of parameters, of decision variables, and especially of constraints rise. Discovering effective analytical tools, as well as computational approaches, for data-driven post-evaluation analysis must also be seen as potentially offering important practical rewards.

Our third point concerns the computational cost of using a population-based metaheuristic. Logging the discovered solutions and loading them into an indexed relational database (as we have done) imposes an additional cost that is expected to be relatively small. Assuming that the decision maker is interested in conducting serious post-evaluation analysis, we note that once the solutions are stored in an indexed database the cost of querying, per our examples (see Tables 1–7), can be very low indeed, especially when compared to the cost of re-executing a standard solver.

[6] We have noted complications when objective or constraint functions are not linear or otherwise simple in form.

7. REFERENCES

[1] AmosWEB LLC. Marginal product curve. `http://www.AmosWEB.com`, [Accessed: April 25, 2006]. AmosWEB Encyclonomic WEB*pedia, `http://www.amosweb.com/cgi-bin/awb_nav.pl?s=wpd&c=dsp&k=marginal+product+curve`.

[2] B. Branley, R. Fradin, S. O. Kimbrough, and T. Shafer. On heuristic mapping of decision surfaces for post-evaluation analysis. In J. Nunamaker, Jr. and R. H. Sprague, Jr., editors, *Proceedings of the*

objval - 17.366	lhsval
1102.618	198.254
1101.72	197.36
1101.32	199.977
1097.648	197.269
1096.75	196.375
1091.662	198.071
1089.93	197.269
1089.493	195.149
1087.312	199.286
1086.718	198.982
1086.692	197.086
1086.15	192.628
1084.96	196.284
1083.071	195.871
1082.173	194.977
1082.112	198.036
1081.214	197.142
1080.589	192.135
1080.492	195.158
1080.458	199.813

Table 5: All solution samples in the database selected by the SQL query "`select distinct objval - 17.366, lhsval from poplog where x49=1 and lhsval <= 200.0 order by objval desc limit 20;`"

objval	lhsval
1022.622	198.597
1022.618	193.262
1021.72	192.368
1017.648	192.277
1016.75	191.383
1014.171	195.692
1009.93	192.277
1007.312	194.294
1005.299	198.763
1003.071	190.879
1002.827	198.909
998.143	187.96
998.101	189.894
996.251	186.751
995.057	198.251
994.363	183.319
994.307	195.317
994	199.617
993.902	184.545
993.04	190.467

Table 6: All solution samples in the database selected by the SQL query "`select distinct objval, lhsval from poplog where x49=0 and lhsval <= 200.0 order by objval desc limit 20;`"

Thirtieth Hawaii International Conference on System Sciences. IEEE Computer Press, January 1997.

[3] S. O. Kimbrough, M. Lu, and S. M. Safavi. Exploring a financial product model with a two-population genetic algorithm. In *Proceedings of the 2004 Congress on Evolutionary Computation*, pages 855–862, Piscataway, NJ, June 19–23, 2004. IEEE Neural Network Society, IEEE Service Center. ISBN: 0-7803-8515-2.

[4] S. O. Kimbrough, M. Lu, and D. H. Wood. Exploring the evolutionary details of a feasible-infeasible two-population GA. In X. Yao et al., editors, *Parallel Problem Solving from Nature – PPSN VIII*, volume 3242 of *LNCS: Lecture Notes in Computer Science*, pages 292–301. Springer-Verlag, Berlin, Germany, 18-22 September 2004.

[5] S. O. Kimbrough, M. Lu, and D. H. Wood. Introducing distance tracing of evolutionary dynamics in a feasible-infeasible two-population (FI-2Pop) genetic algorithm for constrained optimization. In H. Bhargava, C. Forman, R. J. Kauffman, and D. J. Wu, editors, *Proceedings of the Ninth INFORMS Conference on Information Systems and Technology (CIST)*, 2004. Published as a CD only. See: `http://misrc.umn.edu/co-sponsored/informs_cist102304/INFORMS_CIST04_Program_102104.pdf`.

[6] S. O. Kimbrough, M. Lu, D. H. Wood, and D. J. Wu. Exploring a two-market genetic algorithm. In W. B. Langdon, E. Cantú-Paz, and et al., editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2002)*, pages 415–21, San Francisco, CA, 2002. Morgan Kaufmann.

[7] S. O. Kimbrough, M. Lu, D. H. Wood, and D. J. Wu. Exploring a two-population genetic algorithm. In E. Cantú-Paz and et al., editors, *Genetic and Evolutionary Computation (GECCO 2003)*, LNCS 2723, pages 1148–1159, Berlin, Germany, 2003. Springer.

[8] S. O. Kimbrough, S. A. Moore, C. W. Pritchett, and C. A. Sherman. On DSS support for candle lighting analysis. In *Transactions of DSS '92*, pages 118–135, June 8-10, 1992.

[9] S. O. Kimbrough and J. R. Oliver. On automating candle lighting analysis: Insight from search with genetic algorithms and approximate models. In J. F. Nunamaker, Jr. and R. H. Sprague, Jr., editors, *Proceedings of the Twenty-Sixth Annual Hawaii International Conference on System Sciences, Volume III: Information Systems: Decision Support and Knowledge-Based Systems*, pages 536–544, Los Alamitos, CA, 1994. IEEE Computer Society Press.

[10] S. O. Kimbrough and J. R. Oliver. Candle lighting analysis: Concepts, examples, and implementation. In V. C. Storey and A. B. Whinston, editors, *Proceedings of the Second Annual Workshop on Information Technologies and Systems*, pages 55–63, Dallas, Texas, December 12-13, 1992.

[11] S. O. Kimbrough, J. R. Oliver, and C. W. Pritchett. On post-evaluation analysis: Candle-lighting and surrogate models. *Interfaces*, 23(7):17–28, May-June 1993.

[12] S. Martello and P. Toth. *Knapsack Problems: Algorithms and Computer Implementations*. John Wiley & Sons, New York, NY, 1990.

APPENDIX

A. THE KNAP101 MODEL

The model discussed in this paper, Knap101, is a knapsack model with one constraint and 50 0-1 decision variables. The NetLogo code used to set up the model is as follows. From it, the full details of the model are obtainable.

```
to ModelSetup
set numDecisionVariables 50
set constraintRHS 200.0
set objectiveList (list 16.936 31.87 9.938 67.334
83.061 74.642 4.241 40.666 16.028 66.306 98.151
19.547 8.461 2.228 63.851 66.698 57.147 66.432
98.528 39.158 49.67 96.693 16.849 2.086 64.329
27.252 49.374 99.361 75.244 67.33 32.496 4.97
14.08 31.401 45.301 22.688 6.129 55.624 6.418
35.819 74.596 55.203 77.388 95.148 3.123 34.088
92.833 76.007 57.555 97.366)
set constraintList (list 39.628 2.975 30.141
29.355 6.824 31.625 3.415 6.786 7.732 49.275
41.955 2.383 23.821 7.884 47.518 40.901 2.618
23.126 39.101 30.386 36.086 12.196 41.528 13.015
32.113 44.648 2.67 41.426 11.598 24.02 13.358
0.985 41.282 0.346 17.901 21.295 6.847 6.667
45.437 2.085 42.106 38.617 23.309 31.843 40.966
43.892 24.982 17.617 30.331 4.992)
end
```

The optimal solution produces an objective function value of 1119.984 and a left-hand side (LHS) value of 198.254. The bang-per-buck heuristic's best solution is 1119.086.

The data used in this report were obtained from a single run of the Knap101 model, using the NetLogo (version 3.1beta3) implementation, version 1.6. The run described here used 100 as its random seed. The maximum number of solutions was 60 for the feasible population and 60 for the infeasible population. Single point crossover was used with a probability of 0.4. The point mutation rate was 0.025 and tournament selection was used. The run went for 500 generations and called the fitness evaluation function 67,235 times. The file is `knapsack-generate-FI2PopGA.nlogo`.

An optimum solution was found on this run at generation 293. That solution was `[0 1 0 0 1 0 0 1 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 1 1 1 1 0 1 0 0 0 1 0 1 0 0 1 1 0 0 1 1 0 1]`. Not every run finds the optimal solution. The general results, however, are quite typical, in particular those in Figure 2. Key data/log files for the run are: lineage.pl, log-BpB-results.csv, log-GA-results.csv, and log-of-populations-.csv.

These files may be found at `http://opim-sky.wharton.upenn.edu/~sok/sokpapers/2007/knap101-20060430`. The plot in Figure 2 was produced by the Mathematica notebook file, plotting.nb.

objval	200.0 - lhsval
1119.984	1.74600000000004
1119.086	2.63999999999999
1118.686	0.0229999999999961
1115.014	2.73099999999999
1114.116	3.62500000000003
1109.028	1.92900000000003
1107.296	2.73099999999999
1106.859	4.85100000000003
1104.678	0.714000000000027
1104.084	1.018
1104.058	2.91399999999999
1103.516	7.37200000000001
1102.326	3.71600000000001
1100.437	4.12899999999999
1099.539	5.023
1099.478	1.964
1098.58	2.858
1097.955	7.86500000000004
1097.858	4.84200000000001
1097.824	0.187000000000012
1097.397	3.61599999999999
1096.364	0.574000000000012
1095.509	7.048
1095.467	5.114
1095.346	0.80000000000004
1094.8	7.499
1094.569	6.00799999999998
1094.448	1.69399999999999
1093.957	3.80500000000004
1093.617	8.25700000000003
1093.583	3.602
1092.719	9.15100000000001
1092.355	1.30600000000004
1092.17	4.23400000000001
1091.99	1.69899999999998
1091.733	6.35400000000001
1091.729	11.689
1091.457	2.20000000000002
1091.272	5.12799999999999
1091.268	10.463

Table 7: All solution samples in the database selected by the SQL query "`select distinct objval, 200.0 - lhsval from poplog where lhsval <= 200.0 and x49 = 1 order by objval desc limit 40;`"