December 1999

An Agent Enhanced Spreadsheet Solver for Multi-Criteria Decision Making

Paul Bergey
Virginia Polytechnic Institute and State University

An Agent Enhanced Intelligent Spreadsheet Solver for Multi-Criteria Decision Making

Paul K Bergey, Virginia Polytechnic Institute and State University, pkbergey@vt.edu

Introduction

Single criteria optimization methods often fail to capture the complexity of problems faced by decision makers (DMs) in today's rapidly changing business environment. The mathematical foundation for multi-criteria decision making (MCDM) was developed over a century ago [Pareto, 1896]. Given the growing constituency of PC literate DMs in the business community, it is astonishing that the existing tools for MCDM are still too difficult to be used by non-mathematicians. Characteristic of a typical MCDM problem is the absence of a unique global optimum. Rather, multiple solutions to the problem often exist that are superior to (dominate) the others in the solution space. These solutions are known as Pareto optimal solutions.

The DM can be further confounded with multiple constraints associated with the MCDM problem. Most conventional solution methods to constrained MCDM problems provide the DM with only a single solution, usually based upon some pre-specified preference among alternatives. Recently, a great deal of research interest has been spawned in the use of Evolutionary Algorithms (EAs) for the MCDM problem because of the EA's unique ability to provide multiple Pareto optimal solutions in a single run. Furthermore, this can be accomplished without any prerequisite information about preferences from the DM. To this end, the author is developing a Decision Support System (DSS) that provides the DM with a set of Pareto optimal (non-dominated) solutions to constrained MCDM problems.

The search procedure used to generate the Pareto set is based upon a recently introduced algorithm known as Differential Evolution (DE). DE has shown considerable promise for global optimization of a single, continuous space objective function. The author has made several enhancements to DE to address multiple objective functions. The DSS provides the DM with a set of alternative non-dominated solutions from which to choose. The enhanced algorithm is referred to as Pareto Differential Evolution (PDE).

PDE is implemented as a general-purpose spreadsheet solver designed as an add-in for Microsoft Excel. The primary objective of PDE is to help the DM with making better decisions. To accomplish this task PDE provides an interface that is intuitive to use and simple to map MCDM problems into. While EAs generally require specified control parameters to search efficiently, PDE shields the DM from such tasks to the extent that the DM is completely unaware that an EA is even used in the optimization process. The control parameter settings are initially established using heuristics, and then altered during the optimization run by an Optimization Agent (OA) as information about the MCDM problem is discovered. Furthermore, the DM may interact with the optimization run by halting it at any time to consider the current set of non-dominated solutions. Using preferences between the alternative non-dominated solutions, an approximate range of weights are captured by a Utility Agent (UA) and used to guide the subsequent search. The goal of the UA is to focus the population of evolving solutions in the neighborhood of preferred alternatives, making the search process more efficient and increasing the likelihood of finding an acceptable solution for the DM.

EA Based Spreadsheet Solvers

The 1990s will be remembered by many as a decade in which the tools of OR/MS were made accessible to the masses via the power of spreadsheets. At the forefront of this movement are companies like Frontline Systems. In 1990, Frontline won a competition among third-party developers to create a solver on an OEM basis for Microsoft. In 1991, the new solver was introduced with Microsoft Excel 3.0 which optimized a single objective function using the Revised Simplex Algorithm for linear programming and the GRG algorithm for non-linear programming. The latest version of solver from Frontline systems is now available for beta testing and is enhanced with an evolutionary solver for global optimization of a single non-linear objective function.

Evolutionary algorithms designed for solving spreadsheet models are a relatively new concept. Conventional optimization methods left some DMs dissatisfied with the modeling functionality of the installed solver. For example, the current solver tool delivered with Microsoft Excel '97 is not capable of coping with statements of the type {IF, OR, AND, NOT} which represent discontinuities in the response surface of the objective function. DMs of today's business world could not be denied by such a simple obstacle and thus, the paradigm of evolutionary solvers for spreadsheet was born.

At the forefront of EA based spreadsheet solvers are Palisade Corporation and Ward Systems Group, Inc.

Palisade recently released Evolver version 4.0, their evolutionary solver designed as an add-in for Microsoft Excel. Palisade pioneered the concept of integrating the EA paradigm with the flexible modeling environment of the spreadsheet with their first release of Evolver in 1989. Following in their path was Ward Systems Group, Inc. with Gene Hunter in 1995. Today, these two competitors share a rapidly expanding user-base of researchers and managers with the need to solve complex spreadsheet models. The opportunity for information systems professionals to make available new and exciting research ideas to managers with practical problems has never been so great.

PDE essentially picks up where Evolver, Gene Hunter, and Evolutionary Solver (by Frontline Systems) left off. The shortcoming of the above systems is their limited ability to handle multiple objective functions (the MCDM problem). PDE is designed to provide DMs with a useful mechanism to optimize a spreadsheet model that contains multiple objective functions. The functions need not be commensurable, which is often the case in practical problems, and may also be subject to any number of constraints.

Multi-Criteria Decision Making

Many real world problems consist of a variety of performance measures often conflicting in their objectives. The process of simultaneously minimizing (or maximizing) n components of $F_i(x_j)$ for a vector of decision variables x_j is known as the multi-criteria decision making problem. Without loss of generality, it can be represented as follows:

$$\text{Min:} \quad F(x_j) = (f_1(x_j), f_2(x_j), \ldots f_n(x_j)) \qquad j = 1 \ldots m \tag{1}$$

$$\text{subject to:} \quad G(x_j) \geq c_k \quad (\text{or} \leq, <, >) \tag{2}$$

$$\text{where:} \quad G(x_j) = (g_1(x_j), g_2(x_j), \ldots, g_p(x_j)) \qquad c_k = (c_1, c_2, \ldots, c_p) \tag{3}$$

Definition 1 - Pareto Dominance. A vector $x^* \in R^n$ is said to dominate $x \in R^n$ if x^* is better than x in all of its components. Thus, for a minimization problem $F_i(x_j^*) < F_i(x_j)$ for all i and j.

Definition 2 - Pareto Optimal Solutions. A vector $x \in R$ is a Pareto optimal solution if there exists no other solution vector $x' \in R^n$ for which $F(x')$ dominates $F(x)$. Such solutions are also referred to as non-dominated, efficient or non-inferior.

A typical MCDM problem has a set of solutions that are superior to (dominate) the others in the search space and are referred to as the Pareto set. Each solution in the Pareto set is optimal in the sense that it is not possible to improve upon any one of the n components of $F_i(x)$ without deteriorating at least one of the other components. The best solution is often based upon trade-offs between objectives made by the DM. While the final choice is subjective, clearly the solutions presented to the DM should be Pareto optimal (non-dominated).

To solve the MCDM problem, generating a set of Pareto optimal solutions for the DM is only the first step. The problem requires that the DM articulate preferences either before, after, or during the search process in order to arrive at a single final solution [Hwang and Masud, 1979]. The task is usually accomplished via one of the following three methods: First, the weighted sum approach is a classic approach to aggregating a variety of functions to provide a single measure of utility [Hwang and Masud, 1979]. Second, ranking objectives and optimizing them in order is referred to as the lexicographic method [Ben-Tal, 1980]. Third, goal values may be specified by the DM for each objective function. This is often a preferred method because the desired outcomes (specified goals) are easy for the DM to describe and relate to. However, the interpretation of goals can have a variety of hidden complexities as described in [Dinkelbach, 1980].

Evolutionary Algorithms and MCDM

EAs represent a powerful, general purpose optimization paradigm where the computational process mimics Darwin's theory of biological evolution. The popular components of EAs include Genetic Algorithms (GAs) [Holland, 1975], Evolution Strategies (ESs) [Rechenberg, 1973], Evolutionary Programming (EP) [Fogel, 1991], and Genetic Programming (GP) [Koza, 1992].

In a nutshell, most EAs start with a set of chromosomes (numeric vectors) representing possible solutions to a problem. The individual components (numeric values) within a chromosome are referred to as genes. New chromosomes are created by crossover (the probabilistic exchange of values between vectors) or mutation (the random alteration of values within a vector). Chromosomes are then evaluated according to a fitness (or objective) function with the fittest surviving into the next generation. The result is a gene pool that evolves over time to produce better and better solutions to a problem.

The notion of a non-dominated solution set is particularly suitable to a population based search strategy. By exploiting the characteristics of the currently non-dominated solutions in a population, stronger individuals eventually emerge that dominate the previously non-dominated solutions. Early attempts to solve MCDM problems involved consolidating the multiple objective functions into a single aggregate fitness function a priori to the optimization process. If the best solution found is not acceptable to the DM, then the preferences must be revised and the optimization run repeated. Applying EAs to MCDM was motivated by their effectiveness in locating multiple non-dominated solutions in a single optimization run. The seminal work in this area was accomplished using a GA by [Schaffer, 1986] and using an ES by [Kursawe, 1991].

Schaffer's Vector Evaluated Genetic Algorithm (VEGA) was centered on multiple populations evolving using a separate fitness function for each population. In each generation offspring are produced by applying genetic operators (crossover & mutation) that merge the populations of chromosomes. The populations are monitored during the evolutionary process for non-dominated solutions. VEGA has been shown to split into population species when the trade-off surface is concave (chromosome strong in only a single objective), forming clusters of solutions near the extreme areas. For a detailed description of VEGA and a general overview of alternative evolutionary approaches to the MCDM problem see [Fonseca and Flemming, 1995].

A Pareto ranking technique was subsequently proposed in the Pareto Genetic Algorithm (PGA) [Goldberg, 1989]. In PGA, solution vectors that are non-dominated in the current population are given a rank of one and an equal probability of reproducing. Then, the non-dominated solutions are removed from the current population to expose a second layer of previously dominated solutions that have now become non-dominated. These solutions are given a rank of two and an equal but lower probability of reproducing than those of higher rank. The process continues until the entire population of solution vectors is ranked. Goldberg's method proved to be effective on non-convex trade-off surfaces that present difficulties to some other techniques.

Other EA based methods for the MCDM problem that have been developed recently include: Niched Pareto Algorithm [Horn et al 1993], Pareto Ranking Based Genetic Algorithm [Belegundu et al, 1994], Non-dominated Sorting Genetic Algorithm [Srinivas and Deb, 1995], Multi-Sexual Genetic Algorithm [Lis and Eiben, 1997], and the first Hybrid Multi-objective GA [Ishibuchi and Murata, 1998].

Pareto Differential Evolution (PDE)

PDE differs from existing spreadsheet optimizers in its use of software agents to help DMs make better decisions. Software agents have been coined by some as "...the most important computing paradigm in the next ten years" [Gilbert, 1997]. Incorporating this emerging technology into DSS has recently become of great interest to researchers in the field [Whinston, 1997; Elofson, Beranek, & Thomas, 1997; Maturana & Norrie, 1997; Oliver, 1996; Pinson, Louca, & Moraitis, 1997; Hess, Reese, & Rakes, 1999]. Because the MCDM problem in inherently subjective, software agents are particularly suitable for supporting and capturing the necessary information from the DM. PDE incorporates agency concepts that unburden the DM of repetitive or tedious tasks more suitable to a computer program. Such tasks include specifying EA control parameters which ultimately affect the efficiency of the search and the quality of the final solution. For complex problems, the optimal choice of control parameters may change as the search progresses requiring constant attention from some entity capable of enacting a change. PDE is enhanced with an Optimization Agent (OA) designed specifically for this task.

Another unique feature of PDE is the Utility Agent (UA) that guides the search for non-dominated solutions. The PDE search process begins with a randomly generated set of solution vectors. The solution vectors have appended to them a set of weights that determine the search directions within the solution space. Should the DM choose not to interact during the optimization run then random search directions are used throughout. However, if the DM wishes to interact by halting the optimization, the UA will present the current set of non-dominated solutions to the DM. The DM's preferences are expressed by ranking a subset of the solutions. The ranking of alternatives by the DM allow the UA to capture preference information used to adjust future search directions. The UA is able to guide the search because in each new generation it provides updated weights to the appended array. The objective of the UA is to focus the effort of the search process in the general direction desired by the DM. The simple concept is that the final set of non-dominated solution vectors is likely to be in the region of preferred alternatives of the DM, ultimately resulting in better decisions for an MCDM problem.

References

Storn, R., and Price, K., Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces, *Journal of Global Optimization*, (11), 1997, pp.341-359.

Whinston, A., “Intelligent Agents as a Basis for Decision Support Systems,” *Decision Support Systems* (20:1), May 1997, pp. 1.

"Additional references available upon request."