Multiobjective Optimization

Introduction

Many (perhaps most) real-world design problems are, in fact, multiobjective optimization problems in which the designer seeks to optimize simultaneously several performance attributes of the design and an improvement in one objective is often only gained at the cost of deteriorations in other objectives — trade-offs are necessary.

There are two standard methods for treating multiobjective problems, if a traditional optimization algorithm which minimizes a single objective is to be employed. One is to construct a composite objective:

$$\text{Minimize } \tilde{f} = \sum_{i=1}^{N} a_i f_i, \tag{1}$$

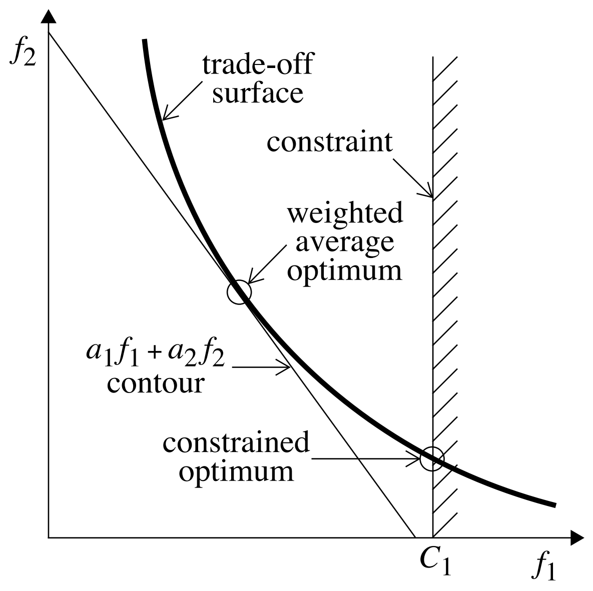
where the f_i are the N objectives to be minimized and the a_i are positive-valued weightings. The other is to place constraints on all but one of the objectives, i.e:

$$\text{Minimize } f_j \text{ subject to } f_i \le C_i \;\; \forall\; i = 1, N;\; i \ne j, \tag{2}$$

where the C_i are the constraint limits.

Whichever of these approaches is used, the solution of the one-objective problem so produced results in the identification of a single point on the trade-off surface, the position of which depends on the designer's preconceptions (the values of a_i or C_i chosen) as illustrated in Figure 1. In order to explore the trade-off surface further a large number of different optimization runs must be executed each with different weightings or constraints — a potentially time-consuming and computationally expensive exercise if even attempted.

Figure 1: Types of Multiobjective Optimum.



Of course, eventually a single solution must be chosen, but it is self-evident that the designer will make a better informed decision if the trade-off surface between the conflicting objectives can be inspected before this choice is made. By using suitably adapted stochastic optimization methods it is possible to reveal the trade-off surface of a multiobjective optimization problem in a single run. In the following sections appropriate adaptations to standard Genetic Algorithm (GA) and Simulated Annealing (SA) implementations will be discussed.

Multiobjective Archiving

Adapting any stochastic optimization algorithm to perform multiobjective optimization will inevitably require a common change to the method of archiving. In multiobjective optimization solutions lying on the trade-off surface (or *Pareto front* as it is also known) are sought.

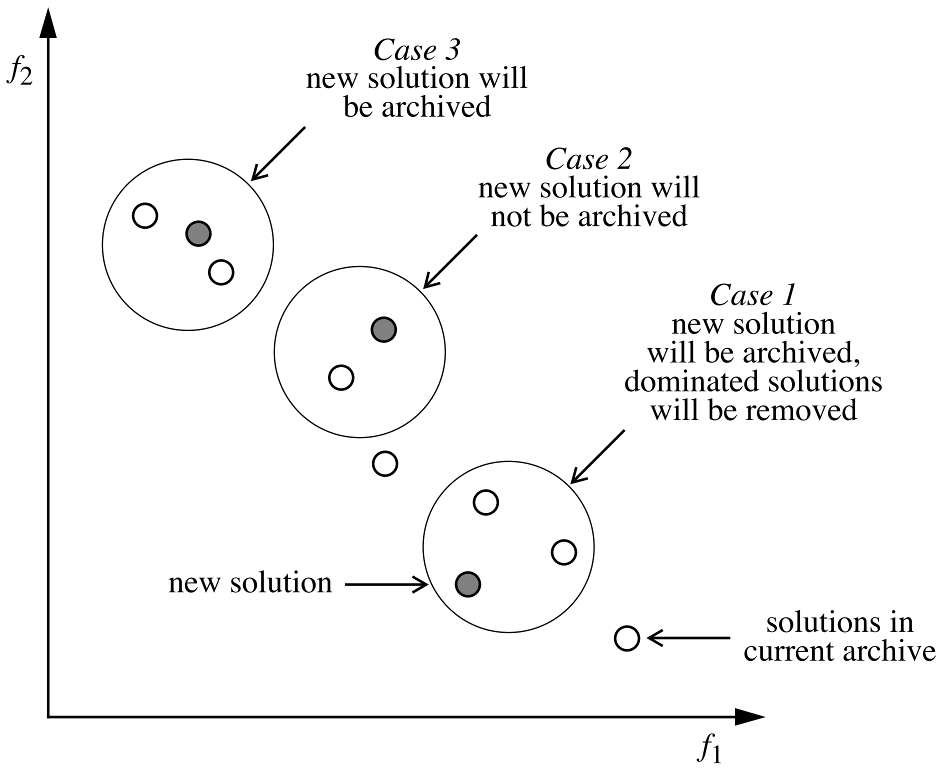
Any solution on the Pareto front can be identified formally by the fact that it is not *dominated* by any other possible solution. A solution **X** is said to be *dominated* by solution **Y** if **Y** is at least as good on all counts (objectives) and better on at least one, i.e., assuming all M objectives are to be minimized, if

$$f_i(\mathbf{Y}) \le f_i(\mathbf{X}) \quad \forall\, i = 1, M \quad \text{and} \quad f_i(\mathbf{Y}) < f_i(\mathbf{X}) \quad \text{for some } i\,. \tag{3}$$

Thus, in multiobjective optimization an archive of the best (i.e. the nondominated) solutions found should be maintained. A suitable archiving scheme, illustrated in Figure 2, is as follows:

- All feasible solutions generated are candidates for archiving.

Figure 2: Multiobjective Archiving.



- If a candidate solution dominates any existing members of the archive, those are removed and the new solution is added (Case 1).
- If the new solution is dominated by any existing members of the archive, it is not archived (Case 2).
- If the new solution neither dominates nor is dominated by any members of the archive, it is added to the archive (Case 3).

Using this scheme, it is hoped that, as the search progresses, the archive will converge onto the true trade-off surface between the objectives.

MULTIOBJECTIVE SA

SA has traditionally been used for single objective optimization. However, Engrand [1997] proposed a multiobjective variant on the SA algorithm. Although Engrand's original algorithm performs adequately, it has a couple of identifiable weaknesses, which are readily overcome. The multiobjective SA algorithm described in the following sections is therefore the improved version of Engrand's proposed algorithm (Suppapitnarm *et al.* [2000]).

The structure of the multiobjective SA algorithm is shown in Figure 3. The key differences compared to a single objective SA implementation are as follows.

Solution Acceptance

The overall acceptance probability is the product of a series of individual acceptance probabilities for each of the M objectives to be minimized:

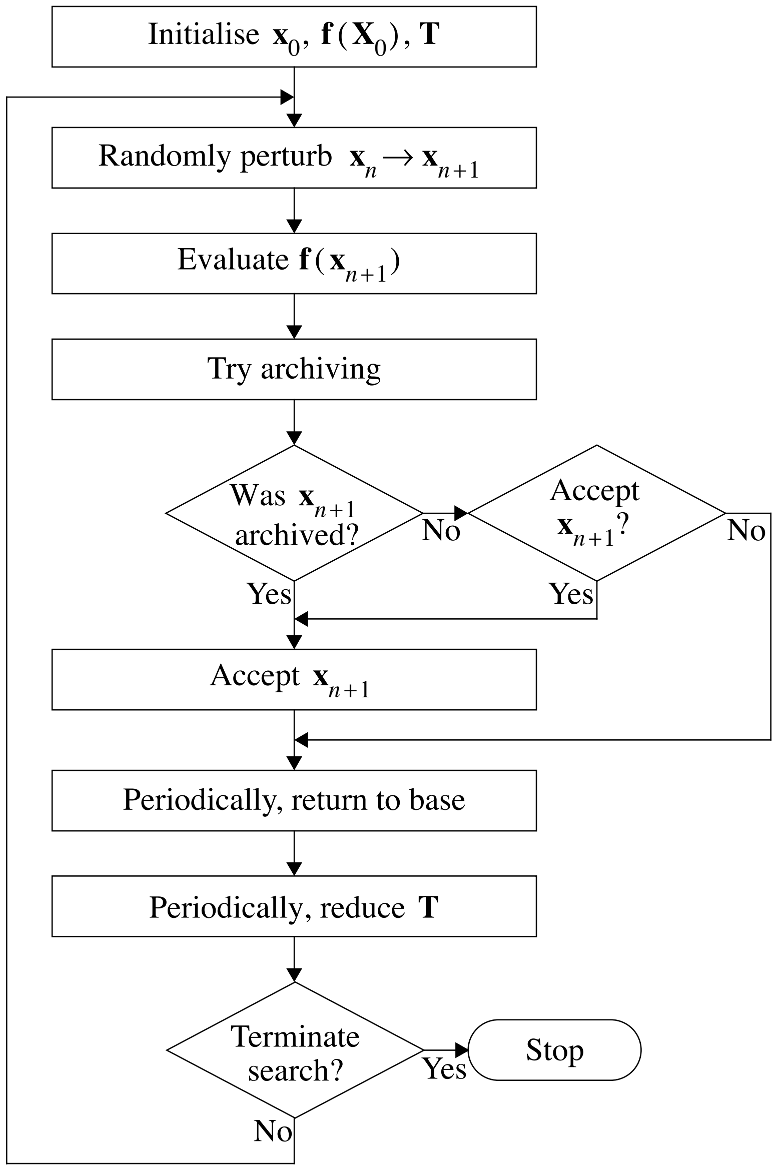
$$P = \prod_{i=1}^{M} p_i = \prod_{i=1}^{M} \exp\left(-\frac{[f_i(\mathbf{x}_{n+1}) - f_i(\mathbf{x}_n)]}{T_i}\right). \tag{4}$$

Note that each objective has its own associated temperature T_i, which obviates the need to scale the objectives carefully with respect to each other, as long as appropriate temperatures can be determined automatically. The individual 'probabilities' p_i will be greater than or less than unity depending whether the move from $\mathbf{x}_n$ to $\mathbf{x}_{n+1}$ has decreased or increased that particular objective f_i. Thus, the overall acceptance probability P can be regarded as giving an indication as to whether the move is predominantly downhill $P \geq 1$ or uphill $P < 1$.

Any move in which at least one objective is decreased is, however, potentially a move onto the trade-off surface. The acceptance probability of such a move (given by equation (4)) depends on the relative changes of all the objectives and the current temperatures, and thus there is no guarantee that the move will be accepted. If it is a move onto the trade-off surface, then it clearly should be accepted (and archived). Therefore, as shown in Figure 3, the acceptance logic has an important additional feature:

- Each new solution generated is *first* submitted as a candidate for archiving.

Figure 3: Multiobjective SA Algorithm Structure.


Initialise $\mathbf{x}_0$, $\mathbf{f}(\mathbf{X}_0)$, $\mathbf{T}$
Randomly perturb $\mathbf{x}_n \rightarrow \mathbf{x}_{n+1}$
Evaluate $\mathbf{f}(\mathbf{x}_{n+1})$
Try archiving
Was $\mathbf{x}_{n+1}$ archived?
No
Accept $\mathbf{x}_{n+1}$?
No
Yes
Yes
Accept $\mathbf{x}_{n+1}$
Periodically, return to base
Periodically, reduce $\mathbf{T}$
Terminate search?
Yes
Stop
No

- If the solution is archived, then it is automatically accepted.
- If the solution is not archived, then it is accepted with a probability given by equation (4).

This ensures that all moves which identify a new solution on the current trade-off surface are accepted and archived.

Annealing Schedules

The formulation of the acceptance probability calculation (equation (4)) requires a different temperature for each objective. One way in which this requirement can be met is by implementing annealing schedules as follows:

- Initially all the temperatures T_i are set to ∞, so that all feasible moves are accepted. After a predetermined number of trials the temperatures are set to appropriate values using White [1984]'s formulation:

$$T_i = \sigma_i , \tag{5}$$

where σ_i is the standard deviation of the variation of f_i observed.

- Periodically thereafter the temperatures are lowered according to the formula:

$$T_i' = \alpha_i T_i , \tag{6}$$

where α_i is determined using the formulation of Huang *et al.* [1986]:

$$\alpha_i = \max\left[0.5, \exp\left(-\frac{0.7T_i}{\sigma_i}\right)\right] . \tag{7}$$

Return To Base (Restarts)

In a traditional single objective SA implementation the "return to base" option retrieves the best solution found and continues the search from there. This only occurs if the search is deemed to have ceased to make satisfactory progress.

In this multiobjective SA implementation when a return to base occurs a solution selected from the archive is retrieved and the search is continued from there. This is done to try to ensure that the entire trade-off surface is explored and therefore returns to base occur much more frequently than in a single objective SA implementation.

Suppapitnarm [1998] investigated various return to base strategies and concluded that a scheme in which returns to base are made to a solution randomly selected from a candidate list consisting of

- the *M* solutions in the archive which have the lowest value of each individual objective, i.e. the solutions at the ends of the trade-off surface, and
- about half-a-dozen solutions selected randomly from the other solutions in the archive

gives consistently good algorithm performance.

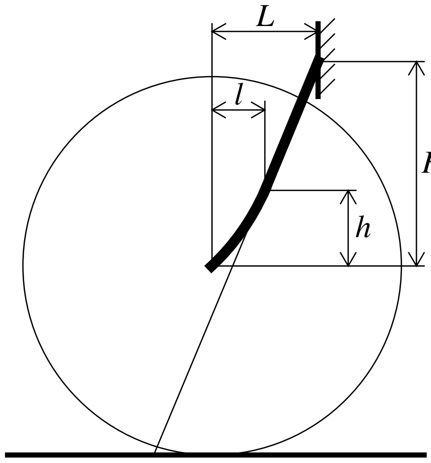
Example: Multiobjective Optimisation Of Bicycle Fork Design

Figures 5 and 6 illustrate the performance of the multiobjective SA algorithm described in optimising the general shape of a pair of bicycle front forks to be manufactured from a predetermined material (steel). Three objective functions were considered:

- minimisation of the material volume,
- minimisation of the deflection under frontal impact loading, and
- minimisation of the angular deflection under cornering loading.

The legs of the forks were assumed to be made of tubular steel with a two-part profile, as shown in Figure 4, consisting of a straight section and a curved (quadratic profile) section.

Figure 4: Bicycle Front Fork Geometry.



The overall height, *H*, and length, *L*, of the forks are fixed, but the height, *h*, and length, *l*, of the curved section may be varied. In addition, the radius of the tubular section at the fork tip, *r*, the tube thickness, *t*, and the degree of tapering of the forks (so that the radius of the tubular

Figure 5: Archive Evolution.

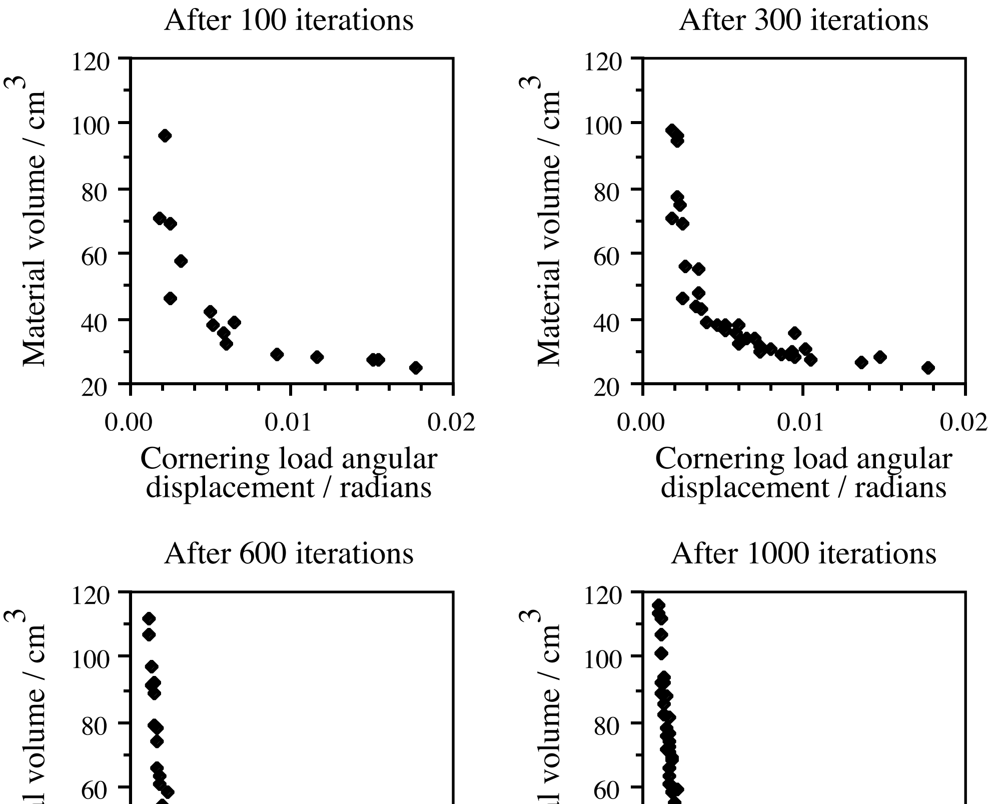
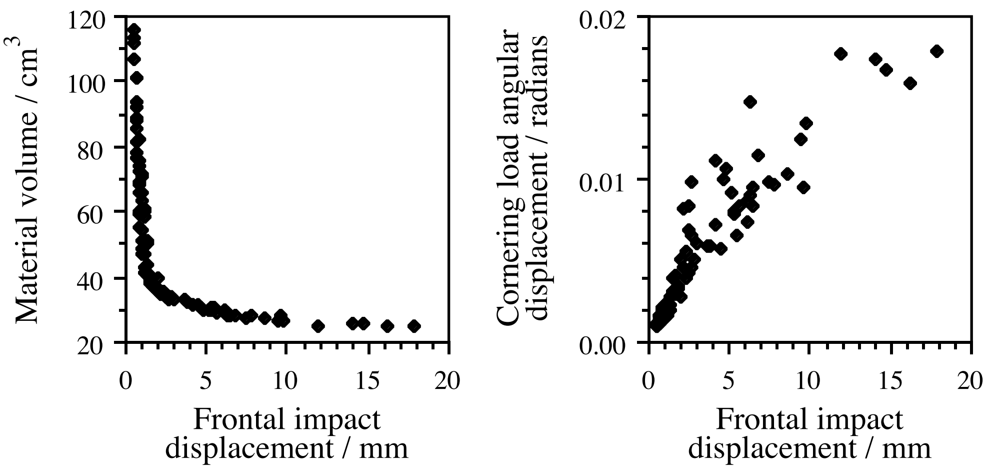


Figure 6: Final Archives In Two-Objective Space.



section increases with distance from the tip) can be varied. Thus, there are five control variables in all. A limit was imposed on the maximum allowed stress in the two loading cases.

Figure 5 shows the evolution in two-objective space (material volume versus cornering load angular displacement) of the contents of the archive, demonstrating how the archive converges onto the trade-off surface as the search progresses.

Figure 6 shows the final archive contents in the other two-objective projections (material volume versus impact load displacement and cornering load angular displacement versus impact load displacement). The trade-offs between the displacement objectives and the material volume are clear. The fact that heavier forks displace less under load is not unexpected, but the severity of the 'elbow' shaped trade-off surfaces revealed by the optimization is, perhaps, surprising.

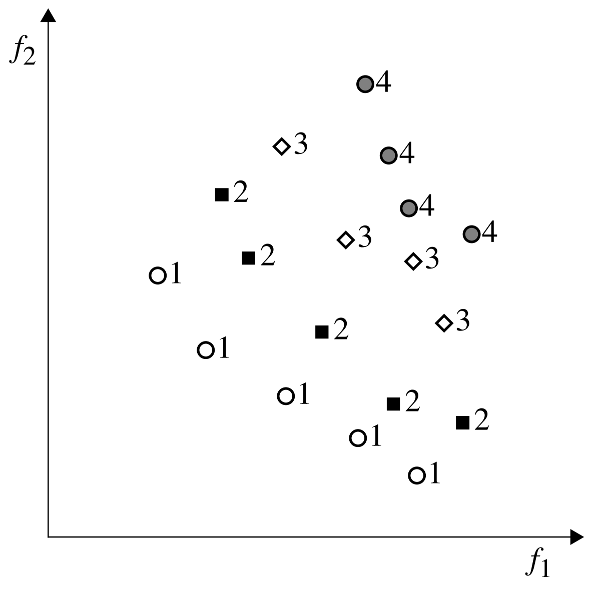
MULTIOBJECTIVE GA

The fact that GAs search from population to population rather than from one individual solution to another makes them very well suited to performing multiobjective optimization. It is easy to conceive of a population being evolved onto the trade-off surface by a suitably configured GA. In fact, with an appropriate archiving scheme in place, the only modification required to a single objective GA, in order to perform multiobjective optimization, is in the *selection* scheme. As with single objective GAs, a wide variety of multiobjective selection schemes have been devised. Three of the most widely used (and most easily implemented) will be described here.

Pareto-Based Selection

This selection scheme was first proposed by Goldberg [1989]. It works by ranking the current population as follows:

Figure 7: An Example of Pareto-Based Population Ranking.



- First, the nondominated members of the population are identified. These are assigned rank 1.
- The rank 1 individuals are then removed from consideration and the nondominated members of the remaining population are identified. These are assigned rank 2.
- The rank 2 individuals are also removed from consideration and the nondominated members of the remaining population are identified. These are assigned rank 3.
- And so on until the entire population has been ranked.

An example of the result of this process is shown in Figure 7.

Having ranked all members of the current population, they are assigned selection probabilities based on their rankings in a manner similar to Baker [1985]'s single criterion ranking selection procedure. The probability of a rank n member of the current population being selected is given by:

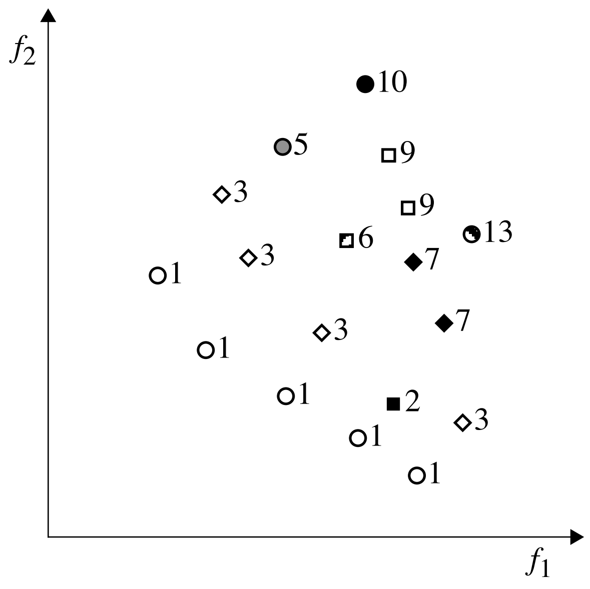
$$p_n = \frac{S(N+1-R_n)+R_n-2}{N(N-1)}, \tag{8}$$

in which N is the population size, S is a selection pressure and

$$R_n = 1 + r_n + 2\sum_{i=1}^{n-1} r_i, \tag{9}$$

where r_i is the number of solutions in rank i. Selection using these probabilities can then be performed using any of the standard GA methods.

Figure 8: An Example of Dominance Count Ranking.



Dominance Count Ranking

Fonseca and Fleming [1993] proposed a slightly different scheme in which an individual's rank corresponds to the number of members of the current population by which it is dominated (plus 1). Thus, as in Goldberg's scheme, all the nondominated solutions have rank 1, but dominated individuals are penalized according the population density in the corresponding region of the trade-off surface. Figure 8 shows the same population as that in Figure 7 ranked using dominance counting. The difference in the ranking of some solutions is clearly quite dramatic. As with Pareto-based ranking, selection probabilities can then be assigned to individuals based on their ranking.

Tournament Selection

Tournament selection can be applied in multiobjective GAs, just as in single objective GAs. A subset of solutions is randomly selected and the nondominated solutions in this subset identified. If more solutions are nondominated than are required to be selected (in tournament selection schemes either one or two parents are chosen in each tournament), then the required number are chosen randomly from these nondominated candidates.

Other Considerations

The other key component parts of a GA (crossover and mutation operators) do not need to be changed in order to perform multiobjective optimization.

Many variants and elaborations on the comparatively simple multiobjective GA described here have been developed. These have been well reviewed recently by Deb [2001].

Algorithm Performance

The case study to be presented in the final lecture of this course will describe the application of a multiobjective GA to a difficult real-world problem, the design of Pressurised Water Reactor reload cores.

MULTIOBJECTIVE TABU SEARCH

To adapt the single-objective Tabu Search (TS) implementation described in an earlier lecture to perform multiobjective optimization the following aspects of the algorithm need to be modified: search point comparison; the Hooke & Jeeves (H&J) move; optimal point archiving and the Medium Term Memory (MTM); search intensification; and restart strategy. The adaptations described here are those developed by Jaeggi *et al*. [2008].

Search Point Comparison

In single-objective TS points are compared on the basis of their objective function values; in multiobjective TS (MOTS) the basis for comparison uses the concepts of dominance and Pareto-equivalence[†].

Hooke & Jeeves Move

A set of candidate neighbouring solutions is generated by incrementing and decrementing individual control variables in the same way as in single-objective TS. The objective functions for each new point are evaluated and, as long as the point is neither Tabu (i.e. not a member of the Short Term Memory) nor violates any constraints, it is considered as a candidate for the next point in the search.

In the single-objective TS algorithm, these candidates are sorted and the one with the lowest objective is chosen as the next point. A similar logic can be applied to the multiobjective case: however, the possibility of multiple points being Pareto-equivalent must be allowed for. This is achieved by classifying each candidate point according to its domination or Pareto-equivalence to the current point:

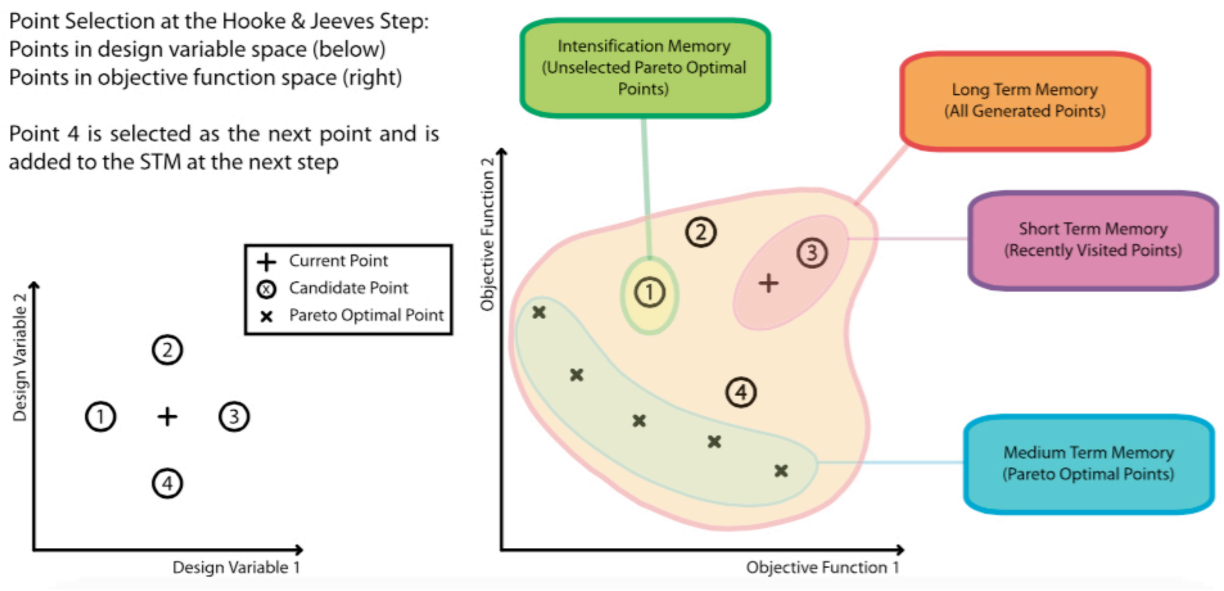
- If there is a single dominating point, it is automatically accepted as the next point.
- If there are multiple dominating points, the dominated points within that group are removed and one is selected at random from those remaining. The other non-dominated points become candidates for intensification (discussed below).
- If there are no dominating points, the same procedure is applied to those candidate points which are Pareto-equivalent to the current point.
- If there are no Pareto-equivalent points, a dominated point is selected in the same fashion.

[†] Two solutions are Pareto-equivalent if neither dominates the other.

Thus, the strategy accepts both “downhill” and “uphill” moves – the next point is simply the “best” allowed point (or one of the Pareto-equivalent best points) selected from the candidate solutions.

A *pattern move* strategy is also implemented: Before every second H&J move, the previous move is repeated. This new point is compared to the current one, and, if it dominates it, is accepted as the next point; if not, a standard H&J move is made. In this way, the search may be accelerated along known “downhill” directions. The basic search pattern is shown in Figure 9.

Figure 9: Point selection for the H&J move and MOTS memories (Jaeggi *et al*. [2008]).


Point Selection at the Hooke & Jeeves Step:
Points in design variable space (below)
Points in objective function space (right)
Point 4 is selected as the next point and is added to the STM at the next step
Current Point
Candidate Point
Pareto Optimal Point
Design Variable 2
Design Variable 1
1
2
3
4
Objective Function 2
Objective Function 1
Intensification Memory (Unselected Pareto Optimal Points)
Long Term Memory (All Generated Points)
Short Term Memory (Recently Visited Points)
Medium Term Memory (Pareto Optimal Points)

Optimal Point Archiving

The MOTS MTM is the (unbounded) set of non-dominated solutions produced by the search. As new points are evaluated, they become candidates for addition to this set. Thus, the MTM represents the Pareto-optimal set for the problem at that stage in the search.

Search Intensification

The original single-objective TS produces intensification points by using the MTM to generate points in the neighbourhood of good solutions. MOTS uses an alternative strategy.

A multi-objective H&J iteration may produce multiple Pareto-optimal points (see Figure 9). As only one point can be selected as the next point, it seems wasteful to discard the other points. Therefore, an *Intensification Memory* (IM) is incorporated into the MOTS algorithm. The IM is a set of Pareto-equivalent points; at each H&J step, points which dominate the current solution, but are not selected as the next point, are considered as candidates for addition to the set. At search intensification, a point is chosen randomly from the IM. The IM is continuously updated and points which become dominated by the addition of a new point are removed. Thus, the IM should always contain points which are on, or near to, the current Pareto-optimal front (stored in the MTM). Figure 9 shows the relationship between the various TS memories.

Restart Strategy

The single-objective TS restart strategy returns the search to the current best point in the MTM (and reduces the step sizes). As the MTM is now a set of Pareto-optimal points, one point is selected at random from this set when the search is restarted.

Algorithm Performance

Examples of the performance of this algorithm on real-world problems can be found in various papers, for example that by Kipouros *et al.* [2008]. A paper by Ghisu *et al.* [2010] examines the performance improvements that can be gained by incorporating an adaptive parametrization based on Principal Component Analysis within the MOTS framework.

References

Baker, J.E., (1985) "Adaptive Selection Methods for Genetic Algorithms", 101-111 in *Proceedings of an International Conference on Genetic Algorithms and their Applications* (J.J. Grefenstette, editor), Lawrence Erlbaum Associates, Hillsdale, NJ.

Deb, K., (2001) *Multi-objective Optimization Using Evolutionary Algorithms*, John Wiley & Sons Ltd., New York, NY.

Engrand, P., (1997) "A Multi-Objective Approach Based on Simulated Annealing and its Application to Nuclear Fuel Management", *Proc. 5th Int. Conf. Nuclear Engineering*, Nice, ICONE5-2523.

Fonseca, C.M., and P.J. Fleming (1993) "Genetic Algorithms for Multi-Objective Optimization: Formulation, Discussion and Generalization", 416-423 in *Genetic Algorithms: Proceedings of the Fifth International Conference* (S. Forrest, editor), Morgan Kaufmann, San Mateo, CA.

Ghisu, T., G.T. Parks, D.M. Jaeggi, J.P. Jarrett and P.J. Clarkson (2010) "The Benefits of Adaptive Parametrization in Multi-objective Tabu Search Optimization", *Eng. Opt.* **42**, 959-981.

Goldberg, D.E., (1989) *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley, Reading, MA.

Huang, M.D., F. Romeo and A. Sangiovanni-Vincentelli (1986) "An Efficient General Cooling Schedule for Simulated Annealing", *Proc. IEEE Int. Conf. Computer Aided Design*, 381-384.

Jaeggi, D.M., G.T. Parks, T. Kipouros and P.J. Clarkson (2008) "The Development of a Multi-objective Tabu Search Algorithm for Continuous Optimisation Problems", *Eur. J. Oper. Res.* **185**, 1192-1212.

Kipouros, T., D.M. Jaeggi, W.N. Dawes, G.T. Parks, A.M. Savill and P.J. Clarkson (2008) "Biobjective Design Optimization for Axial Compressors Using Tabu Search", *AIAA J.* **46**, 701-711.

Suppapitnarm, A., (1998) *A Simulated Annealing Algorithm for Multiobjective Design Optimization*, MPhil Thesis, University of Cambridge.

Suppapitnarm, A., K.A. Seffen, G.T. Parks and P.J. Clarkson (2000) "A Simulated Annealing Algorithm for Multiobjective Optimization", *Eng. Opt.* **33**, 59-85.

White, S.R., (1984) "Concepts of Scale in Simulated Annealing", *Proc. IEEE Int. Conf. Computer Design*, 646-651.