A New, Clustering Evolutionary Multi-Objective Optimisation Technique

A.K. Molyneaux
LENI[*]
adam.molyneaux@epfl.ch

G.B. Leyland
LENI
geoff.leyland@epfl.ch

D. Favrat
Director, LENI

Abstract

This paper introduces a multi-objective EA, termed the Clustering Pareto Evolutionary Algorithm (CPEA). The CPEA finds and retains many local Pareto-optimal frontiers, rather than just the global frontier as is the case of most multi-objective EAs found in the literature. This has been achieved using a clustering technique commonly used in multivariate statistical analysis, which ensures that competition between individuals is local in variable space, allowing the population to grow to resolve as many Pareto-optimal frontiers as necessary. The performance of the CPEA is evaluated on several test problems taken from the literature which have either single optima or multiple local optima and is shown to be extremely effective. The present clustering method is computationally expensive and will be replaced with an incremental method in the near future.

1 Introduction

One of the major area of interest in large, integrated engineering processes is thermodynamic, economic and environmental optimisation. Typically this involves the modelling of a large, integrated system (such as a power station) in terms of both the configuration (choice of machines and sizes) and the operating conditions. The aim is to optimise these models to find a system configuration that will combine good economics with good thermal efficiency and/or minimal environmental impact.

In order to include both economic and environmental criteria (for example pollution), a conventional single-objective optimisation requires that the two criteria be combined. To date, the Laboratoire d'Energétique Industrielle (LENI) has done this by attributing cost factors to each pollutant and then minimising the overall cost of installation, operation and pollution. The choice of these pollution factors poses several difficulties, and has been studied in several research projects, such as [1], where the sensitivity of the optimal solution to changes in pollution costs is dramatic.

The system models used are complex and the parameters and objective functions are often non-linear, non-continuous and disjoint with mixed real and integer variables. The resolution of this class of problem with numerical methods is difficult. EAs provide a robust and effective (if a little slow) alternative.

With most integrated systems engineers would like to have a choice of good solutions covering a range of different technological configurations. Promising results have been obtained [1][2][3] at LENI using a niching EA called the "Struggle GA"[4], developed at MIT. This EA allows real, integer and boolean values to be mixed and tries to keep multiple solutions which are all *good* but which are considered *different* by some user defined distance metric. Ideally the metric represents different regions of the dependent variable domain and hence different system configurations.

The advantage of multimodal optimisation is that the engineer is able to choose from a set of local optima a solution that is perhaps not the *global* optimum of the system defined by the system model and objectives, but which may be the *best* solution of the real system when other, difficult to quantify considerations are taken into account.

Unfortunately, this implementation of the Struggle GA does not tackle the problem of dealing with multiple objectives - hence the need for pollution factors. For environomic optimisation knowledge of the Pareto optimal set of cost/pollution solutions would be ideal. Existing Pareto-optimising EAs tend to eliminate all but the best set of solutions, resulting in only one technological option.

What we want to find is several *local* Pareto-optimal sets where *local* refers to decision variable space - technology type and operating conditions. This implies multiple Pareto-optimal sets in objective space, that may overlap, or where one local set may be entirely dominated by another, that represent different niches in the decision space.

The following sections briefly review existing algorithms, introduce the CPEA, demonstrate its performance on test problems, and discuss directions for future work on the algorithm.

2 Existing Multiobjective Evolutionary Algorithms

A survey of the literature has revealed several approaches to Pareto-EA based optimisation. These are reviewed in [5][6]

[*]Laboratoire d'Energétique Industrielle, Ecole Polytechnique Fédéral de Lausanne, Switzerland

and are described and compared in [7], and this discussion will not be repeated here. However it is worth noting the following points:

- They all appear to use EA schemes with binary coded variables, in contrast to the real-variable coding used by the Struggle GA and the CPEA.
- They all use a population-replacing generational approach, where at each generation an entire new population, is generated from the previous. To avoid losing non-dominated solutions, some algorithms resort to "elitism" or marking the non-dominated solutions as a "special" population that will persist from generation to generation.
- They try to find only the global Pareto-optimal set of solutions for the problem, and tend to lose other local optima.
- Many use the concept of "fitness sharing" in the objective domain to thin the points on the non-dominated surface. Unfortunately, while similar points in parameter space will usually produce similar results in objective space, the converse is not usually true - hence diversity in parameter space is lost.

3 The Clustering Pareto EA

In our algorithm, Pareto set multi-objective optimisation is integrated with clustering concepts commonly used in statistical multivariate analysis. In contrast to the algorithms mentioned above, we use clustering in parameter space to try to preserve local optima, and we use a "breed and die" population control which allows the population to expand over the non-dominated frontier. A detailed description of the algorithm is given below, and discussion follows.

3.1 Description of the Algorithm

- An initial population of p_i individuals is randomly generated and the objectives for each individual are evaluated.
- Distances between every pair of points are calculated.
- A linkage analysis [8] is performed using the distances between the points.
- The solutions are clustered, either using a cutoff value for the inconsistency of the cluster tree lengths[9], or on given number of clusters, n_{cl}.
- The individuals in each cluster are given a rank based on how many other individuals in the cluster dominate or are dominated by the point.
- Each cluster is given the chance to produce n_{ch} children. A single child is created by the following process:
  - A parent is chosen from the cluster in question. The choice is made with a preference for better-ranked parents.
  - A decision is made as to whether the other parent will come from this cluster, or from another.
  - A child is created using a real variable crossover mechanism [4] which provides the search element normally provided by binary crossover and mutation. This creates children in a hypercube defined by the two parents.
  - The cluster which contains the new point is found, and is marked for later "culling".
- In each cluster that has been marked for culling an attempt is made to reduce the cluster size. Several strategies are tried:
  - poorly ranked points are are consecutively marked for removal until the cluster is smaller than a given maximum cluster size, n_{max}.
  - all points with a rank equal to or worse than a r_{max} are marked for removal.
  - if the cluster contains a sufficient number of points in its best rank, its entire worst rank is marked to be removed.
  - if none of the above conditions are met, one randomly chosen point from the worst rank is marked to be removed.
- All the marked points are removed from the population, and the children are added. If the total number of function evaluations is less than n_{ev}, execution returns to the linkage analysis.
- As a final step, the clusters are "cleaned up" - all individuals with a rank worse than one are removed from the population.

3.2 Notes on the CPEA

- For most of the test problems (except Zitzler's t_3), the distances between the points were Euclidian distances in a normalised parameter space. In many cases, however, using only some of the parameters, and/or using some of the objectives, in the calculation of distance may be more appropriate. Ideally, we would like to find an automatic distance measure that works in a large number of cases, however for real problems the distance function will probably have to be defined by the user, as in the Struggle GA.
- Currently, the clustering is a hierarchical Ward clustering performed by the MATLAB function `clusterdata`[9]. This means that a complete clustering algorithm is performed at each iteration, which is very slow. A priority in the development of the algorithm, once

its performance has been proved on large environomic problem will be to implement incremental clustering.

- In our tests, using a fixed number of clusters was far superior to using an inconsistency measure. Unfortunately, this means the user must have *a priori* knowledge about the number of local optima. We would prefer that the clustering algorithm worked without help, and expect that this will be possible with a little work on the clustering.

- The ranking used is *only* local. The "best" point in a locality that is dominated by other clusters has the same rank (1) as the global best point. The ranking currently implemented is that described by Goldberg[10], where the non-dominated are given rank 1 and removed from the rank search, the new non-dominated are given rank 2 and removed and so on until each individual has a rank. Zitzler and Thiele[11] propose a ranking scheme with a higher resolution than that described by Goldberg, and we hope to implement and test this ranking in the near future.

- In one iteration, each cluster can breed several times. Practically, this is because the reclustering and ranking at the beginning of each iteration is very slow. Perhaps ideally, the reclustering could be performed after each birth and death. However, if viewed from the "elitism" point of view, this implies that nearly the entire population is elite, for each (small, single individual) generation. We hope to investigate the effect of the number of children per generation in the near future.

- The algorithm gives an equal chance of breeding to each *cluster*, and not to each individual. This is very important for the performance of the algorithm, as otherwise large clusters (which may have already finished evolving) get far more process time than smaller clusters (which are probably evolving more rapidly).

- The choice of the second parent favours a parent in the same cluster as the first, and small clusters are more likely to chose a local parent than large ones, thus favouring the growth and evolution of small clusters. If the parent is non-local, the parent's cluster is chosen randomly.

- No locally non-dominated points are ever marked for removal, and this means that the population can, and generally does increase. We retain all these points so as to retain information about the non-dominated frontier. Techniques for thinning the non-dominated frontier based on objective space clustering exist [7], but at the moment our goal is to develop an algorithm that can cope with a large population, some of which is close to a local optima, and thus no longer evolving, while other parts of the population are still evolving fast. Hence the equal breeding probability for each cluster, and the importance of the development of an incremental clustering algorithm.

- In all the tests run on the CPEA, r_{max} was 5. We hope to perform tests on the effect of this parameter in the near future. n_{max} was chosen as large as possible while not slowing the algorithm too much. Again, we hope to study its effect on convergence (and not computation time) soon.

- Because more than one point can be removed from a cluster marked for culling, the population can, and occasionally does, decrease. The long term trend, however, is for a growing population.

- In general, after the breeding phase, we expect all clusters to be marked for culling. The *absence* of "to be culled" flag is used to prevent a small cluster that has received no new individuals from having its population reduced.

- The clean-up performed at the end of the CPEA is mainly for the clarity of visualisation of results. If it was wished to continue the simulation from where it ended, it would be better to use the population before clean-up. In most cases it has been observed that the clean-up removes very few points, as the CPEA progresses, most points in the population are non-dominated. Those cleaned-up are usually the worst of the last "generation" of children.

4 Performance on Test Problems

In order to follow the progress of the CPEA, extensive use has been made of problems with two objectives and one or two variables since these may be plotted in two dimensions. In addition the algorithm has been applied to some problems with 30 variables as used by [7]. The results of five problems are presented and discussed in the following section.

4.1 Single Variable, Single Local Optima Test

This is a simple one dimensional test with 2 objectives, as used by Schaffer[12] and used by others [13][7][14].

$$\begin{aligned} \text{Minimise} \quad f_1 &= x^2 \\ \text{Minimise} \quad f_2 &= (x-2)^2 \end{aligned} \tag{1}$$

The problem was solved in the domain $-1000 < x < 1000$ with $p_i = 10$, $n_{cl} = 2$, $n_{ch} = 4$, $n_{max} = 20$ and $n_{ev} = 750$. Figures 1 and 2 show a typical result with 334 individuals distributed over the Pareto-optimal frontier. We have observed that one of the two clusters converges very quickly (within 150 evaluations) to the Pareto-optimal frontier, while the second cluster, favoured for its distance from the first, takes the next 500 or so iterations to converge to the Pareto-optimal frontier. Thus, in this case, the niching behaviour of the

CPEA slows the progress of the *entire* population to the optimal frontier. However, as shown in the following test, niching is necessary where several local optima exist

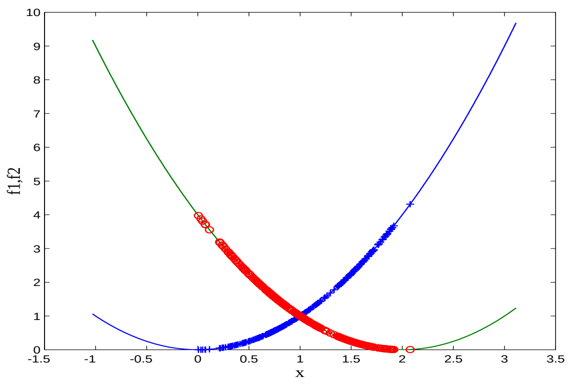

Figure 1: Schaffer's f_1 and f_2 with the Final Population in Variable Space

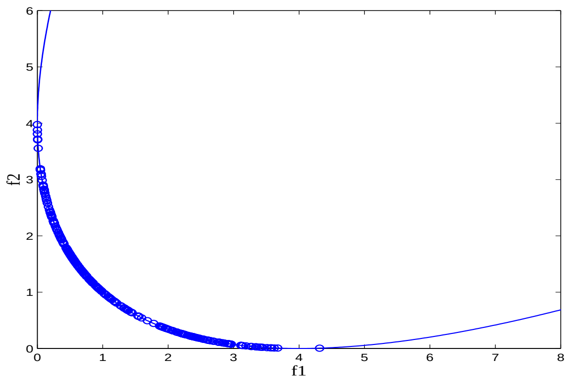

Figure 2: Schaffer's f_1 and f_2 with the Final Population in Objective Space

4.2 Single Variable, Multiple Local Optima

To demonstrate the local optima preserving features of the algorithm, the function below was tested.

$$\begin{aligned} \text{Minimise} f_1 &= \sin(x) * (1 * x/20) \\ \text{Minimise} f_2 &= \cos(x) * (1 * x/20) \end{aligned} \tag{2}$$

This function has three local Pareto-optimal regions in the domain $0 < x < 20$, the rightmost completely dominating the other two. A typical final population from the CPEA ($p_i = 10, n_{cl} = 3, n_{ch} = 4, n_{max} = 10, n_{ev} = 2000$) is shown in Figures 3 and 4. All three local Pareto-optimal frontiers are well defined after only 250 evaluations and remain so even after 2000 evaluations - the dominant frontier does not overwhelm the other two.

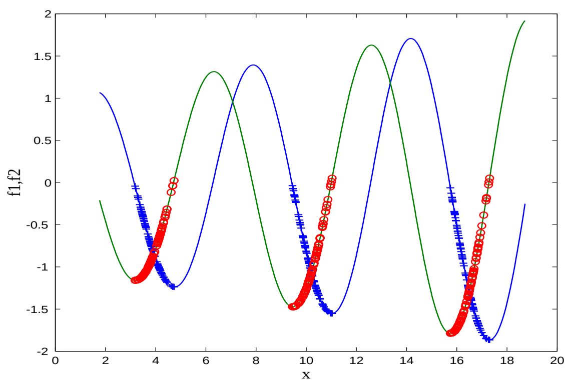

Figure 3: The Sin-Cos Function - Final Population in Variable Space

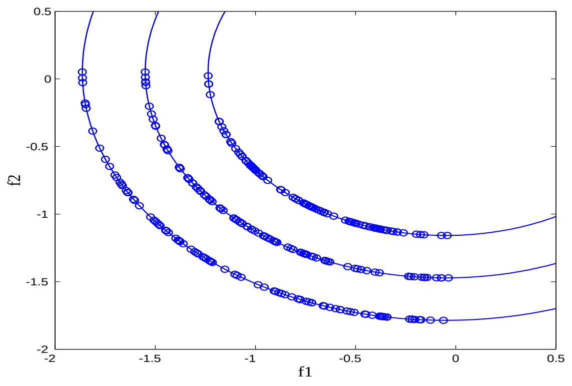

Figure 4: The Sin-Cos Function - Final Population in Objective Space

4.3 Single Optimum Multidimensional Test Problem

Zitzler[7] used a set of test functions based on forms proposed by Deb[15]. These tests take the general form:

$$\begin{aligned} \text{Minimise} \quad \boldsymbol{t}(\boldsymbol{x}) &= (f_1(x_1), f_2(\boldsymbol{x})) \\ \text{subject to} \quad f_2(\boldsymbol{x}) &= g(x_{2\cdots n}) \cdot h(f_1(x_1), g(\cdots)) \end{aligned} \tag{3}$$

The first of these tests, Zitzler's t_1 was run with 30 variables as a comparison with his method.

$$\begin{aligned} f_1(x_1) &= x_1 \\ g(x_{1\cdots n}) &= 1 + \frac{9}{n-1}\sum_{i=2}^{n} x_i \\ h(f_1, g) &= 1 - \sqrt{f_1/g} \end{aligned} \tag{4}$$

The problem has only one, global, Pareto-optimal frontier, and this is attained for a given x_1 when $g = 1$. Because there is only one frontier, the CPEA was run with one cluster. Figure 5 shows all of Zitzler's results for 30 runs with a population of 100 for 250 generations, and all the results of 10 runs of the CPEA ($P_i = 30, n_{cl} = 1, n_{ch} = 10, n_{max} = 200, n_{ev} = 25000$). The final population for the CPEA averaged 630 individuals.

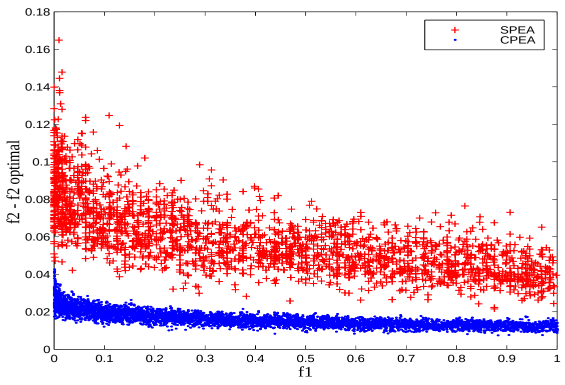

Figure 5: Distance of Final Populations from the Pareto-Optimal Frontier For t_1

From observations the progress of the CPEA it was noted that after a few thousand evaluations nearly all of the points in the population are non-dominated and thus have the same rank. Any "domination-based" ranking is now worthless and the algorithm progresses by breeding along the non-dominated frontier. Once this point has been reached the only way to aid convergence to the Pareto-optimal frontier is to ensure that information about the non-dominated frontier is not lost, and that all computational effort is spent as close to the frontier as possible.

This is achieved by the CPEA's preservation of non-dominated points and growing population - neither the ranking scheme nor the clustering help at all.

Using parameter space clustering for this problem would be to commit the error noted by Spears[16] - preserving diversity for the sake of diversity, not for its appropriateness. Here, it is inappropriate to perform evaluations that are far from the global non-dominated frontier, and thus wasted.

We suspect that the CPEA outperforms the SPEA on this problem because of its very strong elitism. However, it should be noted that many tests were performed on the SPEA, and until we have time to run them all, no conclusions can be drawn about the usefulness of the CPEA's strong elitism.

4.4 Multiple Optima Multidimensional Test Problem

Zitzler's t_3 (Equation 5), unlike t_1, has five local optima, all of which contribute a part of the global Pareto-optimal frontier.

$$\begin{aligned} f_1(x_1) &= x_1 \\ g(x_{1\cdots n}) &= 1 + \frac{9}{n-1}\sum_{i=2}^{n} x_i \\ h(f_1, g) &= 1 - \sqrt{f_1/g} - (f_1/g)\sin(10\pi f_1) \end{aligned} \tag{5}$$

Obviously, t_3 is a good function for testing the clustering, however, clustering on all of $\boldsymbol{x}$ yields very poor results. This is because the $x_{2\cdots30}$ have a much larger combined effect on the distance function than x_1, the only variable with a clear

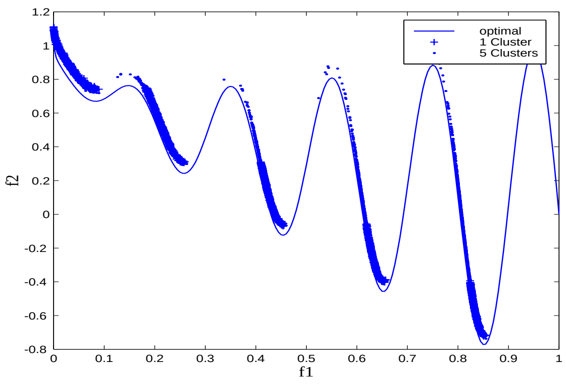

Figure 6: Final Populations for t_3 with One and Five Clusters.

effect on the objectives. Clustering on just x_1 however, gives an interesting result.

The final populations from solving the problem with a single cluster ($P_i = 30, n_{cl} = 1, n_{ch} = 10, n_{max} = 100, n_{ev} = 25000$, 8 runs, average final pop. 330), and with 5 clusters ($P_i = 30, n_{cl} = 1, n_{ch} = 3, n_{max} = 40, n_{ev} = 25000$, 6 runs, average final pop. 560) in x_1 is shown in Figure 6.

Here, it is clear that the single cluster solution only preserves information about the global optimal frontier, while the multiple cluster solution preserves the entirety of each local frontier.

One can also see that the five cluster solution is at least as close to the global optimal frontier as the single cluster solution, and fact, the five cluster solution shown here is *closer* to the optimal frontier. This is despite the fact that the five cluster solution has supposedly "wasted" evaluations that could be used developing the global frontier on developing each local frontier.

This preliminary result would seem to indicate that by using an algorithm like the CPEA, information about entire local frontiers can be obtained with a similar amount of work as would be needed to develop just the global optimal frontier. This is similar to the way a multi-objective EA can be used to obtain information about an entire optimal region, with little more work than it takes a single-objective EA to develop a single optimal point.

4.5 Himmelbau's Function

Himmelbau's function has been used as a niching and multimodal test problem by several authors [17][4]. The function has four equal height peaks, and we test to see if all solutions are found and maintained for a large number of generations. Here, to make a multiobjective problem, f_2 is a second Himmelbau function of half the size.

$$\begin{aligned} f_1(x,y) &= 5 - \frac{(x^2+y-11)^2 + (x+y^2-7)^2}{200} \\ f_2(x,y) &= f_1(\alpha,\beta) \quad \text{where} \quad \alpha = 2x, \beta = 2y \end{aligned} \tag{6}$$

To be considered effective for this problem, an algorithm must not only find all four optima quickly, but it must also

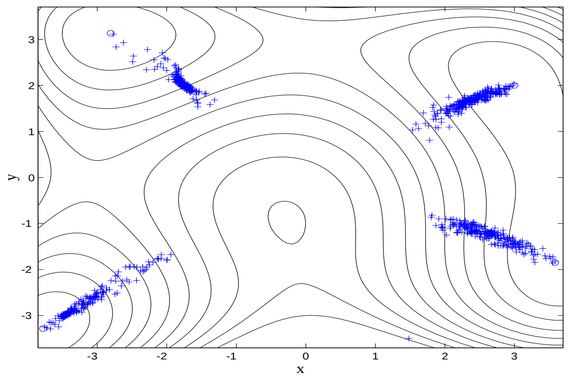

Figure 7: Contours of Himmelbau's f_1 and the Final Population

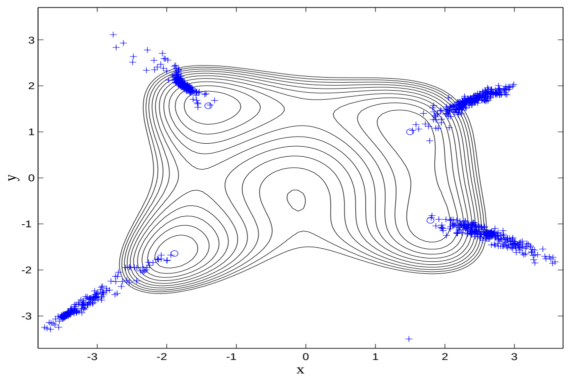

Figure 8: Contours of Himmelbau's f_2 and the Final Population

be able to "keep" those optima for many thousands of evaluations past the point that they are found.

The results of running the CPEA ($P_i = 100, n_{cl} = 5, n_{ch} = 2, n_{max} = 20, n_{ev} = 6000$) are shown in Figures 7 and 8. The contours are from $-5 < f_{(1,2)} < -4$ and the locations of the real optima for each function, as defined by [4] are shown by the circles. Note that the Pareto-optimal frontiers are generally well defined after 1500 iterations, here, we attempt to show that the optima all remain (and become better defined) if the algorithm is left to run long after convergence. When we wish to find many local optima, this kind of stability is very desirable. Here, though we know the function has only four local optima, we choose five clusters so that the clustering algorithm can handle outlying points without losing one of the optima. This is viewed as a weakness of the current clustering algorithm which we hope to rectify in the near future.

5 Conclusions

A new evolutionary algorithm using a clustering technique from multivariate statistical analysis has been presented and its performance evaluated on several test problems taken from the literature. The Clustering Pareto Evolutionary Algorithm (CPEA) has been shown to be as effective as the existing algorithms in finding global Pareto-optimal frontiers.

In addition the CPEA has been shown to find and keep multiple local Pareto frontiers thus providing the designer with multiple different *good* solutions.

6 Future Work

Work is continuing in an attempt to remove the need to specify the number of clusters, to examine the importance of the scaling function, and to find the best variables on which to cluster. An incremental clustering algorithm will be implemented to speed up the algorithm and so allow a more thorough investigation of its behaviour on test cases.

The relationship between "elitist" algorithms, and our "breed and die" approach will be investigated, as will the effect of clustering on the objectives as well as the problem variables or on a reduced set of these.

For the moment, we wish to implement an algorithm that performs well with large populations, but it is possible that in the future we will investigate methods for "thinning" at the non-dominated frontier.

Work has also begun to apply the algorithm to a real environomic problem which has been studied in [1] consisting of approximately 50 independent variables, both real and integer.

Acknowledgement

The authors gratefully acknowledge the support of the Alliance for Global Sustainability, which funds this work as part of the Holistic Design Project.

Bibliography

[1] Curti, V. Von Spakovsky, M. Favrat, D.: *An Environomic Approach for the Modeling and Optimization of a District Heating Network Based on Centralized and Decentralized Heat Pumps, Cogeneration and/or Gas Furnace (Part I: Methodology)*, International Journal of Thermal Sciences, (2000) 39, pp.721-730

[2] Pelster, S. Favrat, D. Von Spakovsky, M.: *Optimisation Thermo-Économique et Environomique de Nouvelles Centrales de Production d'Électricité à Cycles Combinés*, GWA Gas Wasser und Abwasser, 8/2000.

[3] Olsommer, B. Favrat, D. Comte : *Time-Dependent Thermoeconomic Optimization of the Future Waste Incineration Power Plant in Posieux, Switzerland*, 5th World Congress on Integrated Resources Management, Toronto, Canada, 5 June 2000

[4] Grüninger, T.: *Multimodal Optimisation using Genetic Algorithms*, Diploma Report, MIT and Universität Stuttgart, 1996.

[5] Coello, C.: *An Updated Survey of Evolutionary Multiobjective Optimization Techniques : State of the Art and Future Trends*, Labatorio Nacional de Infortmática Avanzada, México.

[6] Fonseca, J., Fleming J.: *An Overview of Evolutionary Algorithms in Multiobjective Optimisation*, IEEE conference on Evolutionary Computation, 1994

[7] Zitzler, E.: *Evolutionary Algorithms for Multiobjective Optimisation: Methods and Applications,* Ph.D. Thesis, ETHZ, Switzerland, 1999.

[8] Arabie, P., Hubert, L., De Soete, G.: *Clustering and Classification* World Scientific, 1996.

[9] Mathworks: *Matlab Reference Manual v5.3*, The Mathworks, 1998.

[10] Goldberg, D. E.: *Genetic Algorithms in Search, Optimisation and Machine Learning*, Reading, MA: Addison-Wesley, 1989

[11] Zitzler, E.,Thiele, L.: *Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach*, IEEE Transactions on Evolutionary Computation, 3(4), pages 257-271, November 1999.

[12] Schaffer, J.: *Multiple Objective Optimization with Vector Evaluated Genetic Algorithms* Genetic Algorithms and their Applications: Proceedings of the First International Conference on Genetic Algorithms, pages 93–100. Lawrence Erlbaum, 1985.

[13] Horn, J., Nafpliotis, N., Goldberg, D.: *A Niched Pareto Genetic Algorithm for Multiobjective Optimisation,* IEEE conference on Evolutionary Computation, 1994

[14] Srinivas, N., Deb, K.: *Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms* Journal of Evolutionary Computation Vol 2 No 3 pg 221-248.

[15] Deb, K.: *Multi-Objective Genetic Algorithms - Problem Difficulties and Construction of Test Problems,* Evolutionary Computation, vol 7, pp. 205-230, 1999.

[16] Spears, W.: *Simple Subpopulation Schemes*, Proc. Evolutionary Programming Conference, 1994,World Scientific, 1994.

[17] Mafhoud, S.: *Niching Methods for Genetic Algorithms*, Ph.D. Thesis, University of Illinois, 1995.