Multiplicity and Local Search in Evolutionary Algorithms to Build the Pareto Front

Héctor A. Leiva, Susana C. Esquivel, Raúl H. Gallard
Proyecto UNSL-338403
Departamento de Informática
Universidad Nacional de San Luis
5700 – San Luis, Argentina
{aleiva, esquivel, rgallard}@unsl.edu.ar

Abstract

In multicriteria optimization determination of the Pareto-optimal front, is of utmost importance for decision making. Simultaneous parallel search for multiple members of an evolutionary algorithm can lead to effective optimization.

In a previous approach [6] extending the ideas of a former work of Lis and Eiben [5], we proposed the multi- sexual-parents-crossovers genetic algorithm (MSPC-GA), a method which by allowing multiple parents per sex and multiple crossovers per mating action attempted to balance the explorative and exploitative efforts, which are present in any evolutionary algorithm. The performance of the method produced an evenly distributed and larger set of efficient points.

Following this concept the present proposal incorporates a hybridisation of global and local search to the multiplicity approach. Now the evolutionary approach combined with simulated annealing and neighbourhood search produced better results.

Keywords: *multiobjective optimisation, evolutionary algorithms, local search, Pareto optimality.*

1 Introduction

Many real world problems, mostly engineering, are characterized by the simultaneous achievement of several goals. Each of these goals corresponds to the optimum of an objective function to be optimized. In most cases, the objective functions are in conflict, being infeasible to optimize any of the objective functions without damaging at least one of the others. This is known as the concept of Pareto optimality [12]. The simultaneous optimization (maximization and/or minimization) of these multiple objective functions constitutes the problem we are trying to solve. But this is not simple. Most of multi-objective optimization problems tend to be characterized by a very large set of admissible solutions, known as the set of Pareto optimal solutions or efficient points. These final solutions conform the so-called Pareto front which is the global optimum of the multi-objective problem considered. Knowledge of the complete Pareto front is very important when search is applied before decision making.

Study of Multiobjective Evolutionary Algorithms (MOEAs) is well established within the Evolutionary Computation field. Since the first work by Schaffer [14] and Fourman [7] many researchers have gone more deeply in this subject ([3, 4, 5, 13, 15, 16, 18, 19, 20, 21]). Most of these studies have been motivated by a suggestion of a non-dominated Genetic Algorithm (GA) outlined in [8] based on the concept of Pareto optimality. The primary reason for these studies is a main feature of GAs: population approach. Since these kinds of EAs work with a population of solutions, multiple Pareto optimal solutions can be found in a population in a single run [4]. At the same time, little research has been done in the field of local search methods applied to multi-objective optimization (e.g. [10]) and combining local search with a population approach (e.g. [11]). In our work a hybrid MOEA based on a non-canonical GA and two local search methods, Simulated Annealing (SA) and Neighborhood Search (NS), is introduced.

Most, if not all, of MOEAs designed and tested so far seek to find the true Pareto front by finding the *best* non-dominated set of solutions. But this does not guarantee that the discovered solutions effectively belong to it. to overcome this problem, post-optimal testing may be performed to establish Pareto-optimality of members already in the non-dominated set [4]. This was the motivation behind the experiments reported in this paper.

In an earlier work [6] some modifications were made to the multisexual genetic algorithm (MSGA) of Eiben and Lis [5] to allow multiple crossover between multiple parents per sex, and MSPC-GA was introduced.

The present paper is involved with improvements of the behaviour of MSPC-GA. Here we attempt to ameliorate performance by locally perturbing the already found solutions in different ways. Local perturbation was achieved by adding local search heuristics in different stages of the current MSPC-GA. So, the performance of this new algorithm called *multi-sexual-parents-crossovers with local search* (MSPC-LS) using testing functions as cited in [14,17] is contrasted here against MSPC-GA.

2 Pareto optimality and multisexual evolutionary algorithms

In an m-objective optimization problem the search space can be seen as an m-dimensional space and therefore each solution is an m-vector of attribute components.

The Pareto criterion simply states that a solution is better than another one if it is so good in all attributes, and better in at least one of these attributes. More formally, let $\prec$ and $\succ$ be the binary relations denoting 'worse' and 'better', respectively, where 'worse' means

$$f_j(x_1) \prec f_j(x_2) \text{ or } f_j(x_1) = f_j(x_2).$$

Then for a maximization problem having more than one objective function (f_j, with $j = 1, 2,\ldots,m$ and $m > 1$), a solution x_1 is said to dominate other solution x_2, if both following conditions hold:

1. The solution x_1 is no worse than x_2 in all objectives; $f_j(x_1)$ *not* $\prec f_j(x_2)$ $\forall$ $j = 1, 2,\ldots,m$ objectives.

2. The solution x_1 is strictly better than x_2 in at least one objective; $f_j(x_1) \succ f_j(x_2)$ for at least one $j \in \{1, 2,\ldots,m\}$.

If any of the above conditions is violated, the solution x_1 does not dominate the solution x_2. But if only the first condition holds and the second solution is not worse than the first then a *conflict* exists between these two solutions and no one dominates each other, they are said to be *non-commesurate*. This happen for instance, when given arbitrary functions f_i, f_k, and $F=\{ f_l \mid f_l \neq f_i, \text{ and } f_l \neq f_k \}$, with $i, k, l \in \{1,\ldots,M\}$) it results $f_i(x_1) \succ f_i(x_2)$, and $f_k(x_2) \succ f_k(x_1)$, and, $f_l(x_2) = f_l(x_1)$, $\forall$ $f_l \in F$.

MSGA as the first multisexual evolutionary approach assigns a sex, to each individual. There are as many sexes as optimization criteria exist. In this way each individual is specialized and tries to fulfil certain optimization criterion. An individual is represented as the corresponding genetic code plus a sex marker, which initially can be randomly set. Consequently the population is divided into subpopulations, each one in accordance to some optimization criterion and individuals are evaluated correspondingly through the appropriate fitness function.

Once individuals are evaluated they are sorted according to their fitness and the rank obtained is the basis for future selection. Ranks are determined independently for each sex.

Recombination is performed as follows: one individual is chosen, as a parent, from each sex and then they undergo *uniform scanning crossover*, which generates a single offspring. Each gene in the child is provided from any of the corresponding genes in the parents with equal probability. The sex of the child is inherited from that parent, which supplied the largest number of genes. If more than one parent supplies to the offspring the same maximal number of genes then the sex of the offspring is randomly chosen from these parent's sexes. Mutation takes place only in the genetic code of the chromosome.

MSPC-GA differs from MSGA in the following:

- uses proportional selection instead of ranking selection.
- selects an equal number of multiple parents per sex, not only one parent as MSGA does.
- uses a multirecombination approach called *multiple crossovers per mating action* (MCPMA), which applies repeatedly uniform scanning crossovers on the selected parents.
- for insertion in the next population, it gives preference to those offspring, which are classified so far, as *globally* non-dominated (belonging to the P_{known} set).

3 The hybrid approach

Referring to Pareto optimality we adopt the notation used by Van Veldhuizen and Lamont in [18]. During a MOEA execution, a "local" set of Pareto optimal solutions is determined at each generation and termed $P_{current}$. This MOEA implementation uses a secondary population termed P_{known} that stores non-dominated solutions found through the generations. Because a solution's classification as Pareto optimal depends upon the context within which it is evaluated, the corresponding vectors of this last set must be periodically tested, removing solutions that became dominated. To reflect the possible changes in membership between generations the variable t is added to represent the completion of t generations. $P_{known}(0)$ is defined as ϕ (empty set) and P_{known} as the *final* set of solutions returned by the MOEA.

Of course, the true Pareto optimal solutions set (termed P_{true}) is not explicitly known for most of problems of any difficulty.

4 Implemented techniques

In what follows when referring to the evolutionary algorithm we adopt a notation similar to that used by Bäck [1,2].

Let μ be the population size, n_0 the number of sexes (objectives), n_1 the number of parents per sex and n_2 the number of crossovers. These are the main parameters of the MSPC-GA. Given $P(t) = (a_1(t), ..., a_\mu(t)) \in I^\mu$ a population of μ individuals at generation t. Then within the cycle creating the next generation, *selection*, *recombination* and *mutation* are described as the following operators

These operators also depend on additional parameters

$$s : I^\mu \to I^{n_0 n_1}, \quad r : I^{n_0 n_1} \to I^{n_2}, \quad m : I^{n_2} \to I^{n_2}$$

Θ_s, Θ_m, *and* Θ_r which are features of the operator and representation of individuals. A termination criterion Θ_t is also defined.

In the sequel, a brief sketch of the MSPC used allowing multiple crossovers on multiple parents per sex, combined with local search is shown.

In what follows this algorithm is called *multi-sexual-parents-crossovers with local search* (MSPC-LS), and returns PF_{known}, the Pareto front set corresponding to P_{known} set.

Let $MP_i\,(t)$ be the set of multiple parents at the i^{th} selection step in generation t to create the next generation t+1,
O_i the set of offspring produced by multiple crossover applied to MP_i,
O_{nond} the subset of these new offspring that are globally nondominated,

MSPC-LS algorithm
begin
Input: μ, n_0, n_1, n_2, Θ_s, Θ_m, Θ_r, Θ_t
Output: PF_{known}

t ← 0
  initialise($P(t)$), the initial population
  evaluate($P(t)$)
  $P_{known}(t+1)$ ← select-nondominated($P(t)$, $P_{known}(t)$)
  while the termination criterion Θ_t does not hold
    $i \leftarrow 0$
    while the new population is created
      $MP_i(t)$ ← select($P(t)$, n_1, Θ_s)   // select $n_1$parents of each sex
      $O_i(t)$ ← recombine($MP_i(t)$, n_2, Θ_r)   // apply n_2 crossover operations over this i^{th} Multiple Parents set
      $O'_i(t)$ ← mutate($O_i(t)$, Θ_m)   // to produce n_2 offspring
(1) $O''_i(t)$ ← Local-Search($O'_i(t)$)
      evaluate($O''_i(t)$)
      O_{nond} ← select-nondominated($O''_i(t)$, $P_{known}(t)$)
      if O_{nond} <> ϕ **then** insert O_{nond} into the new population $P(t+1)$
            else insert $n_2/2$ offspring randomly chosen from $O''_i(t)$ into the new population $P(t+1)$
      $i = i+1$
    end while
    $P_{current}$ ← select-nondominated($P(t+1)$, $P_{known}(t)$)
    $P_{known}(t+1) \leftarrow P_{known}(t) \cup P_{current}$
    $t \leftarrow t+1$
  end while
(2) P_{known} ← Local-Search(P_{known})
  end

Fig. 1: The MSPC-LS Algorithm

As it can be seen in the preceding code, after a multiple parents set is selected and the corresponding operators are applied, the globally non-dominated children, generated from the current multiple parents set, are collected in O_{nond} set. Globally non-dominated, stands for those children that are superior or *non-commesurate* when contrasted against the current solutions already in $P_{known}(t)$. They will be chosen as new alternative solutions. If O_{nond} is not the empty set, then all the solutions belonging to it are copied into the next generation. Otherwise, half of the generated children are randomly chosen to be copied into the next generation. This later insertion strategy will help to maintain genetic diversity.

Instructions numbered (1) and (2) correspond to modifications made to the original MSPC-GA algorithm. In some cases, Local-Search stands for a SA method and in other cases for an NS one. These two local heuristics were either used together or separated, in different cases.

The local search methods were implemented in such a way that they do not only modify the genotypes, and eventually the phenotypes, of the current already found solutions (Pareto optimal or not). In addition to this if, by applying any of these methods, new non-dominated solutions are found (new means that they are not already in $P_{known}(t)$) they are added to the current $P_{known}(t)$ set. The mentioned methods are then applied to the new added solutions too.

5 Experiments

After many initial runs the following experiments were more deeply studied because, according to the preliminary results, best performance was obtained with them. The four following groups of experiments were conducted and their performance evaluated on the functions detailed in the next section:

1. MSPC-GA no local search is considered. Some results are listed in [6].
2. MSPC-LS1: Once MSPC-GA returns P_{known} set, applies a local search (SA) to each solutions in P_{known}. Corresponds to sentence numbered (2) described in the code of the preceding section.
3. MSPC-LS2: Each time one individual has to be evaluated a local search (SA) starts with this individual as the starting point. Corresponds to sentence number (1).
4. MSPC-LS3: After MSPC-LS2 returns the P_{known} set, a local search (NS) is applied to every individual in the set.

5.1 Test Problems and Experimental Results

To evaluate the performance of MSPC-LS the algorithm was tested on the same problem suite used in [6]. They are *typical* multi-objective problems. These problems and the corresponding MSPC-LS parameters are listed bellow.

Problem 1: Srinivas and Deb [17]

Minimize $f_1(x_1,x_2)$ and $f_2(x_1,x_2)$ where

$$f_1(x_1,x_2) = (x_1^2 + x_2^2)^{\frac{1}{8}}$$

$$f_2(x_1,x_2) = ((x_1 - 0.5)^2 + (x_2 0.5)^2)^{\frac{1}{4}}$$

with $-5 \leq x_1, x_2 \leq 10$

Population size:	30
Crossover rate :	0.8
Mutation rate :	0.01
Chromosome length:	32

Problem 2: Schaffer F3 [14]

Minimize $f_1(x)$ and $f_2(x)$ where

$$f_1(x) = \begin{cases} -x & \text{if } x \leq 1 \\ -2+x & \text{if } 1 < x \leq 3 \\ 4-x & \text{if } 3 < x \leq 4 \\ -4+x & \text{if } 4 < x \end{cases}$$

$$f_2(x) = (x-5)^2$$

with $-5 \leq x \leq 10$

Population size:	100
Crossover rate :	0.3
Mutation rate :	0.001
Chromosome length:	16

Problem 3: Schaffer function F2 [14].

Minimize $f_{21}(x)$ and $f_{22}(x)$ where

$$f_{21}(x) = x^2$$

$$f_{22}(x) = (x-2)^2$$

with $-6 \leq x \leq 6$

Population size :	100
Crossover rate :	0.85
Mutation rate :	0.01
Chromosome length:	14

For all versions of MSPC-LS the PF_{known} size was limited to store 1200 efficient points. Experiments were undertaken for diverse values of n_1 and n_2 on each function. In general, good results were achieved for values of n_1 and n_2 between 2 and 4. In this section we report results obtained assigning values $n_1 = n_2 = 3$.

The following figures show relevant results.

MSPC-LS1 versus MSPC-GA

In *Problem 1* after 100 generations 441 non-dominated solutions, were found under MSPC-GA (Fig. 1)and 415 under MSPC-LS1 (Fig. 2), but here a more evenly distribution of points is obtained showing a better delineated Pareto front.

The reduction in the number of efficient points can be explained as follows. Applying MSPC-LS1 the local search continues in the PF_{known} set obtained under MSPC-GA and many of those points (not only the ancestors used in the SA algorithm) become dominated and disappear as PF_{known} is updated.

For the remaining problems the number of non-dominated solutions found was incremented. In *Problem 2* after 600 generations 585 and 798 non-dominated solutions, were found under MSPC-GA (Fig. 4) and MSPC-LS1 (not shown), respectively. In *Problem 3* after 200 generations 583 and 799 non-dominated solutions, were found under MSPC-GA (Fig. 6) and MSPC-LS1 (not shown), respectively

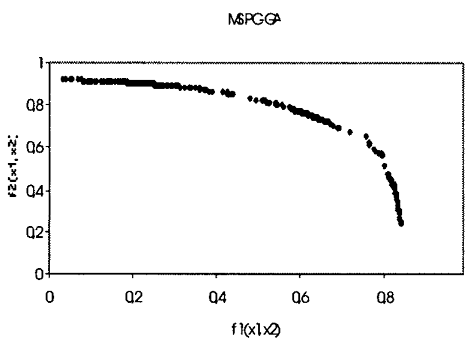

Fig. 2: The Pareto front for Problem 1, MSPC-GA. 441 efficient points

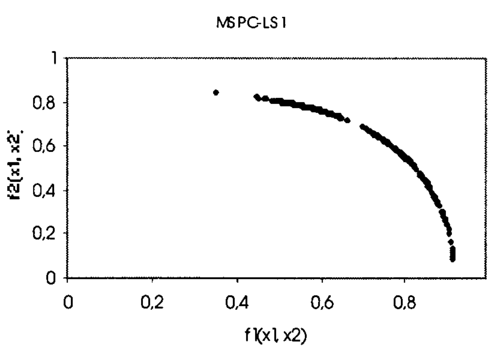

Fig. 3: The Pareto front for Problem 1, MSPC-LS1. 415 efficient points.

MSPC-LS2 versus MSPC-GA

Under MSPC-LS2 the local search (SA) is applied each time an individual is created. For both, Problem 2 and 3 the number of non-dominated solutions reached the maximum storage capacity of 1200.

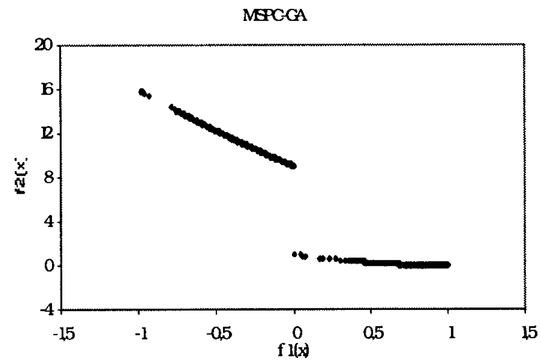

Fig. 4: The Pareto front for Problem 2, MSPC-GA 585 efficient points.

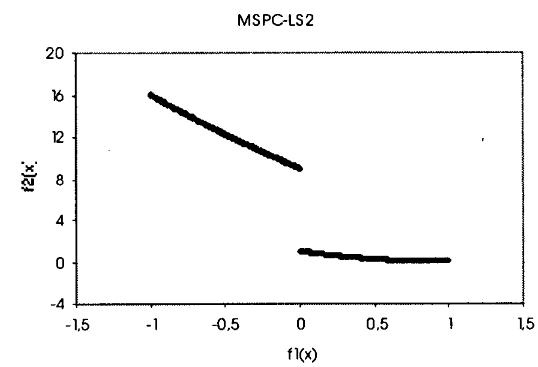

Fig. 5: The Pareto front for Problem 2, MSPC-LS2. 1200 efficient points.

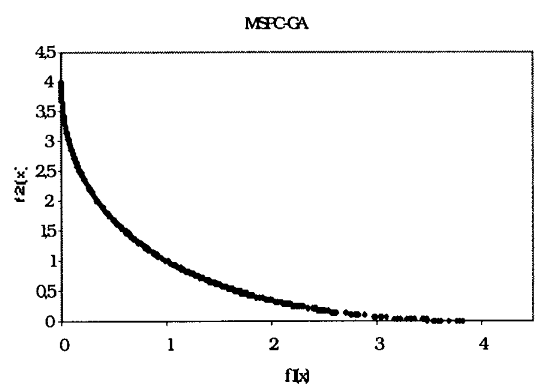

Fig. 6: The Pareto front for Problem 3, MSPC-GA 583 efficient points.

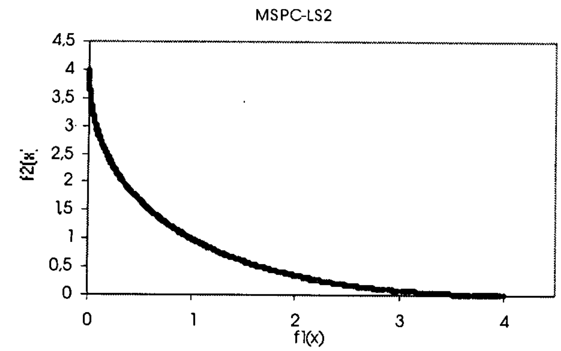

Fig. 7: The Pareto front for Problem 3, MSPC-LS2. 1200 efficient points.

For problem 1 a further reduction in the number of efficient points was observed, only 205 points were found. This limitation of the variants LS1 and LS2 lead us to implement LS3.

MSPC-LS3 versus MSPC-GA

MSPC-LS3 adds a final NS local to each individual of the P_{known} set provided by MSPC-LS2. As a result the algorithm found 1200 efficient points for all three problems (the maximum number allowed). Figure 8 show the resulting Pareto front for problem 1, the hardest of the test suite.

Table 1 summarises the number of efficient points found under each of the discussed algorithms.

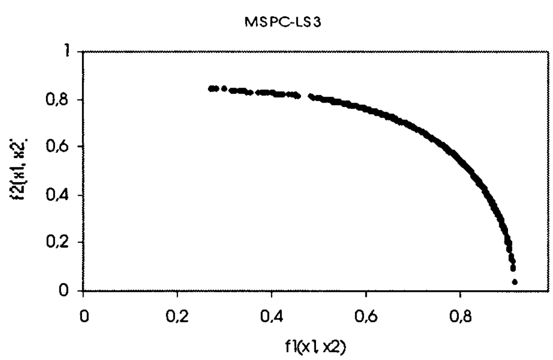

Fig. 8: The Pareto front for Problem 1, MSPC-LS3. 1200 efficient points.

	Problem 1	Problem 2	Problem 3
MSPC-GA	441	585	583
MSPC-LS1	415	798	799
MSPC-LS2	205	1200	1200
MSPC-LS3	1200	1200	1200

Table 1: Results under MSPC-GA and MSPC-LS

6. Conclusions

Recently MSPC-GA, emphasising multiplicity of some of the techniques used in evolutionary computation, was proposed as a new approach to face multicriteria problems. In that work it was allowed multiple parents per sex and multiple crossovers per mating. Parent's selection rewarded for reproduction those best performer individuals. Moreover, offspring selection for replacement favoured those new created individuals that are prone to reside in the Pareto front. If no one fulfilling that condition exits then a random selection is done to maintain genetic diversity.

The present proposal attempts to enhance performance by means of a hybrid approach, MSPC-LS, using both global and local search and maintaining the main characteristics of the earlier proposal.

Three versions of MSPC-LS were implemented. LS1, simply adds a final local search stage to MSPC-GA, using simulated annealing. LS2 applies simulated annealing each time a new offspring is created. LS3, adds a final local search stage to LS2 using neighbourhood search .

When contrasting both methods, current results outperform previous findings with the same parameter settings on the same problem test suite. Distribution of efficient points is now better in all cases. Also, all three variants enlarge the size of the Pareto front for problems 2 and 3. For the hardest problem 1, LS3 is the only one reaching this later goal.

These results are promising and encourage us to deep forward investigation by testing MSPC-LS on harder multicriteria problems to establish the abilities and possible limitations of this approach.

7 Acknowledgements

We acknowledge the cooperation of the project group for providing new ideas and constructive criticisms. Also to the Universidad Nacional de San Luis and the ANPCYT from which we receive continuous support.

8 Bibliography

[1] Bäck T.: Evolutionary algorithms and their standard instances: Introduction. In [2] pp B1.1:1, B1.1:4.

[2] Bäck T., Fogel D., Michalewicz Z.: Handbook of Evolutionary Computation. Oxford University Press, 1997.

[3] Deb K.: Multiobjective optimization using nondominated sorting in genetic algorithms. Evolutionary Computation, 1994.

[4] Deb K.: Multi-objective genetic algorithms: problem difficulties and construction of test problems. Evolutionary Computation, 1999 by the MIT.

[5] Eiben A., Lis J.: A multisexual genetic algorithm for multiobjective optimization. Proc. of 4th IEEE International Conf. on Evolutionary Computation. April 1997.

[6] Esquivel S., Leiva H. A., Gallard R. H.: Multiplicity in genetic algorithms to face multicriteria optimization. Congress on Evolutionary Computation, Washington, D.C., July 1999. IEEE Service Center.

[7] Fourman M. P.: Compactation of Symbolic Layout Using Genetic Algorithms. In [9].

[8] Goldberg D.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison Wesley, 1989.

[9] Grefenstette J. J., editor: Proceeding of the First International Conference on Genetic Algorithms and Their Applications, Hillsdale, New Jersey: Lawrance Erlbaum Associates, Publishers, July 1985.

[10] Hansen M. P.: Tabu Search for Multiobjective Optimization: MOTS. Presented at MCDM'97, Cape Town, South Africa, January 1996.

[11] Knowles J. D., Corne D. W.: The Pareto Archived Evolution Strategy: A New Baseline Algorithm for Multiobjective Optimisation. Congress on Evolutionary Computation, Washington, D.C., July 1999. IEEE Service Center.

[12] Pareto V.: Cours d'Economie Politique. Switzerland, Lausanne: Rouge, 1896.

[13] Savic D. A., Walters G. A.: Genetic Operators and Constraints Handling for Pipe Network Optimization. Evolutionary Computing, Springer Verlag Notes in Computer Science, 1995.

[14] Schaffer J. D., *Some experiments in machine learning using vector evaluated genetic algorithms*, Doctoral dissertation, Department of Electrical Engineering, Vanderbilt University. 1984.

[15] Schaffer J. D.: Multiple Objective Optimization with Vector Evaluated Genetic Algorithms. In [9].

[16] Simpson A. R., Dandy G. C., Murphy L. J.: Genetic Algorithms Compared to other Techniques for Pipe Optimization. Water resource Planning Management, 1994.

[17] Srinivas N., Deb K., *Multiobjective optimization using nondominated sorting in genetic algorithms*, Evolutionary Compt. 2, pp 221-248, 1995

[18] Van Veldhuizen D. A., Lamont G. B.: Multiobjective Evolutionary Algorithm Test Suites. In janice Carrollm, Hisham Haddad, Dave Oppenheim, Barrett Bryant and Gary B. Lamont, editors, proceedings of the 1999 ACM Symposium on Applied Computing, pages 351-357, San Antonio, Texas, 1999. ACM

[19] Van Veldhuizen D. A., Lamont G. B.: Evolutionary Computation and Convergence to a Pareto Front. Late Breaking Papers at the Genetic Programming 1998 Conference, edited by John R. Koza. Satanford, CA: Stanford University Bookstore, July 1998.

[20] Van Veldhuizen D. A., Lamont G. B.: Multiobjective Evolutionary Algorithm Research: A History and Analysis. Technical Report TR-98-03, Air Force Institute of Technology, 1998.

[21] Vemuri R., Cedeño W.: A New Genetic Algorithm for Multiobjective Optimization in Water Resource Management. Proceeding of the 1st IEEE International Conference on Evolutionary Computation. Orlando, USA, 1994.