An Hybrid Optimization Technique Coupling Evolutionary and Local Search Algorithms*

Vincent Kelner[a,†], Florin Capitanescu[b], Olivier Léonard[a] and Louis Wehenkel[b]

[a] *Department of Aerospace, Mechanical and Materials Engineering Sciences*
Turbomachinery Group, University of Liège, Chemin des Chevreuils 1, 4000 Liège, Belgium
[b] *Department of Electrical Engineering and Computer Science*
Stochastic Methods, University of Liège, Montefiore Institute, 4000 Liège, Belgium

Abstract

Evolutionary Algorithms are robust and powerful global optimization techniques for solving large scale problems that have many local optima. However, they require high CPU times, and they are very poor in terms of convergence performance. On the other hand, local search algorithms can converge in a few iterations but lack a global perspective. The combination of global and local search procedures should offer the advantages of both optimization methods while offsetting their disadvantages. This paper proposes a new hybrid optimization technique that merges a Genetic Algorithm with a local search strategy based on the Interior Point method. The efficiency of this hybrid approach is demonstrated by solving a constrained multi-objective mathematical test-case.

Keywords: Nonlinear Programming, Genetic Algorithm, Interior Point Method, Multiobjective Optimization.

1 Introduction

Genetic Algorithms (GAs), initially developed by Holland [5], remain the most recognized and practiced form of Evolutionary Algorithms which are stochastic optimization techniques that mimic the Darwin's principles of natural selection and survival of the fittest.
As a zero-order optimization method, GAs can be used in the case of discontinuous objective functions, within disjoined and/or non-convex design spaces, and together with discrete, continuous or integer design variables. With respect to local search methods (e.g. gradient-based) GAs minimize the risk to converge to a local optimum thanks to the simultaneous processing of the whole candidate solutions.
However, GAs generally require a great number of iterations and they converge slowly, especially in the neighbourhood of the global optimum. It makes thus sense to incorporate a faster local optimization algorithm into a GA in order to overcome this lack of efficiency while keeping advantages of both optimization methods.
The Interior Point Method (IPM) emerged in the middle 50's [4] but was largely developed by Fiacco and McCormick [3]. In the last two decades the IPM gained rightfully more and more

*This work was financed by the RW (Région Wallonne) in the framework of a *Initiative Research Program* (project 215187).

†Corresponding author. E-mail: v.kelner@ulg.ac.be.

attention. First, it clearly outperformed the well-known simplex method in solving (very large) linear programming problems [6]. Then it was successfully applied to solve large nonlinear optimization problems [11].
The salient features of the IPM concern its fast convergence and its adequate handling of inequality constraints by logarithmic barrier functions. Another great advantage of the IP based algorithms is that a strictly feasible initial point is not required, only the non-negativity conditions must be satisfied at each iteration.
We propose in this paper a new hybrid optimization technique that merges a GA with a local search strategy based on the IPM.
The paper is organized as follows. The GA and the IPM are successively introduced in section 2 and section 3. Section 4 presents the proposed coupling procedure. Section 5 yields an illustrative example of the new hybrid optimizer while some conclusions are drawn in section 6.

2 Efficient GA for Complex Optmization Problems

2.1 Mechanics of GAs

GAs work with artificial populations of individuals that represent candidate solutions and, in spite of their diversity, most of them are based on the same iterative procedure (figure 1).
The individuals are characterized by genes, which result from the coding of the parameters of the optimization problem. Each individual is evaluated according to the objectives and to the constraints of the optimization problem. This evaluation is used in the process of selection, which determines the probability that an individual is part of the following generation. Successive new individuals (children) are generated by using the best features of the previous generation (parents) and sometimes, innovating ones. The evolution of those individuals, through the genetic operators, tend to improve the quality of the population and to converge to a global optimum.

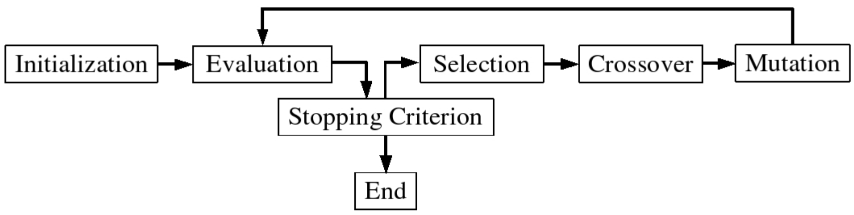

Figure 1: Classical GA flowchart

2.2 GAs and Multi-Objective Optimization Problems

A Multi-objective Optimization Problem (MOP) can be stated as:

$$\min \mathbf{f}(\mathbf{x}) \tag{2.1}$$

where $\mathbf{f}(\mathbf{x})^T = [f_1(\mathbf{x}), f_2(\mathbf{x}), \cdots, f_{nfo}(\mathbf{x})]$. Because of the conflicting nature of the objectives, a MOP has no unique solution, but rather a set of compromised solutions that can be classified, with the Pareto dominance concept, into dominated and non dominated ones. The non dominated solutions representing the best compromise are distributed on the so-called Pareto front. Traditional *a priori* methods [2] are based on a decision phase that transforms the MOP into a single objective one through an aggregating approach (weighted sum, goal programming,

weighted min-max, etc). Unfortunately these techniques lead to a unique optimized solution on the Pareto front (depending on the a priori weights for example).
Conversely, as a powerful *a posteriori* method which works with a population of candidate solutions, GAs solve the true MOP and provide a complete discretization of the Pareto front in a single run. Furthermore, GAs are less susceptible to the shape or continuity of the Pareto front: they can approximate concave or non continuous Pareto fronts, which an aggregating approach does not allow. These advantages have made them very popular to solve *unconstrained* MOPs and numerous Pareto based approaches (MOGA, NSGA, NPGA, etc) have been proposed and compared in the litterature [2, 13].

2.3 The proposed GA

The GA that has been used in this study is based on a previous version of a home-made computer code [7, 8]. It includes the classical genetic operators, and its main features are the following: a real-valued coding for the decision variables, a BLX-alpha crossover [8], a mutation operator, and a Pareto based approach coupled with an original constraint-handling technique.
As we can see in figure 2, the constraints are firstly evaluated for each individual. On the one hand, the feasible solutions are ranked according to the MOGA algorithm proposed by Fonseca and Fleming [2]. On the other hand, each infeasible solution receives a penalty fitness (R_{const}) computed on the basis of the violation of the constraints. At last, a selection, based on a "penalized tournament", is applied. This consists of randomly choosing and comparing the (generally two) individuals :

- if they are all feasible, the best ranked element (according to MOGA) wins,
- if they are all infeasible, the one having the lower R_{const} value wins,
- if one is feasible and the others are infeasible, the feasible individual wins.

This constraint-handling technique, which allows a Pareto dominance approach, has already been validated and successfully applied to design problems [7, 8].

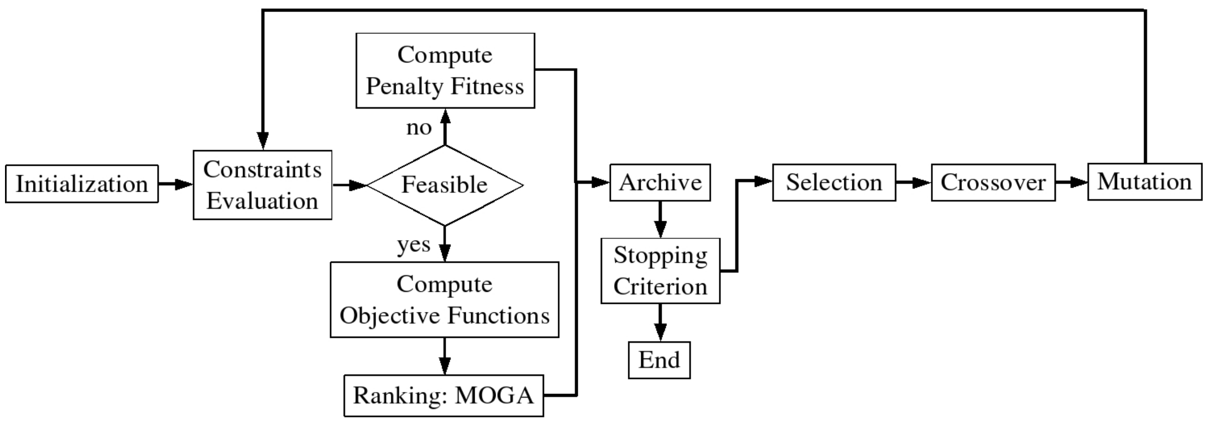

Figure 2: Proposed GA flowchart

Moreover, in the framework of this study, an archiving procedure has been added to the GA. This new operator externally stores the non dominated solutions found at each generation in the following way:

1. Copy all the individuals of the current Pareto front to the archive.
2. Remove any dominated solutions from the archive.
3. If the number of non dominated individuals in the archive is greater than a given maximum N_{arch}: apply a clustering strategy [9].
4. Continue the genetic process (figure 2).

This archiving procedure has been inspired from the SPEA (Strength Pareto Approach) proposed by Zitzler [13]. However, in our implementation, the individuals stored in the archive do not participate to the selection phase, which results in a less disturbed evolution process.
The clustering step (e.g. reducing the size of the archive while maintaining its characteristics) is mandatory: the Pareto front (and the archive) could sometimes contain a huge number of non dominated individuals. However, the designer is not interested in being offered with a too large number of solutions from which he has to choose.

3 Primal-Dual Interior Point Method

In this section we briefly describe the direct primal-dual interior point method for nonlinear programming. A nonlinear programming problem can be compactly expressed in its general form as:

$$\min f(\mathbf{x}) \tag{3.2}$$
$$\text{subject to :} \quad \mathbf{g}(\mathbf{x}) = \mathbf{0} \tag{3.3}$$
$$\underline{\mathbf{h}} \leq \mathbf{h}(\mathbf{x}) \leq \overline{\mathbf{h}} \tag{3.4}$$

where dimension of unknowns vector $\mathbf{x}$, and functions $\mathbf{g}(\mathbf{x})$ and $\mathbf{h}(\mathbf{x})$ are n, m and p respectively. IPM combines three concepts: logarithmic barrier function to handle inequality constraints [4], Lagrange theory of optimization subject to equality constraints [3] and Newton method, which will be illustrated below.
One first transforms the inequality constraints into equality constraints by adding slack variables to inequality constraints:

$$\min f(\mathbf{x}) \tag{3.5}$$
$$\text{subject to :} \quad \mathbf{g}(\mathbf{x}) = \mathbf{0} \tag{3.6}$$
$$\mathbf{h}(\mathbf{x}) - \underline{\mathbf{h}} - \underline{\mathbf{s}} = \mathbf{0} \tag{3.7}$$
$$-\mathbf{h}(\mathbf{x}) + \overline{\mathbf{h}} - \overline{\mathbf{s}} = \mathbf{0} \tag{3.8}$$
$$\underline{\mathbf{s}}, \overline{\mathbf{s}} \geq \mathbf{0} \tag{3.9}$$

where the vectors $\mathbf{x}$, $\underline{\mathbf{s}} = [\underline{s}_1, \ldots, \underline{s}_p]^T$ and $\overline{\mathbf{s}} = [\overline{s}_1, \ldots, \overline{s}_p]^T$ are called *primal variables*.
Slack variables $\underline{s}_i$ and $\overline{s}_i$ $(i = 1, \ldots, p)$ are added to the objective function as logarithmic barrier terms, resulting in the following equality constrained optimization problem:

$$\min f(\mathbf{x}) - \mu \sum_{i=1}^{p} (\ln \underline{s}_i + \ln \overline{s}_i) \tag{3.10}$$
$$\text{subject to :} \quad \mathbf{g}(\mathbf{x}) = \mathbf{0} \tag{3.11}$$
$$\mathbf{h}(\mathbf{x}) - \underline{\mathbf{h}} - \underline{\mathbf{s}} = \mathbf{0} \tag{3.12}$$
$$-\mathbf{h}(\mathbf{x}) + \overline{\mathbf{h}} - \overline{\mathbf{s}} = \mathbf{0} \tag{3.13}$$

where μ is a positive scalar called *barrier parameter* which is gradually decreased to zero as the iteration progresses.
The Lagrangian of the above equality constrained optimization problem is:

$$\mathcal{L}_\mu = f(\mathbf{x}) - \mu \sum_{i=1}^{p} (\ln \underline{s}_i + \ln \overline{s}_i) - \boldsymbol{\lambda}^T \mathbf{g}(\mathbf{x}) - \underline{\boldsymbol{\pi}}^T (\mathbf{h}(\mathbf{x}) - \underline{\mathbf{h}} - \underline{\mathbf{s}}) - \overline{\boldsymbol{\pi}}^T (-\mathbf{h}(\mathbf{x}) + \overline{\mathbf{h}} - \overline{\mathbf{s}})$$

where the vectors of Lagrange multipliers $\boldsymbol{\lambda}$, $\underline{\boldsymbol{\pi}}$ and $\overline{\boldsymbol{\pi}}$ are called *dual variables.*
The *perturbed* Karush-Kuhn-Tucker (KKT) first order necessary optimality conditions of the resulting problem are [3]:

$$\mathbf{F}(\mathbf{y}) = \begin{bmatrix} -\mu\mathbf{e} + \underline{\mathbf{S}}\,\underline{\boldsymbol{\pi}} \\ -\mu\mathbf{e} + \overline{\mathbf{S}}\,\overline{\boldsymbol{\pi}} \\ -\mathbf{h}(\mathbf{x}) + \underline{\mathbf{h}} + \underline{\mathbf{s}} \\ \mathbf{h}(\mathbf{x}) - \overline{\mathbf{h}} + \overline{\mathbf{s}} \\ -\mathbf{g}(\mathbf{x}) \\ \nabla f(\mathbf{x}) - \mathbf{J_g}^T \boldsymbol{\lambda} - \mathbf{J_h}^T (\underline{\boldsymbol{\pi}} - \overline{\boldsymbol{\pi}}) \end{bmatrix} = \mathbf{0} \tag{3.14}$$

where $\underline{\mathbf{S}}$, $\overline{\mathbf{S}}$ are diagonal matrices of slack variables, $\mathbf{e} = [1, ..., 1]^T$, $\nabla f(\mathbf{x})$ is the gradient of f, $\mathbf{J_g}$ is the Jacobian of $\mathbf{g}(\mathbf{x})$, $\mathbf{J_h}$ is the Jacobian of $\mathbf{h}(\mathbf{x})$, $\boldsymbol{\lambda}$ is the vector of Lagrange multipliers associated to equality constraints (3.3) and $\mathbf{y} = [\underline{\mathbf{s}}\ \overline{\mathbf{s}}\ \underline{\boldsymbol{\pi}}\ \overline{\boldsymbol{\pi}}\ \boldsymbol{\lambda}\ \mathbf{x}]^T$.
The perturbed KKT optimality conditions (3.14) may be solved by the Newton method. Let us remark that at the heart of IPM is the theorem [3], which proves that as μ tends to zero, the solution $\mathbf{x}(\mu)$ approaches $\mathbf{x}^\star$, the solution of the problem (3.2-3.4). The goal is therefore not to solve completely this nonlinear system for a given value of μ, but to solve it approximately and then diminishing the value of μ iteratively until convergence is reached.
The outline of the method is as follows:

1. Initialize $\mathbf{y}^0$, taking care that slack variables and their corresponding dual variables are strictly positive $(\underline{\mathbf{s}}^0, \overline{\mathbf{s}}^0, \underline{\boldsymbol{\pi}}^0, \overline{\boldsymbol{\pi}}^0) > \mathbf{0}$. Chose $\mu^0 > 0$.

2. Solve the linear system of equations:

$$\mathbf{H}(\mathbf{y}^k)\, \Delta\mathbf{y}^k = -\mathbf{F}(\mathbf{y}^k) \tag{3.15}$$

   where $\mathbf{H}$ is the Jacobian of KKT optimality conditions (3.14).

3. Determine the step size length $\alpha^k \in (0, 1]$ such that $(\underline{\mathbf{s}}^{k+1}, \overline{\mathbf{s}}^{k+1}, \underline{\boldsymbol{\pi}}^{k+1}, \overline{\boldsymbol{\pi}}^{k+1}) > \mathbf{0}$. Update solution:

$$\mathbf{y}^{k+1} = \mathbf{y}^k + \alpha^k \Delta\mathbf{y}^k \tag{3.16}$$

4. Check convergence. A (locally) optimal solution is found when: primal feasibility, dual feasibility, complementarity gap and objective function variation from an iteration to the next fall below some tolerances.

5. Compute the barrier parameter for the next iteration:

$$\mu^{k+1} = \sigma^k \frac{\rho^k}{2p} \tag{3.17}$$

   where $\rho^k = (\underline{\mathbf{s}}^k)^T \underline{\boldsymbol{\pi}}^k + (\overline{\mathbf{s}}^k)^T \overline{\boldsymbol{\pi}}^k$ is the *complementarity gap*, and usually $\sigma^k = 0.2$. Go to 2.

4 Hybrid Genetic Algorithm

Incorporating a local search strategy into a GA can be done in various ways [12]. In this study, the hybridization technique follows the so-called Lamarckian approach in which the local search algorithm is applied to some newly generated individuals to drive them to a local optimum (figure 3). These locally optimal solutions replace the current individuals in the population to prepare the next generation.

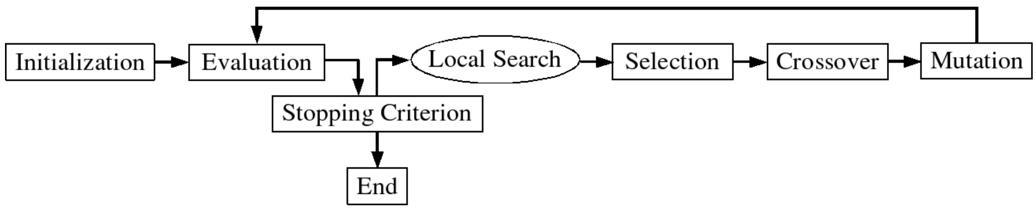

Figure 3: Hybrid GA flowchart

The coupling procedure that has been implemented in the proposed GA can be outlined as follows:

1. Evaluate the n_{nfo} objective functions for each individual of the current population.

2. Locally optimize the n_{pf} non dominated solutions of the current generation. As depicted in figure 4, each i_p^{GA} $(p : 1 \rightarrow n_{pf})$ individual of the Pareto front undergoes the IPM optimizer.
   In order to take into account the multi-objective aspects of the problem, the IPM is applied for one randomly fixed objective function, while maintaining the other functions to a constant value:
   $$\min f_i \qquad \forall i : 1 \rightarrow nfo \tag{4.18}$$
   with
   $$f_j = f_j(\mathbf{i}_p^{\mathrm{GA}}) \qquad \forall j : 1 \rightarrow nfo,\ j \neq i \tag{4.19}$$

3. Discard each i_p^{GA} individual from the population and replace it by its locally optimized element.

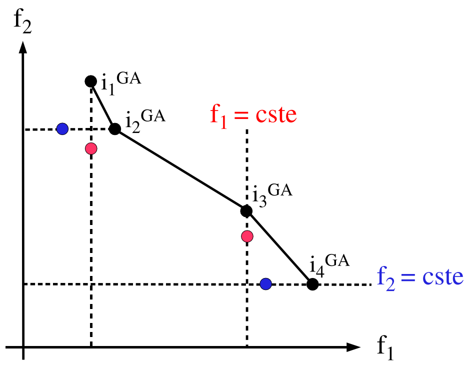

Figure 4: Local optimization of the non dominated solutions

This coupling procedure not only helps the algorithm to escape from being trapped in a local optimum but also seeks out the global optimum with fast convergence.

5 Illustrative Example of the New Hybrid Optimizer

In order to evaluate the efficiency of the previous coupling technique, our hybrid optimizer has been tested on various multi-objective mathematical problems [1, 10]. One relevant test-case is presented here:

$$\text{minimize} \quad f_1(\mathbf{x}) = -x_1 - x_2^2 \tag{5.20}$$
$$f_2(\mathbf{x}) = -x_1^2 - x_2 \tag{5.21}$$
$$\text{subject to} \quad g_1(\mathbf{x}) \equiv 12 - x_1 - x_2 \geq 0 \tag{5.22}$$
$$g_2(\mathbf{x}) \equiv x_1^2 + 10\ x_1 - x_2^2 + 16\ x_2 - 80 \geq 0 \tag{5.23}$$
$$\text{with} \quad x_1 \in [2, 7] \tag{5.24}$$
$$x_2 \in [5, 10] \tag{5.25}$$

First of all, this test-case has been solved by the proposed GA and the IPM, independently (without coupling them). Figure 5 shows non dominated solutions obtained by both methods. For comparison purpose, we also depicted the archived Pareto front.
To be able to deal with this multi-objective optimization problem, the IPM has to transform it into a single objective one. This has been done through the classical weighted sum method, thus resulting in the minimization of the preference function $\Phi(\mathbf{x}) = w_1 f_1 + w_2 f_2$. Furthermore, in order to discretize the Pareto front, the IPM has been launched 11 times with different values for the weights varying from $\{w_1 = 1; w_2 = 0\}$ to $\{w_1 = 0; w_2 = 1\}$ by a constant step. As it was already mentioned (section *2.2*), this technique suffers from the drawback of missing non convex portions of the Pareto front: the IPM has only found 3 non dominated solutions which are located on the extreme sides and close to the middle of the Pareto front (figure 5).
Note that the presence of the middle point on the Pareto front issued of the IPM is intriguing, since theoretically with such method and composite objective function one should converge to the corners of the Pareto front only. A close examination of this anomaly revealed that this point is a local minimum of the problem, obtained for the pair of weights $\{w_1 = 0.4, w_2 = 0.6\}$, while its global optimum corresponds to the right corner of the Pareto front.

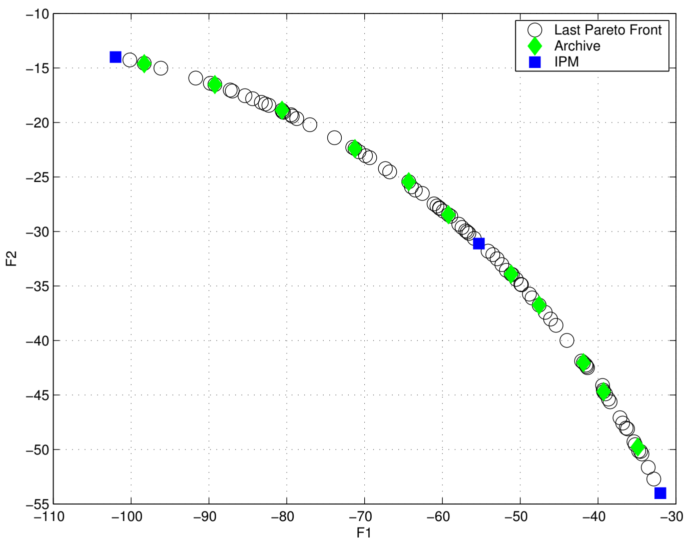

Figure 5: Non dominated solutions - GA vs IPM

On the other hand, the proposed GA (section *2.3*) has been run with a population of $N_{pop} = 100$ individuals during $N_{gen} = 100$ generations (corresponding to 10000 function evaluations). This GA demonstrates its ability to clearly discretize this non convex Pareto front in a single run (figure 5). Moreover, the obtained non dominated solutions are uniformly distributed along the archive (with $N_{arch} = 11$).

In order to reduce the number of function evaluations, the same GA has been tested with only 20 individuals during 10 generations (corresponding to 200 function evaluations). Unfortunately, as illustrated by figure 6, the Pareto front and the archive are not any more correctly identified by this micro GA.

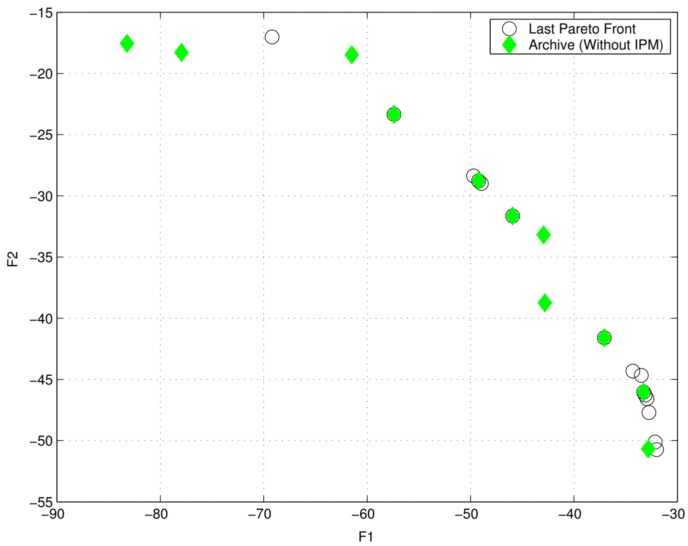

Figure 6: Non dominated solutions - micro GA

Finally, the GA coupled with the IPM has been run with the same previous parameters for the GA ($N_{pop} = 20$, $N_{gen} = 10$, $N_{arch} = 11$). Figure 7 shows that the last Pareto front and the archive obtained by this hybrid optimizer are well discretized, and this only after 200 function evaluations. Furthermore, during the optimization process, the IPM solver was called more or less 50 times. For this test-case, the hybrid approach decreases the computational effort by a factor of 3 with respect to the GA alone, and for a similar quality of the archived Pareto front. This result should be somewhat improved by linking the GA and the IPM in a more efficient way.

Making the parallel between figure 6 and figure 7, one can observe in figure 8 that the proposed coupling approach is efficient. This figure clearly shows the significant improvement of the quality of the archive when using the hybrid approach comparatively with the use of the micro GA alone. This positive effect is due to the fact that the GA is able to explore the promising areas of the design space, involving the global optima at an early stage, while quickly converging to the final non dominated solutions thanks to the IPM. This observation suggests that the major benefit of the coupling is obtained when the GA works with a reduced number of both individuals and generations.

We also mention that a very good exploration of the Pareto front with our hybrid optimizer has been observed for all the studied test-cases.

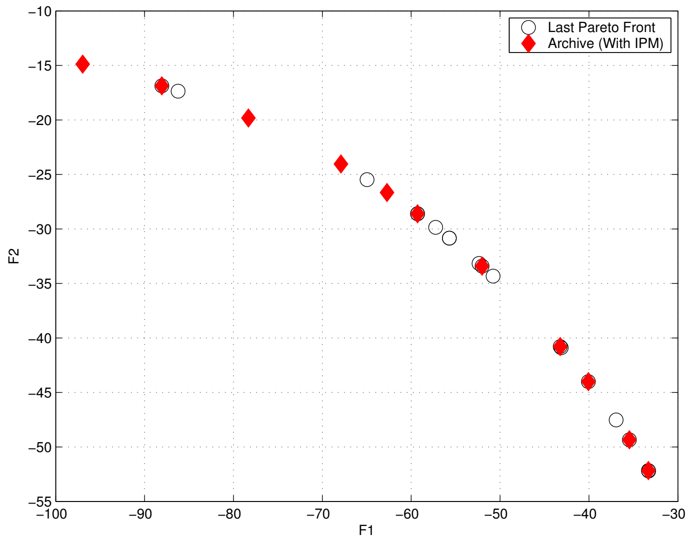

Figure 7: Non dominated solutions - micro GA coupled with IPM

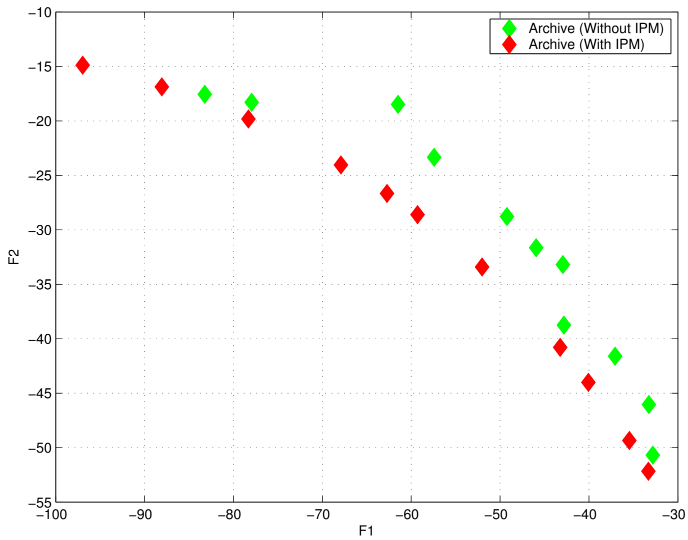

Figure 8: Archive - micro GA vs hybrid optimizer

6 Conclusions

This study presents a promising global optimization technique. Based on a strong coupling approach between a GA and the IPM, a new hybrid optimizer has been developed. This tool exploits the main advantages of both GA and IPM, namely the ability to deal with problems exhibiting several local minima for the former, and the quick convergence to the optimal solution for the latter. Another advantage of this hybrid optimizer with respect to a pure GA is that, for a similar quality of results, it allows the use of a smaller number of function evaluations, and thereby an important reduction of the computational time.

This hybrid optimizer has successfully solved various constrained and unconstrained multi-objective optimization problems.
Work in progress concerns the finding of a satisfactory criterion to assess the performances of the proposed coupling approach. A future work aims to validate this hybrid optimizer on real-life engineering optimization problems (e.g. issued from mechanical engineering and power systems).

References

[1] R.F. Coelho. *Multicriteria Optimization with Expert Rules for Mechanical Design.* PhD Thesis, Université Libre de Bruxelles, 2004.

[2] C. Coello, A Comprehensive Survey of Evolutionary-Base Multiobjective Optimization, *Knowledge and Information Systems* **1(3)** (1999) 269–308.

[3] A.V. Fiacco, G.P. McCormick, *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*, John Willey & Sons, 1968.

[4] K.R. Frisch, The Logarithmic Potential Method of Convex Programming, *Manuscript at Instiute of Economics*, University of Oslo, Norway, 1955.

[5] J. Holland, *Adaptation in Natural and Artificial Systems*, The University of Michigan Press, Ann Arbor, 1975.

[6] N. Karmarkar, A New Polynomial-Time Algorithm for Linear Programming, *Combinatorica* **4** (1984) 373–395.

[7] V. Kelner, O. Léonard, Optimal Pump Scheduling for Water Supply Using Genetic Algorithms. In: G. Bugeda, J.-A. Désidéri, J. Periaux, M. Schoenauer, G. Winter (eds.), *Proceedings of the Fifth International Conference on Evolutionary Computing for Industrial Applications*, Barcelona, Spain, 2003.

[8] V. Kelner, O. Léonard, Application of Genetic Algorithms to Lubrication Pump Stacking Design, *Journal of Computational and Applied Mathematics* **168** (2004) 255–265.

[9] J.N. Morse, Reducing the Size of the Nondominated Set: Pruning by Clustering, *Computers and Operations Research* **7(1-2)** (1980) 55–66.

[10] D. Van Veldhuizen, G. Lamont. *Multiobjective Evolutionary Algorithm Research: A History and Analysis.* Technical Report TR-98-03, Department of Electrical and Computer Engineering, Graduate School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, Ohio, 1998.

[11] S.J. Wright *Primal-Dual Interior-Point Methods*, SIAM, 1997.

[12] Y.G. Xu, G.R. Li, Z.P. Wu. A Novel Hybrid Genetic Algorithm Using Local Optimizer Based on Heuristic Pattern Move, *Applied Artificial Intelligence* **15** (2001) 601–631.

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