The Influence of Optimization Algorithm on Suction Muffler Design

Talita Wajczyk
rodrigo_link@embraco.com.br

Rodrigo Link

Follow this and additional works at: http://docs.lib.purdue.edu/icec
The Influence of Optimization Algorithm on Suction Muffler Design

Talita WAJCZYK1, Rodrigo LINK2*

EMBRACO, Research & Development Group, Joinville, Santa Catarina, Brazil
1Talita_Wajczyk@embraco.com.br
2Rodrigo_Link@embraco.com.br

*Corresponding Author

ABSTRACT

The standards for refrigerators energy consumption are increasing faster in the recent past years, so, the situation demands compressor with a higher efficiency level. To develop such compressors the performance of the most important compressor’s components must be increased. However, the pressure for fast time to market are turning more difficulty the optimization process of a big amount of components, specially using experimental validations.

One of the most important components in the compressor is the suction muffler. The main suction muffler functions are the thermal insulation of gas from the evaporator and noise attenuation. To increase the performance of muffler is mandatory to modify the length and diameter of tubes, the geometry and volume of chambers. During the muffler development is common the necessity of modifications in at least six parameters, and is easy to verify that, with this number of parameters is almost impossible to find the optimized solution only using the iterative method, so, it is necessary to use some algorithm to optimize all these parameters to find the best solution for the suction muffler.

This paper presents a series of suction muffler optimization, using in house computational code for muffler simulation and the commercial code modeFRONTIER to optimize the objective functions. The optimizing algorithms selected to this task are Non-dominated Sorting Genetic Algorithm II (NSGA-II), Multi-Objective Game Theory (MOGT) and Multi-Objective Genetic Algorithm II (MOGA-II). Using the same variables and objective functions for all algorithms, the performances of algorithms are evaluated to define the best strategy to optimize the muffler.

1. INTRODUCTION

The refrigeration area is becoming more and more relevant for human being’s life. Every day new applications are presented to the market for medical, food or technology segment. Considering this, higher levels of energy efficiency standard are being demanded, specially for commercial and household refrigeration systems. In order to attend these requests, a multi disciplinary analysis should be done to improve the performance of each compressor system, where thermodynamics, mechanical and electrical areas are mainly involved.

The interaction between pulsation, superheating and valve dynamics occurred at suction system is highlighted as relevant for compressor efficiency. Not just important for volumetric efficiency and power consumption but also for acoustic. Therefore, accessing data of suction chamber pulsation, valve movement, piston displacement and cylinder pressure is the starting point for its optimization. Many techniques allow obtaining these signals by experimental tests or numerical analysis. Under simulation way, 1D, 2D and 3D differential methodologies are available.

For 1D and 2D methodologies, the mathematical model has the integral formulation for compression process inside cylinder regarding the first law of thermodynamics and the suction muffler is solved by finite volume method, for more details read Ussyk (1984), Todescat et al. (1992) and Deschamps et al. (2002). The 3D model, however, evaluates real geometry of chambers and valves with commercial CFD code base on finite volume. For more details the authors recommended the work of Pereira (2006). Takemori (2010) presents a fluid-structure interaction approach, which is a more complex 3D methodology that allows the whole compressor simulation. This methodology takes into account the forces over valves, calculated in CFD code, to simulate the structural behavior,
using a FEM code, and evaluates the valve displacement. Regarding this, each method will afford different levels of details and computational cost, what should be considered when the proposal is optimization.

The Figure 1 presents the numerical predictions of suction chamber pulsation and valve movement for 1D, 2D and 3D models, respectively. The 1D model does not have a very accuracy prediction of gas pressure drop during suction process impacting over suction valve dynamics. Even with meaningful distinction between data, 1D and 2D formulations are simpler, less computationally expensive and adequate for preliminary design. Generally, 1D is ultimately selected as muffler model for preliminary design.

![Figure 1: Numerical prediction for 1D, 2D and 3D models: a) suction pressure pulsation and b) valve dynamics.](image)

Fast responses with relative quality are important characteristics for optimization process due to an amount of parameters that are evaluated. Otherwise, industrial applications would be restricted. The motivation behind this tool is the lower time of simulation, reduction of cost with prototype components, quantity and quality of results when compared to experimental tests.

Techniques such as design of experiments, DOE, are useful to evaluate the impact over compressor performance for some parameters, but it does not guarantee the solution space is well explored and the best result is achieved. In the other way, the optimization algorithms allow the problem exploration in an effective way. Many of them were developed according to biological principals and the other ones based on different theories: the genetic algorithm, considers evolutionary theory; the swarm optimization is motivated on social behavior, the ant colony optimization, which has as reference on ant food collection and the MOGT, inspired by game theory.

The main proposal of this paper is to analyze the affects over Pareto frontier on suction muffler optimization of reciprocating compressors with two objectives. The objective functions elected are cooling capacity and coefficient of performance, COP, as a minimization-maximization problem. Another arrangement could be done, but generally for compressors problems the configuration stabilized guarantees an expressive Pareto frontier. Three algorithms were selected, the reasons of being sorted out with the respectively comparison will be described on next sections of this paper. The compressor model simulated uses R134a as refrigerant, has 6.75cm$^3$ of displacement and achieves 262W of cooling capacity for -23.3°C / 40.5°C test condition. The muffler is composed by two chambers and two tubes as shown on Figure 2. Eight geometric variables will be parameterized, being four of them the diameters and the lengths of the tubes.
2. MATHEMATICAL MODEL

Sets of Pareto frontier will be produced when the optimization process finishes and, at this time, metrics will quantify the algorithm performance. Considering the limitation that the true Pareto frontier is unknown the following three metrics, presented by Lee et al. (2005), will be applied: i) size of the dominated space; ii) coverage of two Pareto frontiers and c) non-uniformity of the Pareto frontier. The modeFRONTIER code will be responsible to define initial population and to start the run while a black box with compressor model is going to solve the simulation.

2.1 The Compressor Model

The compressor model, the black box, is a code that uses an integral formulation to simulate the complete reciprocating compressor. The mass, momentum and energy equations are numerically solved for compressor suction and discharge chambers by Finite volume method. The valves are modeled as one-degree-of-freedom system and the effective force and flow area determine the mass flow through valves. The piston position is obtained by specific equation for crankshaft mechanism, shaft and eccentric bearings are formulated as short bearing and motor efficiency is acquired from motor curve. For more details see Ussyk (1984).

2.2 Algorithms Description

According to Serapião (2009), the evolutive algorithms class, generally named EA, has been extremely studied and often required to solve scientific and engineering problems by reason of simplicity, robustness and flexibility. They act over a possible population of solutions through the individual diversity principle, the survival of the stronger and the well-adapted to the environment. Genetic concepts are imitated by operators, which are responsible to reproduce stronger descendant. The genetic algorithms belong to this class, such as Non-Dominated Sorting Algorithm II (NSGA-II) and Multi Objective Genetic Algorithm II (MOGA-II).

The first method is widely applied to generate the Pareto frontier and the offspring is created by crossover and mutation operators, more details see Kanagarajan et al. (2008). Figure 3 shows the NSGA-II and the MOGA-II algorithms main loop, the solution is classified into non-dominated frontiers followed by crowded tournament. This last operator returns to the winner regarding to non-dominated front and crowding distance. As new populations are formed, operators will act over them and provide a new generation and, finally, non-dominating sorting ranks the individuals based on dominance, while crowding distance, defined as perimeter of the cuboid formed by the nearest neighbors as the vertices, estimates the density of solutions in the objective space.

MOGA-II was developed considering standard genetic algorithms and has one-point and directional crossover, mutation and selection as operators for reproduction. On one-point crossover, portion of genetic material is exchange between parents. During the directional crossover, the fitness value of an individual is compared with its parents from previous generation. The new individual, then, is created by a randomly weighted direction moving between original individual own direction and its parents. For selection method, since the first candidate to reproduce is selected from a list of all possible individual that is prepared and the candidate with best fitness is chosen.

The mutation ensures diversity to next generation and robustness to the algorithm. A level of perturbation on binary string is applied by mutation operator. The efficiency of this algorithm is also ruled by elitism which the main proposal is preserving the individuals that are closest to the Pareto frontier and the ones that have the best dispersion. On reproduction process one operator will be applied according probability settings for each individual. The requested number of parents to compose the next population will be randomly extracted from union of children and elitism sets.
Multi Objective Game Theory (MOGT) is the last algorithm analyzed in the present paper. The objective of comparison between genetic algorithms and MOGT is to understand the impact over suction muffler optimization when a simpler method is selected. This algorithm is inspired by game theory and has been found its application in social science and economics problems. Clarich et al. (2004) define MOGT as the most recommended algorithm for its highly constrained and non-linear problems and for being able to find non-dominated solutions with few computations if compared to other multi-objective algorithms.

Pareto frontier is a result of both competitive and collaborative game. Players represent the objectives while variables of optimization problem are distributed among players at the beginning. The initial decomposition is random but during the game it is changed according to statistical analysis. Each player is forced to solve its own objective function and variable by Simplex Algorithm with all others players’ variables fixed. At the end of Simplex iterations the players results are constrained by the others ones and the fixed variables values are updated. The game will continue until the defined maximum number of player steps is reached, as shown in Figure 4.

2.3 Metrics of Algorithm Performance
The aim of multi-objective optimization is to find a single solution giving the best compromise between multiples objectives. According to Zitzler (1999), the quantity of metrics required to evaluate the performance of an optimization problem should be at least equal to the number of objectives. The performance assessment of multi-objective algorithms will be described below.
2.3.1 Size of the dominated space: The size of the dominated space \( S(A) \) was introduced by Zitzler and Thiele (1998) and estimates the quantity of objective space dominated by a given non-dominated set \( A \). For an optimization which objective one is maximization and objective two is minimization, \( S(A) \) is the area below solution as presented in Figure 5. Higher values of \( S(A) \) indicate better performance.

![Figure 5: Space dominated by a given Pareto frontier.](image)

2.3.2 Coverage of two Pareto fronts: The Coverage of two Pareto fronts metric was also introduced by Zitzler and Thiele (1998) and shows that the outcomes of one algorithm dominate the outcomes of another algorithm, although it does not tell how good it is. When two Pareto sets are given \( A \) and \( B \), the coverage, \( C(A,B) \), maps the ordered pair \((A,B)\) to the interval \([1,0]\).

\[
C(A,B) = \frac{|\{b \in B | a \in A: a \succ b\}|}{|B|} \tag{1}
\]

where \(|B|\) means the number of solutions in the set \( B \), and \( a \succ b \), that the solution \( a \) dominates solution \( b \). Furthermore, \( C(A,B) \), gives the fraction of \( B \) dominated by \( A \). When \( C(A,B) = 1 \), all the individuals in \( B \) are dominated by \( A \); if \( C(A,B) = 0 \), then no individual in \( B \) is dominated by \( A \). \( C(A,B) \) is not necessarily equal to \( 1 - C(B,A) \). If \( C(A,B) > C(B,A) \), this means that the set \( A \) has better solutions than \( B \). For more details see Lee et al. (2005).

2.3.3 Non-uniformity of Pareto front: Non-uniformity of Pareto frontier \( D(A) \) measures how uniformly the individuals are distributed in Pareto optimal set \( A \). \( D(A) \) is given by the distribution of Euclidian distance, \( d_i \), between two consecutive points of Pareto frontier. This metric is a standard deviation of distances normalized by the average distance \( \bar{d} \). If \( D(A) = 0 \), the distribution is uniform, therefore, lower value of \( D(A) \) is desired.

\[
D(A) = \frac{\sum (d_i/\bar{d} - 1)^2}{|A| - 1} \tag{2}
\]

2.4 Parameters Settings

DOE methods selected to create initial population for genetic algorithms were random and incremental space filler, named ISF, with fifty and ten individuals, respectively. Others methods are available at modeFRONTIER and were tested, but considering suction muffler optimization response and computational time, initial population size with sixty individuals was suitable. MOGT takes into account just the first point generated by DOE.

The parameters of optimization algorithms were defined by sensibility tests in order to guarantee higher performance. Each operator was evaluated with at least two levels upper and lower than the default value. The Table 1 below indicates the parameters setting used to solve the optimization problem. Twenty independent runs
were performed with different seeds and genetic algorithms were kept equal from both the number of generations and size of initial population to allow future comparison.

### Table 1: Parameter settings of NSGA-II, MOGA-II and MOGT.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Population initial size</th>
<th>Number of generations</th>
<th>Crossover probability</th>
<th>Probability of directional crossover</th>
<th>Probability of selection</th>
<th>Probability of mutation</th>
<th>Maximum number of players</th>
<th>Significance threshold of variable acceptance</th>
<th>Simplex maximum number of iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td>NSGA-II</td>
<td>60</td>
<td>100</td>
<td>0.90</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>60</td>
</tr>
<tr>
<td>MOGA-II</td>
<td>60</td>
<td>100</td>
<td>-</td>
<td>0.25</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>100</td>
</tr>
<tr>
<td>MOGT</td>
<td>1</td>
<td>100</td>
<td>-</td>
<td>-</td>
<td>0.05</td>
<td>0.20</td>
<td>100</td>
<td>0.85</td>
<td>discrete</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

3. RESULTS

Twenty independent runs with different random seeds were performed with NSGA-II, MOGA-II and MOGT and the quality of Pareto frontier obtained is measured by three metrics. The Table 2 resumes the mean and standard deviation of $S(A)$ in $W$. The NSGA-II and MOGA-II have higher values of $S(A)$, 81.7$W$ and 78.8$W$, respectively; denoting better results are achieved with genetic algorithms. The upper and lower values reached by NSGA-II and MOGA-II are closest with $S(A)$ between 73$W$ and 96$W$. The dominated space of NSGA-II is 3.7% higher than MOGA-II and 42.4% than MOGT. However, the difference among NSGA-II and MOGA-II mean is smaller than standard deviation. The magnitude of MOGT standard deviation evidences the performance is statistically more instable and this fact is proved by maximum and minimum value of $S(A)$. Algorithms that present huge range of $S(A)$ for the configuration stabilized suggest strong dependence of initial individual or population. Moreover, the progress is slow and the non-dominated set is restricted to a local region of space solution.

### Table 2: $S(A)$ metric for NSGA-II, MOGA-II and MOGT.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Mean</th>
<th>Max. Value</th>
<th>Min. Value</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>NSGA-II</td>
<td>81.71</td>
<td>96.41</td>
<td>73.77</td>
<td>6.24</td>
</tr>
<tr>
<td>MOGA-II</td>
<td>78.79</td>
<td>96.32</td>
<td>73.58</td>
<td>5.80</td>
</tr>
<tr>
<td>MOGT</td>
<td>57.38</td>
<td>98.04</td>
<td>20.78</td>
<td>17.90</td>
</tr>
</tbody>
</table>

The distribution on Pareto front are evaluated by non-uniformity metric $D(A)$. According to Table 3, where small value suggests uniform spread of solution, MOGT shows better uniformity, 14.2% and 36.7% smaller when compared to MOGA-II and NSGA-II, respectively. Although MOGT has superior $D(A)$, the $S(A)$ is extremely low and the solutions were almost dominated by genetic algorithms.

### Table 3: $D(A)$ metric for NSGA-II, MOGA-II and MOGT.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Mean</th>
<th>Max. Value</th>
<th>Min. Value</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>NSGA-II</td>
<td>1.71</td>
<td>4.28</td>
<td>0.98</td>
<td>0.79</td>
</tr>
<tr>
<td>MOGA-II</td>
<td>1.43</td>
<td>4.00</td>
<td>0.88</td>
<td>0.73</td>
</tr>
<tr>
<td>MOGT</td>
<td>1.25</td>
<td>2.10</td>
<td>0.81</td>
<td>0.35</td>
</tr>
</tbody>
</table>

Coverage of two Pareto frontiers is a metric that assess the fraction of Pareto front $B$ dominated by set $A$. As presented in Table 4 MOGA-II dominates 40.1% of NSGA-II and 91.5% of MOGT. 92.6% of MOGT solution set is dominated by NSGA-II. MOGT had generated weak non-dominated set when analyzed with present genetic
algorithms. \( C(A, B) \) is a complementary metric and is very useful when \( S(A) \) of two algorithms are similar. This situation occurred with NSGA-II and MOGA-II where the \( S(A) \) is not statistically distinct. Once again, the absolute difference between MOGA-II and NSGA-II on \( C(A, B) \) metric is smaller than standard deviation.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>Mean</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOGA-II</td>
<td>NSGA-II</td>
<td>0.40</td>
<td>0.16</td>
</tr>
<tr>
<td>MOGA-II</td>
<td>MOGT</td>
<td>0.92</td>
<td>0.14</td>
</tr>
<tr>
<td>NSGA-II</td>
<td>MOGA-II</td>
<td>0.36</td>
<td>0.15</td>
</tr>
<tr>
<td>NSGA-II</td>
<td>MOGT</td>
<td>0.93</td>
<td>0.12</td>
</tr>
<tr>
<td>MOGT</td>
<td>NSGA-II</td>
<td>0.00</td>
<td>0.01</td>
</tr>
<tr>
<td>MOGT</td>
<td>MOGA-II</td>
<td>0.00</td>
<td>0.01</td>
</tr>
</tbody>
</table>

Table 4: \( C(A, B) \) metric for NSGA-II, MOGA-II and MOGT.

Six thousands of individuals were solved by each genetic algorithm. On average, 317 and 312 solutions of MOGA-II and NSGA-II, respectively, belong to Pareto frontier. The mean and standard deviation of number of solutions in non-dominated set are plotted in Figure 6. According to the graphic, just 10% of genetic algorithms Pareto frontier compose the MOGT set. When mean results are evaluated MOGA-II has 5 solutions more than NSGA-II. This quantity is not significant if information about standard deviation is available.

In Figure 7 are plotted MOGA-II, NSGA-II and MOGT Pareto fronts after twenty independent runs. The Pareto frontier was constructed by selection of non-dominated solutions when all runs have put together. For cooling capacities above 225W, all algorithms have similar behavior. MOGT appears to have difficult to minimize the cooling capacity below 215W, and to maximize COP between 215W and 225W range. Just at the region below of 215W, NSGA-II demonstrates to find out better COP solutions.

The MOGT algorithm has obtained 57.38W of \( S(A) \) with high standard deviation, 17.70. The mean number of solutions at Pareto front is low with 31 solutions. However, after twenty runs, the Pareto front is quite similar to genetic algorithms analyzed by this paper, as shown in Figure 7. As earlier mentioned, algorithms that presents huge range of \( S(A) \) for the configuration stabilized suggests strong dependence of initial individual or population. This fact can be verified when two rounds of MOGT are plotted together, as in Figure 8. MOGT uses just one individual as initial population.

Due computational time importance for industrial applications is necessary to mention the algorithms performance related with this topic and in Table 5 is presented the average of optimization time for each algorithm. The average of simulation time of MOGA-II and NSGA-II are similar while MOGT optimization process is faster. It is important to note that simulation time is 20 time faster for MOGT algorithm. This difference became the MOGT a good option to optimization of suction muffler during the product development process in industrial applications.

\( \text{International Compressor Engineering Conference at Purdue, July 16-19, 2012} \)
Table 5: Optimization process time for NSGA-II, MOGA-II and MOGT.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Optimization process time [hr]</th>
</tr>
</thead>
<tbody>
<tr>
<td>NSGA-II</td>
<td>20.4</td>
</tr>
<tr>
<td>MOGA-II</td>
<td>20.6</td>
</tr>
<tr>
<td>MOGT</td>
<td>0.8</td>
</tr>
</tbody>
</table>

Figure 7: MOGA-II, NSGA-II and MOGT Pareto fronts after twenty independent runs.

Figure 8: Two rounds of MOGT with different initial individual.

4. CONCLUSIONS

The present paper analyzed the performance of three different algorithms in a suction muffler optimization process. The performance of optimization algorithms was evaluated using three metrics specially designed to evaluate the algorithms in maximum-minimum problems, as the suction analysis performed in this work.
Analyzing the metrics is possible to verify a similarity of NSGA-II, MOGA-II and MOGT, as differences are possible to point strong influences of initial population for the MOGT algorithm, situation which is not perceptive to the genetic algorithms, furthermore, the computational time is another advantage of genetic algorithms, for example, the NSGA-II calculates 2.7 times more solutions per minute than the game theory algorithm.

**NOMENCLATURE**

- \( S(A) \): size of the dominated space (W)
- \( C(A,B) \): coverage of two Pareto fronts (-)
- \( D(A) \): non-uniformity Pareto fronts (-)
- \( COP \): coefficient of performance (-)
- \( A \): solution a (-)
- \( B \): solution b (-)
- \( d_1 \): Euclidian distance (W^-1)
- \( \bar{d} \): average distance (W^-1)
- \( |A| \): number of solution in set A (-)
- \( |B| \): number of solution in set B (-)
- \( V \): volume (-)
- \( T \): tube (-)

**REFERENCES**


Takemori, C. K., 2010, Numerical Simulation of the Fluid-Structure Interaction Applied to the Reed Valves of a Miniaturized Compressor under High Frequency Conditions, Ansys South American Conference and ESSS Users Meeting, Sao Paulo, Brazil.


