Pareto-based branch and bound algorithm for multiobjective optimization of a safety transformer

Stephane Brisset, Tuan-Vu Tran

PARETO-BASED BRANCH AND BOUND ALGORITHM FOR MULTIOBJECTIVE OPTIMIZATION OF A SAFETY TRANSFORMER

Stephane BRISSET and Tuan-Vu TRAN

Univ. Lille, Centrale Lille, Arts et Metiers ParisTech, HEI, EA 2697 - L2EP - Laboratoire d'Electrotechnique et d'Electronique de Puissance, F-59000 Lille, France
E-mail: stephane.brisset@centralelille.fr

Abstract. A multiobjective branch and bound method is presented and applied to the bi-objective combinatorial optimization of a safety transformer. New criteria are proposed for the branching and discarding. They are based on the Pareto dominance and contribution metric. The comparison with exhaustive enumeration and non-dominated sorting genetic algorithm confirms the solutions. It appears that exact and approximate methods are both very sensitive to their control parameters.

Keywords: Benchmark, Branch and bound, Multiobjective optimization, Pareto-optimality, Safety transformer.

INTRODUCTION

The design of electromagnetic devices is mainly expressed in the literature in term of problem with continuous parameters. However, these problems are in the second part of the design process and often limited to the fine tuning of some parameters corresponding to the structure selected in the first part. Despite recent progress in topological [1] and combinatorial [2]-[4] optimizations, there is a lack of decision tools for the choice of the structure and materials when dealing with conflicting goals. At this stage, the parameters are mainly discrete and not sorted.

Moreover, the production in very small series practiced by some small and medium firms is supported by standards. It is thus a question of choosing among a great but finite number of solutions rather than to optimize some dimensions finely.

Optimization with discrete variables requires different concept than the conventional continuous one. The computation time of combinatorial optimization is also far more expensive. This is worsening in the design of electromagnetic devices because models are non-linear and time-consuming. Heuristic, Tabu search [5], and Branch and Bound algorithm [2]-[4] can solve combinatorial problems. The formers compute approximate solutions in an affordable time while the latter find exact solutions with higher computing cost.

The first part of the paper is devoted to introduce the context of combinatorial optimization in electrical machines and the main issues for solving this kind of problems. In the second part, the mechanisms of BB algorithm are explained and new criteria for the branching and the initialization are proposed for multiobjective problems. The Pareto-based branch and bound algorithm is applied to the bi-objective optimization of a safety transformer. Results are compared to exhaustive enumeration and non-dominated sorting genetic algorithm. Finally, some conclusions and prospects are given.

CONTEXT AND ISSUES

Context

In the design of electromagnetic devices and especially electrical machines, the variables in optimization problems are often discrete. The variables can be integer and sorted like the number of slots and the number of magnets. When dealing with manufacturing in small series, continuous variables like the diameter of wire and the dimensions of magnetic core may be changed to discrete variables with values taken in the manufacturers' catalogs to take advantage of the price and availability.

Other design variables are more difficult to handle in optimization. They are discrete and cannot be sorted. For instance, the topologies of a rotating electrical machine are inner rotor, outer rotor, and axial flux as shown in Figure 1. The materials are also impossible to sort. This is illustrated by the case of conducting materials. If we compare copper to aluminum, copper has less losses but aluminum is lighter. So they cannot be sorted if both criteria appear in the optimization problem.

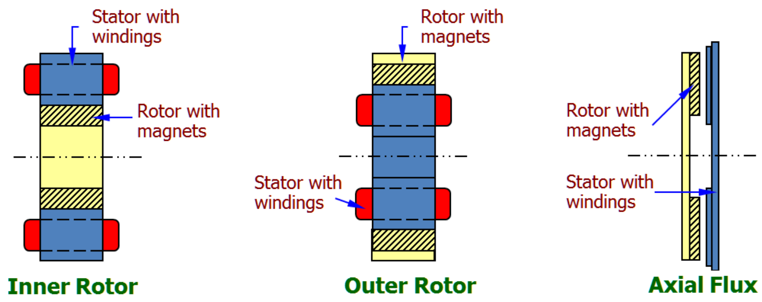

Figure 1. Topologies of a rotating electrical machine

Issues

Three of the main issues that arise with combinatorial optimization are detailed to highlight the difficulties encountered in solving such problems. The first one is called integrality gap and is illustrated in the mathematical example with two design variables, one objective to minimize and two inequality constraints. The optimization problem is expressed in Eq. (1). One simple idea is to search the solution of combinatorial problem in the neighborhood of the solution of the relaxed problem expressed in Eq. (2) that is the problem with all integer variables changed to continuous. This last is the blue disc, written X_c^* in Figure 2. Among the four integer solutions in its vicinity, the best one is the blue square X_d . It can be seen that the distance with the integer optimum X_d^* at red square is significant.

$$
\begin{aligned}
&\min_{x_1,x_2} f(x_1,x_2) = (x_1-3)^2 + (x_2-3)^2 \\
&\text{with} \quad x_1 \in \{0.5, 1.0, 1.5, 2.0, 2.5\} \\
&\qquad\quad\;\; x_2 \in \{0.5, 1.0, 1.5, 2.0, 2.5\} \\
&\text{s.t.} \quad\;\; g_1(x_1,x_2) = 10\,x_1 + 9\,x_2^3 - 17 \le 0 \\
&\qquad\quad\;\; g_2(x_1,x_2) = 2\,x_1 + 5\,x_2 - 8.3 \le 0
\end{aligned} \tag{1}
$$

$$
\begin{aligned}
&\min_{x_1,x_2} f(x_1,x_2) \\
&\text{with} \quad 0.5 \le x_1 \le 2.5 \\
&\qquad\quad\;\; 0.5 \le x_2 \le 2.5 \\
&\text{s.t.} \quad\;\; g_1(x_1,x_2) \le 0 \\
&\qquad\quad\;\; g_2(x_1,x_2) \le 0
\end{aligned} \tag{2}
$$

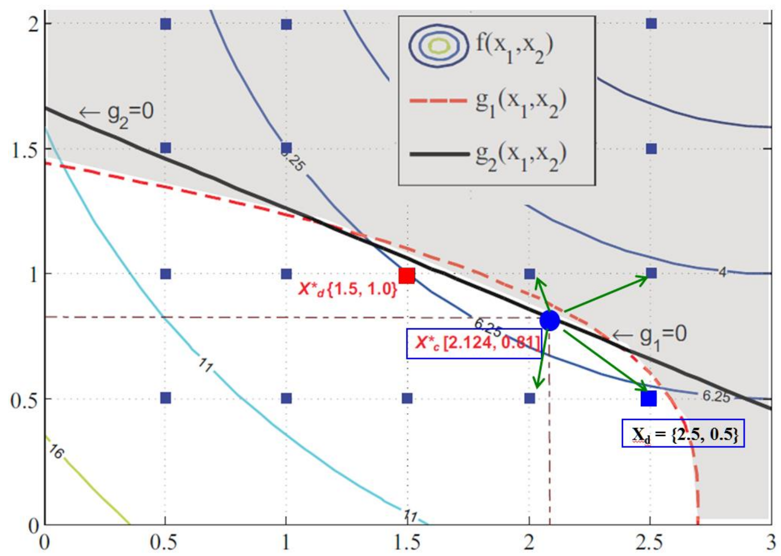

Figure 2. Mathematical example with two design variables (x_1 is abscissa and x_2 is ordinate)

The second issue is that derivatives exist only with continuous variables. Therefore this information cannot be used to define a search direction to guide the optimization. As a consequence, the number of evaluations required for combinatorial optimization is very high.

For multiobjective optimization, scalarization techniques cannot be applied to find the complete set of solutions. It is proved that the weighted sum of objectives fails to find it [6] and the authors' experience is that epsilon-constraint is inefficient and unable to find the complete set of non-dominated solutions.

BRANCH AND BOUND ALGORITHM

The principle of branch and bound algorithm is to divide the search domain into subdomains with lower dimensionality. The separation is made according to hyperplanes orthogonal to the variables axis. A tree spanning all combinations is erected (Figure 3). Given n, the number of integer design variables, there are n levels in the tree and one root at ground level. At the root node, all integer variables are relaxed, i.e. transformed to continuous variables. At the next levels, the evaluation of partial solutions at the nodes is an optimization with some integer variables relaxed and others being constant. Considering level k, the first k integer variables are fixed to constant values and the n-k lasts are relaxed. Therefore, the dimensionality of the optimization problem for the evaluation of partial solutions is decreasing while the level number increases. The leaves are at the last level with all integer variables set to constant value.

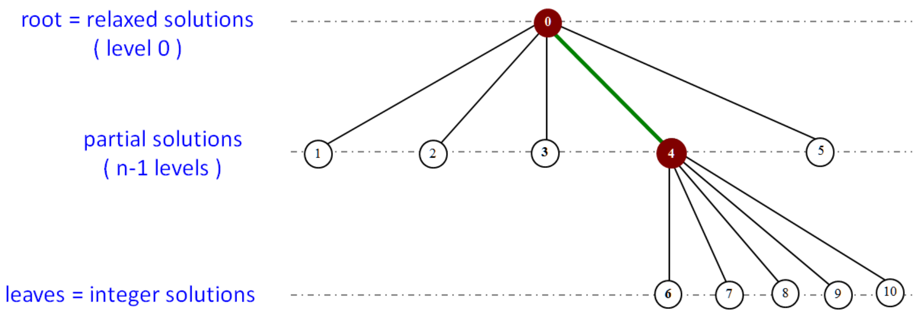

Figure 3. Spanning tree of combinations used in BB algorithm

The mechanisms used in BB algorithm are bounding, branching, and pruning as detailed below.

Branching scheme

Branching is the mechanism used to explore the tree in order to find all the solutions in a reduced computing time and memory requirement. The branching schemes available for BB are depth-first, best-first, and hybrid.

The former scheme is used here. This way, the tree is explored recursively by following the branch leading to the most promising node among all branches connected to the current node. The descent is continued until a leaf is reached. In order to reduce the amount of memory used, the tree is not described entirely during the initialization of algorithm but expanded progressively by adding a sub-tree after each branching. The depth-first scheme also offers the advantage to be the fastest to reach leaves and to enable the pruning of the tree.

The definition of the most promising node is based on a criterion to compare nodes to each other. In BB algorithms, this criterion is the bound.

Bounding method

BB methods have proved to perform well on many combinatorial optimization problems on the condition to provide a good bounding function. The bound can be computed by using an optimization process on a sub-problem with some discrete variables fixed to feasible values and others relaxed, i.e. transformed to continuous variables. This makes sense for discrete variables that can be sorted if the objective and constraint functions could be defined for any real value within the range bounded by the minimum and maximum discrete values.

For the sake of simplicity, we consider first a single objective optimization as stated by the single axis in Figure 4. As the search sub-domain of nodes at the level k+1 is included in the domain at level k, the minimum of the problem for a node at level k+1 is higher or equal to the one for the node at level k to whom it is connected. Therefore, the partial solution at level k stands as a lower bound for all nodes at higher levels connected to it. Moreover, if no feasible solution is found at level k, no solution could be found at next levels. As a consequence, the most promising node is the partial solution with the lowest objective value and all constraints fulfilled.

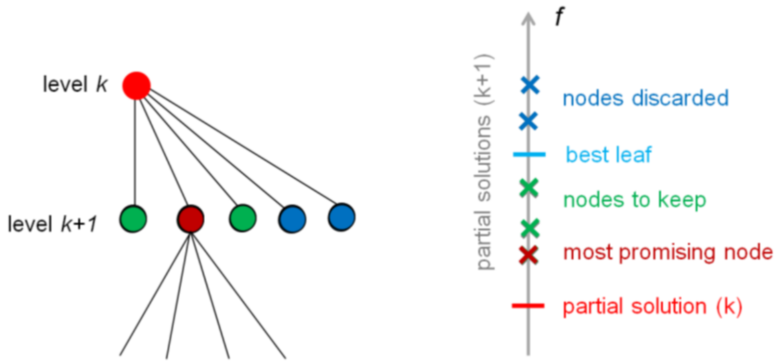

Figure 4. Branching and discarding in single-objective BB algorithm

Elimination strategy

One of the advantages of the branching scheme called depth-first if to be the fastest to reach the leaves. This is done by using $\sum_{i=1}^{n-1} v_i$ partial evaluations, i.e. optimizations and v_n evaluations while the number of combinations, i.e. leaves is $\prod_{i=1}^{n} v_i$ where v_i is the number of values allowed for the discrete variable x_i. The BB algorithm explores the tree iteratively by starting a descent each time that the leaf level is reached until no node remains. So, this is important to prune the tree as soon as possible to reduce the number of iterations and evaluations.

It is reminded that the partial solution is a lower bound. Therefore, a node is discarded if the value of the objective function is above the one of the best leaf as illustrated in Figure 4.

Example with two design variables

In Figure 5 is the spanning tree for the example with two variables expressed in Eq. (1). At level 1, the domain is separated according to x_1. The five partial solutions are found by solving optimization problems with x_1 constant and x_2 continuous.

During the first descent, i.e. iteration, the branching is made to the node number 4 with lowest objective value. At level 2, the subdomain corresponding to $x_1 = 2.0$ is separated according to x_2. All nodes are leaves that are evaluated by a computation of objective and constraints. Four leaves are infeasible. The first descent is completed and the objective value of the best leaf is 7.25.

The second iteration starts with the elimination process. The node number 1 is discarded because its objective value (8.982) is higher than best leaf one. Three nodes remain and the descent starts from the node with the lowest objective value (6.014) whose number is 3. A new best leaf is found at leaf number 12 with an objective value of 6.25.

All remaining nodes have higher objective value than the new best leaf and are discarded. The algorithm stops because no better solution can be found. For this reason, BB algorithm is called exact method.

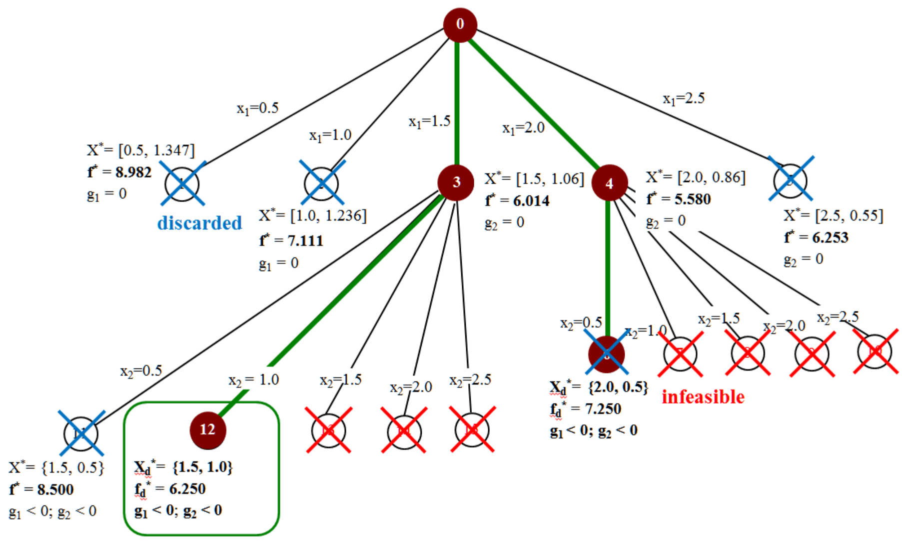

Figure 5. Spanning tree for the example with two design variables

MULTIOBJECTIVE BRANCH AND BOUND ALGORITHM

The branching scheme is kept unchanged for multi-objective optimization branch and bound (MOBB) algorithm and new criteria are proposed to define the most promising node such as the elimination strategy.

Bounding method

Various approaches can be considered to deal with multiple conflicting goals. On one hand, it is tempting to reduce the number of objective to a single one to ease the comparison of partial solutions as in mono-objective BB algorithm. Several methods have been reported in the literature for MOBB algorithms such as scalarization (weighted sum of objectives, etc.).

On the other hand, the solutions of a multi-objective optimization problem are non-dominated solutions and referred as Pareto front when plotted in the objective space. Several metrics are proposed in the literature to assess the performances of multi-objective algorithms and to compare Pareto fronts [6]-[11]:

- D-metric assess the convergence to a reference Pareto front,
- Δ-metric assess uniformity of points along the Pareto front,
- ∇-metric compare Pareto front in term of extent,
- Coverage metric evaluates the dominated area given by a front, and
- Contribution metric evaluates the proportion of Pareto solutions given by each front.

Intermediate approaches in terms of computing time and precision are reported in literature, for instance by using the ideal point whose coordinates are minimum values of objectives minimized separately [4].

In this paper, partial solutions are compared by using the Pareto dominance and the bound is computed with the contribution metric [7]. Therefore, the bound is no more an absolute value but a relative measurement of a node contribution among all the nodes at the same level linked to the same node at the previous level. This bounding method is valid for the depth-first branching scheme only. The most promising node is the one with the highest contribution.

As illustrated in Figure 6, among the solutions nodes at level k+1, nine are non-dominated. Four points belong to pink node and five to green node. As this last has the highest contribution, it is selected for branching.

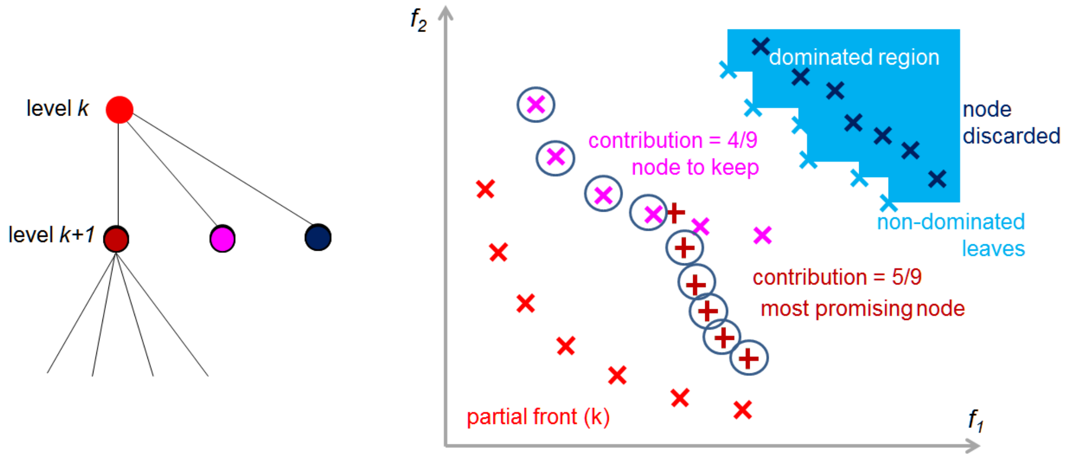

Figure 6. Branching and discarding in multi-objective BB algorithm

Elimination strategy

The set of non-dominated leaves allows defining a dominated region as shown in blue in Figure 6. It is reminded that the partial solution is a lower bound that dominates all solutions at the next levels. Therefore, if all solutions at one node are within the dominated region then the node is discarded.

Initialization

An initial approximation of the Pareto front can lead to reduce the number of evaluations required by the multi-objective branch and bound algorithm. This front may be found by metaheuristic methods but here, it is proposed to use the partial front at root node.

In the left part of Figure 7, the solutions of partial front at root node are plotted in the objectives space and in the right part they are plotted in the design variables space. For each solution, the integer solutions in its vicinity are evaluated and kept if all constraints are fulfilled. The set of non-dominated initial leaves becomes the initial front.

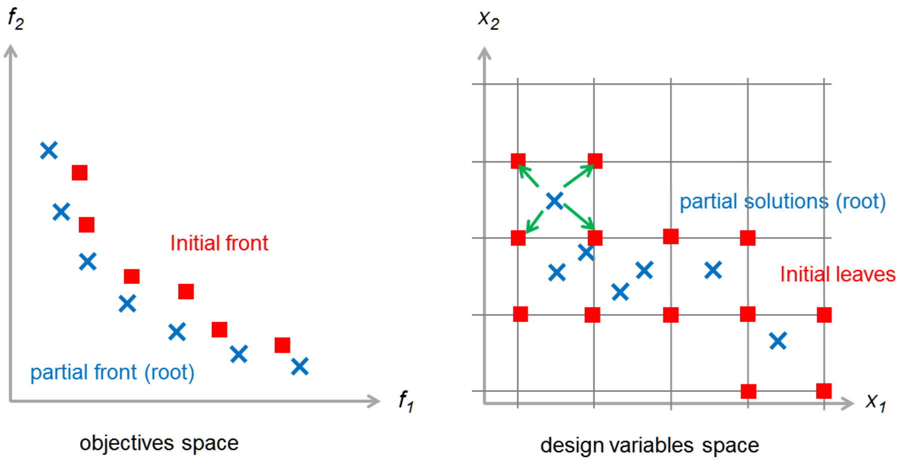

Figure 7. Initial front for multi-objective BB algorithm

Mixed variables

BB algorithms can solve mixed variables problems, i.e. problems with discrete, integer, and continuous variables. Continuous variable can take any real value within a range. Integer variables are sorted and assumed to be relaxable. This means that the objectives and constraints functions are defined for any real value between the minimum and the maximum discrete values and are smooth. This can be critical when using a finite element model and considering an integer variable like the number of slots. Discrete variables cannot be sorted and are not relaxed during the partial evaluations of BB algorithm.

Therefore, a first spanning tree is raised with the discrete variables and its leaves become the roots of sub-trees with integer and continuous variables. The leaves of the sub-trees are evaluated by an optimization where all the integer variables are set to constant value and the continuous variables are unchanged.

COMBINATORIAL OPTIMIZATION OF SAFETY TRANSFORMER

Device and models

The electromagnetic device studied here is a safety isolating transformer [12] (Fig. 8). An analytical model is used to compute the objectives and constraints. This model accepts continuous parameters as inputs and can thus be used with relaxation techniques.

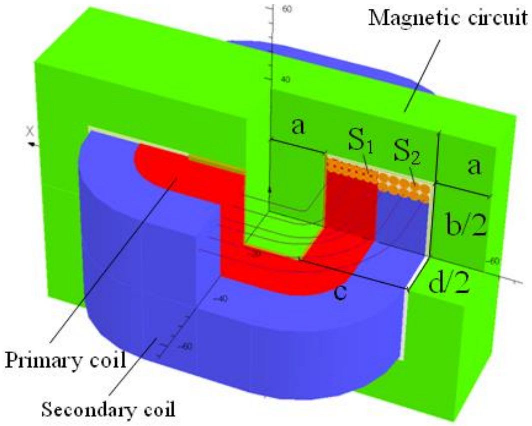

Figure 8. Safety transformer variables

Thermal and magnetic phenomena are both modeled by using 3D FEA on one eighth of the transformer due to the symmetries (Fig. 9). All magnetic and electric quantities are assumed sinusoidal. Full-load and no-load simulations are used to compute all the characteristics. The iron loss is computed with Steinmetz formula and the leakage inductances are calculated with the magnetic co-energy. The magneto-thermal coupling is weak and consistency loop requires 2 hours.

To reduce the computing time, a lumped-mass model is preferred. The additional modeling hypotheses are uniform distribution of induction in the iron core, no voltage-drop due to the magnetizing current, and uniform temperatures in coils and lamination. The multiphysic coupling is strong and the computing time is 50 ms.

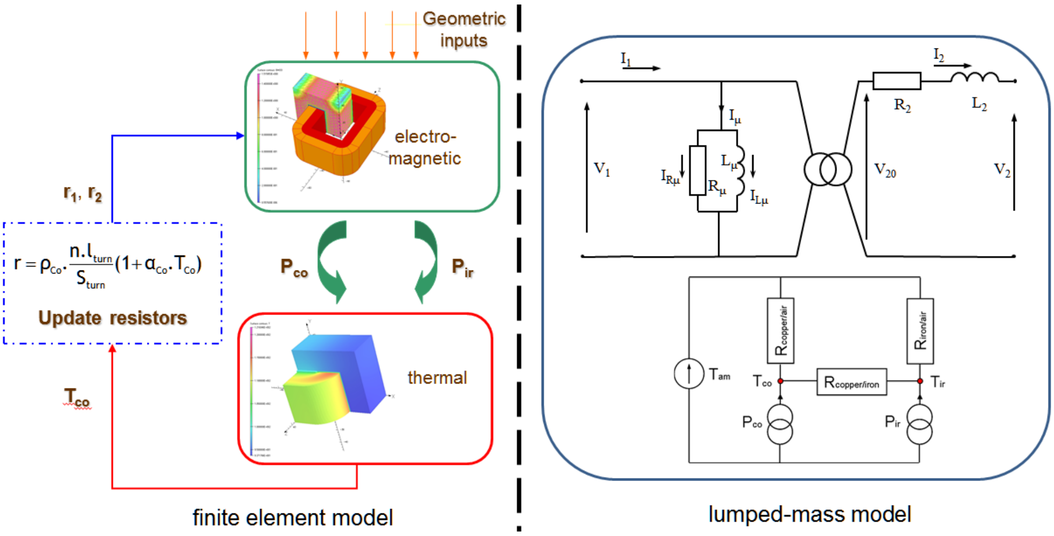

Figure 9. Thermal and electromagnetic models of the safety transformer

Optimization problem

The problem contains 7 integer design variables that are shown in Fig. 8. There are three parameters (a, b, c) for the shape of the lamination, one for the frame (d), two for the section of enameled wires (S_1, S_2), and one for the number of primary turn n_1. As the first four integer variables are linked, the problem becomes a mixed-integer problem with one discrete variable for the combination of the firsts four and the three remaining integer variables. There are 24 types of lamination from catalogue r.bourgois®, 62 possible matches between the laminations EI and the frames from catalogue isolectra-martin®, and 62 types of enameled wires from invex®. The number of primary turn n_1 is integer but only 1001 values are allowed, leading to 238,566,328 combinations.

There are 7 inequality constraints in this problem. The copper and iron temperatures T_{cond}, T_{iron} respectively are less than 120°C and 100°C. The magnetizing current I_{10}/I_1 and drop voltage $\Delta V_2/V_2$ are less than 10%, the filling factor of both coils f_1, f_2 is lower than 1, and the residue of coupled equations is less than 10^{-6}. The objective is to minimize the total mass M_{tot} of iron and copper materials and to maximize the efficiency η:

$$
\begin{array}{ll}
 & \min \quad M_{tot} \quad \max \quad \eta \\
(3) & \text{s.t.} \quad \begin{array}{llll} T_{cond} \le 120°C & T_{iron} \le 100°C & & \dfrac{I_{10}}{I_1} \le 0.1 \\ f_1 \le 1 & f_2 \le 1 & residue < 10^{-6} & \dfrac{\Delta V_2}{V_{20}} \le 0.1 \end{array}
\end{array}
$$

$$
\begin{array}{lll}
 & \{a, b, d, c\} \in EI & \text{(62 configurations)} \\
(4) & n_1 \in \{200, \cdots, 1200\} & \text{(1001 values)} \\
 & S_1 \in W \quad S_2 \in W & \text{(62 values each)}
\end{array}
$$

Results

A time consuming exhaustive enumeration of solutions is performed with distributed computing on 24 cores and has found 749 non-dominated solutions with 238,566,328 evaluations of the model.

For MOBB, the bound is computed at each node with the Pareto set of the sub-problem. This set is found with sequential quadratic programming (SQP), ε-constraint, and multi-start. Its precision is depending on the number of starting points and the number of Pareto points requested. With 20 Pareto points and one starting point, the whole set of solutions is found. Initialization of MOBB with integer solutions in the vicinity of relaxed ones slightly decreases the number of evaluations by 7% as detailed in Table 1.

The number of evaluations and the size of the Pareto set found by MOBB are compared with NSGA-II [8]. An implementation of this algorithm is given by Song Lin in Matlab Central (NGPM v1.4, 2011). For NSGA-II, the results depend on the population size and the number of generations. One thousand individuals and two hundred generations are sufficient to find the whole solutions with five times less evaluations than MOBB.

Table 1. Comparison of algorithms

algorithm	population size or Pareto points	generations or starting points	number of evaluations	size of Pareto set
MOBB	10	1	592,784	732

MOBB + initialization	10	1	549,243	732
MOBB	10	10	7,326,511	732
MOBB + initialization	10	10	7,089,802	732
MOBB	20	1	957,042	**749**
MOBB + initialization	20	1	**888,433**	**749**
NSGA2	750	200	150,000	646
NSGA2	750	300	225,000	671
NSGA2	750	400	300,000	681
NSGA2	1000	200	**200,000**	**749**

On Figure 10 are two Pareto fronts. The Pareto front found by solving the relaxed problem at root node is in blue. It is continuous and contains one hundred points linked by lines. The solutions of the combinatorial problem are in red and the front is discontinuous. All the solutions of the combinatorial problem are dominated by those of the relaxed one, as expected.

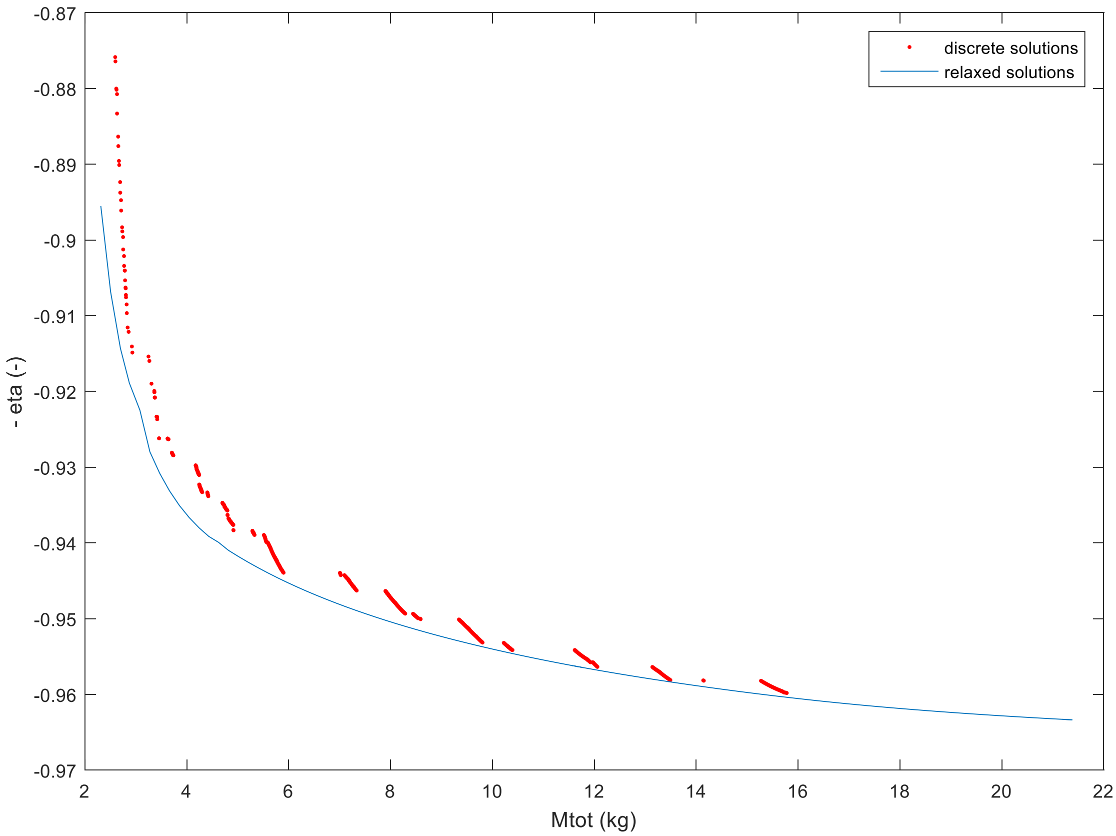

Figure 10. Pareto front found by of MOBB (discrete) and initial front (relaxed)

Discussions

The main weakness of MOBB is the computing burden to evaluate partial solutions while the main drawbacks of genetic algorithms that are the precision of the solution and especially the respect of constraints vanish in integer optimization. It could differ with a mixed-integer optimization problem because of the lack of precision of genetic algorithm for the continuous components of the solutions and the fulfillment of constraints when solutions are on the limit-state.

One track to reduce the number of evaluations of MOBB is to lower the precision of bounds as their purpose is to compare the partial solutions among them and not to calculate the integer solutions. However, we shall remain cautious because Figure 10 shows that the partial front at root node is close to the integer solutions at leaves while they are situated at the opposite extrema of the spanning tree. It may also be possible to be less sensitive to the number of Pareto points by using a most accurate criterion than contribution metric to compare partial solutions.

OSYCZKA AND KUNDU TEST FUNCTION

Optimization problem

The problem is given in Eq. (5) and (6) where all the variables are continuous within bounds [13]. It is changed to a combinatorial optimization problem by using a step of 0.25 for all variables. Obviously, one objective and one constraint are non-convex. As a consequence, SQP, ε-constraint, and multi-start may fail to find exact partial solutions of sub-problems.

$$\text{min} \quad f_1 = -25(x_1-2)^2 - (x_2-2)^2 - (x_3-1)^2 - (x_4-4)^2 - (x_5-1)^2$$
$$f_2 = x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 + x_6^2 \tag{5}$$
$$\text{s.t.} \quad x_1 + x_2 - 2 \geq 0 \quad 6 - x_1 - x_2 \geq 0 \quad 2 + x_1 - x_2 \geq 0$$
$$2 - x_1 + 3x_2 \geq 0 \quad 4 - (x_3-3)^2 - x_4 \geq 0 \quad (x_5-3)^2 + x_6 - 4 \geq 0$$

$$0 \leq x_1, x_2, x_6 \leq 10 \quad 1 \leq x_3, x_5 \leq 5 \quad 0 \leq x_4 \leq 10 \tag{6}$$

On Figure 11 are two Pareto fronts. The Pareto front found by solving the relaxed problem is in blue. It is found by SQP and ε-constraint with 500 values of ε and 500 starting points. The solutions of the combinatorial problem are in red.

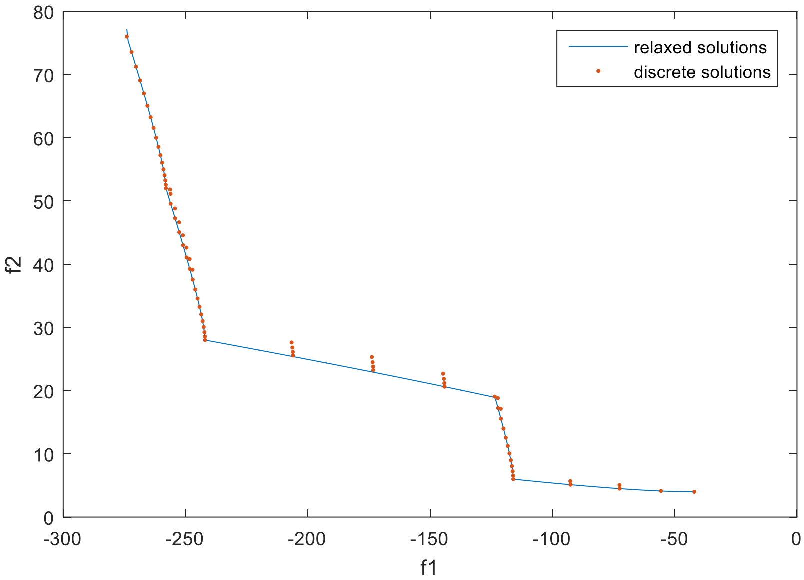

Figure 11. Pareto fronts for discrete and relaxed variables

Results

An exhaustive enumeration of solutions has found 73 non-dominated solutions with 497,954,225 evaluations. In table 2, it can be seen that MOBB fails to find the complete set of solutions in an affordable number of evaluations while NSGA2 succeed. This test highlights that the proposed algorithm has poor performances with non-convex functions.

Table 2. Comparison of algorithms

algorithm	population size or Pareto points	generations or starting points	number of evaluations	size of Pareto set
MOBB	100	10	37,576,081	47
MOBB + initialization	100	10	17,356,443	47
MOBB	200	10	85,519,229	60
MOBB + initialization	200	10	35,677,027	57
NSGA2	76	100	**7,600**	**73**

CONCLUSIONS

MOBB and NSGA-II succeed to find the complete Pareto set of the bi-objective safety transformer combinatorial problem with integer variables but NSGA-II is yet four to five times faster. The performances of both algorithms are sensitive to their control parameters.

Three prospects are proposed to reduce the number of evaluations required by exact methods. The first one is to do a preliminary pruning of the spanning tree by evaluating explicit constraints on the filling factor of both coils.

The main drawback of the algorithm proposed is the high number of evaluations required to compute the partial solutions at nodes. The second prospect is to lower the precision of bounds in order to reduce the computation burden and to use a new criterion to compare partial solutions.

Branch and cut algorithms process is similar to BB but the separation is made according to cut plane defined by the constraints of the optimization problem. The third prospect is to test it to see if it is faster than BB for hard constrained problems.

Osyczka and Kundu test function highlights that the proposed algorithm has poor performances with non-convex objective and constraint.

REFERENCES

[1] M. Stolpe, “Topology optimization – Recent applications and generalizations”, Keynote lecture, OIPE 2014 conference, Delft, The Netherland, September 2014.

[2] E.I. Amoiralis, M.A. Tsili, P.S. Georgilakis, A.G. Kladas, and A.T. Souflaris, "A parallel mixed integer programming-finite element method technique for global design optimization of power transformer", IEEE Transaction on Magnetics, vol. 44, no. 6, pp. 1022-1025, June 2008

[3] T. Hemker, O.v. Stryk, H.D. Gersem, and T. Weiland, "Mixed-integer nonlinear design optimization of a superconductive magnet with surrogate functions", IEEE Transaction on Magnetics, vol. 44, no. 6, pp. 1110-1113, June 2008

[4] F. Sourd, O. Spanjaard. “A multiobjective branch-and-bound framework: Application to the biobjective spanning tree problem”, INFORMS Journal of Computing, Published Online, August 1, 2008, pp. 472-484.

[5] F. Glover and M. Laguna, *Tabu Search*, Collin R. Reeves Edition, Modern Heuristic Techniques for combinatorial problems, Blackwell Scientific Publication, Oxford, pp. 70-150, 1993.

[6] M. Ehrgott, *Multicriteria Optimization*, 2nd edition, Springer, 2005

[7] C. Erbas, S. Cerav-Erbas, and A.D. Pimentel, “Multiobjective optimization and evolutionary algorithms for the application mapping problem in multiprocessor system-on-chip design”, IEEE Trans. on Evolutionary Computation, vol. 10(3), 2006, pp. 358-374.

[8] K. Deb, A. Pratap, and S. Agarwal. “A fast and elitist multiobjective genetic algorithm NSGA-II”, Evolutionary Computation, vol. 6(2), 2002, pp. 182-197.

[9] C.M. Fonseca and P.J. Fleming, “On the performance assessment and comparison of stochastic multiobjective optimizers”, Lecture Notes in Computer, Science, Springer-Verlag, 1996, 1141, 584593.

[10] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. Grunert da Fonseca, “Performance assessment of multiobjective optimizers: an analysis and review”, Technical report 139, Computer Engineering and Networks Laboratory, ETH Zurich, June 2002

[11] D. Joshua, D. Knowles, L. Thiele, and E. Zitzler, “A tutorial on the performance assessment of stochastic multi-objective optimizers”,. TIK-Report 214, Computer Engineering and Networks Laboratory, ETH Zurich, February 2006

[12] T.V. Tran, S. Brisset, and P. Brochet. “Proposal of a benchmark for multi-level optimization with 3D finite element model”, COMPUMAG 2009 conference, Florianopolis, Brazil, November 2009, http://optimisation.l2ep.ec-lille.fr/benchmarks/safety_transformer/safety_transformer_benchmark_part_II.pdf

[13] A. Osyczka, S. Kundu, « A new method to solve generalized multicriteria optimization problems using the simple genetic algorithm », Structural Optimization, vol. 10, 1995, pp. 94–99