

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

185,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Tackling the Industrial Car Sequencing Problem Using GISMOO Algorithm

Arnaud Zinflou¹ and Caroline Gagné²

¹*Institut de recherche d'Hydro-Québec (IREQ)*

²*Université du Québec à Chicoutimi
Canada*

1. Introduction

In many industrial sectors, managers are confronted with problems of an ever-growing complexity. The problem could be bus route optimization for a public transporter, production cost minimization, decision-making support, electronic circuit performance enhancement, or computer system process scheduling. In many cases the situation can be expressed as a combinatorial optimization problem. Solving an optimization problem consists of determining the best solution(s) validating a set of user-defined constraints and goals. To determine if one solution is better than another, the problem must include at least one performance evaluation metric that allows solutions to be compared. The best (or *optimal*) solution, is thus the one with the best evaluation, with respect to the defined goal. When only one goal is specified (e.g. total distance minimization), the optimal solution is clearly defined (the one with the smallest distance).

However, in many situations there are several contradictory goals that have to be satisfied simultaneously. In fact, real-world optimization problems rarely have a single goal. This is the case for the Industrial Car Sequencing Problem (ICSP) on an automobile assembly line. The ICSP consists of determining the order in which automobiles should be produced, taking into account the various model options, assembly line constraints, and production environment goals. In this context, the optimal solution is not a single point, but rather a set of *compromise solutions* called the *Pareto-optimal front*. We can thus define two main goals in multi-objective optimization: (i) Find a set of compromise solutions whose evaluation is as close as possible to the Pareto-optimal front; and (ii) Find a set of compromise solutions as diverse as possible. Attaining these two goals in realistic time is an important challenge for any multi-objective algorithm.

However, in the literature, the ICSP, despite its multi-objective character, has been treated as a problem with a single goal or with several goals lexicographically ordered (Benoist, 2008; Briant *et al.*, 2008; Cordeau *et al.*, 2008; Estellon *et al.*, 2008; Ribeiro *et al.*, 2008). To our knowledge, the only references that treat the ICSP from a purely multi-objective viewpoint are those of Zinflou *et al.* (2009) and of de Oliveira dos Reis (2007); the latter only examines small instances (fewer than 60 automobiles).

Most of the algorithms proposed recently for multi-objective problems are Evolutionary Algorithms (EA) (Deb, 2000; Knowles & Corne, 2000a; Knowles & Corne, 2000b; Zitzler *et al.*, 2001). This is so, doubtlessly because EA's can traverse a large search space to generate

an approximation of the Pareto-optimal front in a single optimization step (Francisci, 2002). One of these algorithms, proposed by Zinflou *et al.* (2011), is a hybrid between a Genetic Algorithm (GA) and an artificial immune system. This approach, called GISMOO, has a small number of parameters to calibrate, is easy to implement, and has been shown to be efficient in solving classical benchmarks in both discrete and continuous optimization. The goal of this chapter is to deepen the understanding of the ICSP from the Pareto viewpoint, by adapting the GISMOO algorithm to solve this problem. On one hand, we mean to show that this is an interesting approach in the solution of the ICSP from a multi-objective viewpoint. On the other hand, because only a few workers have treated this problem from a Pareto viewpoint, we wish to compare the performance of GISMOO with that of other known algorithms, on a real-world industrial problem. This chapter is organized in the following manner. Section 2 briefly describes the industrial car sequencing problem. We subsequently describe, in Section 3, the GISMOO algorithm in order to solve the problem in a Pareto sense. Section 4 exposes the details of the numerical experiments carried out in this chapter. Sections 5 and 6 present the experimental results of GISMOO in comparison to those of two other evolutionary algorithms well known in the literature: NSGAII and PMS^{MO}. Finally, the last section offers some concluding remarks.

2. Industrial car sequencing problem (ICSP)

The ICSP analysed in this chapter was proposed by the automobile manufacturer Renault for the ROADEF Challenge 2005 (Nguyen & Cung, 2005). Each production day, the clients' orders are transmitted to the assembly factory in real time. The daily task of the factory planners consists of: (1) assigning an assembly day to each car ordered, subject to assembly line capacity and clients' delivery time constraints; and (2) scheduling cars within each production day, subject to the constraints of the three shops on the assembly line, as illustrated in Figure 1. Renault specified that the factory technology was such that the body shop constraints could be neglected. Therefore, the ICSP consists of determining the production sequence of a set of cars (*Nb_cars*) subject to the paint and car assembly shop constraints. Since the scheduling goals are thereby achieved, the sequence determined is applied to the whole assembly line.

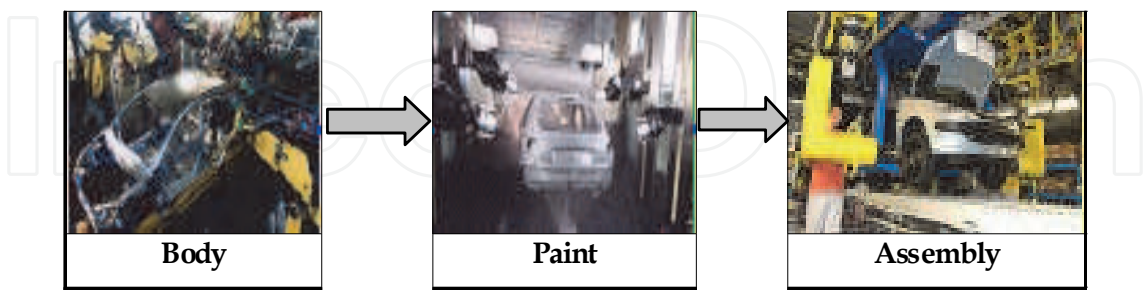


Fig. 1. The three workshops of the assembly line (Nguyen & Cung, 2005)

In the paint shop, we want to keep cars of the same colour together in consecutive order, so that the number of paint gun purges is minimized. To preserve quality, in fact, the paint guns have to be cleaned with solvents, between two cars of different colours and after a certain number (rl_{max}) of cars of the same colour. So the first goal to optimize in the ICSP is to minimize the number of colour changes (COLOR). There is also the constraint that each

purge necessarily means a colour change; therefore, any solution with a number of consecutive cars of the same colour greater than rl_{max} is not feasible.

In the assembly shop, several pieces are added to the painted body to complete the car assembly. This is characterized by a set O of options for which the corresponding workstations are designed to accommodate up to a certain percentage of the cars on the line. These capacity constraints are expressed in the form of a ratio r_o/s_o , meaning that for any subsequence of s_o cars in the assembly line, no more than r_o cars can have the option o . Cars will thus be positioned so that the workload will be well distributed at each place in the line. If it is impossible to respect the capacity constraint for an option o in a subsequence of s cars, then the number of cars that exceeds r defines what is called a *conflict*. The ICSP subdivides the assembly shop capacity constraints into two priority types in order to minimize the two following objectives: the number of conflicts for High Priority Options (HPO) and the number for Low Priority Options (LPO).

The proposed implementation puts identical cars into *classes*: cars with the same HPO's and LPO's are included in V classes for which we know the number of cars to produce (c_v). In fact these quantities represent the problem's production constraints. Table 1(a) presents an example instance with 25 (Nb_cars) cars, 5 options (O) generating 6 classes (V), and a possibility of 4 different colours across each class. A production sequence for this ICSP is defined by two vectors representing the car class (*Classes*) and the respective colour code (*Colors*), as presented in Table 1(b). In the rest of this chapter, the solution sequence will be noted as $Y = \{Classes/Colors\}$, and the element at position i of the sequence will be noted as $Y(i) = Classes(i)/Colors(i)$.

			Classes #					
o	r	s	1	2	3	4	5	6
1	1	2	0	1	1	0	0	0
2	2	5	1	0	1	0	1	1
3	1	3	0	1	0	0	0	0
4	3	5	0	0	0	1	0	1
5	2	3	0	1	1	0	1	0
c_v			5	5	4	4	3	4
C o l o u r s	1		2	1	1	2	1	1
	2		1	1	0	2	1	1
	3		1	3	2	0	0	2
	4		1	0	1	0	1	0

(a)

Y	1	2	3	4	5	6	21	22	23	24	15
Classes	3	5	5	4	6	4		3	1	4	5	1
Colors	4	4	2	2	2	2		3	3	1	1	1

(b)

Table 1. Example of an ICSP and its solution

Another ICSP characteristic is the link between the current production day (J), the previous day ($J-1$) and the following day ($J+1$). Any solution for day J has to take into account the

solution for $J-1$ and an extrapolation of the minimum number of conflicts for day $J+1$. Moreover, we add a colour change if the colour of the first car on day J is different from the colour of the last car on $J-1$.

To evaluate the number of option conflicts, a binary matrix S , of dimension O by Nb_cars , is generated by the information contained in the solution vector $Y = \{Classes/Colors\} : S_{oi} = 1$ if the car class $Classes(i)$ in position i has the option o , and $S_{oi} = 0$ otherwise. The decomposition of $Classes$ in Table 1 to form S is presented in Table 2. We can see that Table 2(a) carries forward the last part of the previous day's solution, in order to compute the number of conflicts between the previous and current days. Moreover, the solution in Table 2(b) is computed partly from the following day, assuming there are no options.

For the current day J , options 1, 3 and 4 have no conflict: there are no cases of more than r cars in any subsequence of s cars that have the same option. However, for option 2 there are two conflicts, because there are 4 consecutive cars with option 2, in positions 1 to 5. There is also a conflict in positions 2 to 6, and two more in positions 21 to 25, because the 2/5 capacity constraint is not respected. For option 5, there is a conflict because the 2/3 constraint is violated in positions 1 to 3. Between days J and $J-1$, there is a conflict in positions -1 to 1 for option 1, and another in positions -1 to 2 for option 5. Between J and $J+1$, there is a conflict in positions 22 to 26 for option 2.

If the first 3 options are of high priority and the other two are of low priority, then there are, in this solution Y , 6 HPO conflicts and 2 LPO conflicts. The only step remaining in the computation of Y is to count the colour changes (COLOR).

		Previous day (J-1)					Current day (J)						
Positions		-5	-4	-3	-2	-1	1	2	3	4	5	6
Classes		4	1	4	4	2	3	5	5	4	6	4	
O P T I O N	1/2	0	0	0	0	1	1	0	0	0	0	0	
	2/5	0	1	0	0	0	1	1	1	0	1	0	
	1/3	0	0	0	0	1	0	0	0	0	0	0	
	3/5	1	0	1	1	0	0	0	0	1	1	1	
	2/3	0	0	0	0	1	1	1	1	0	0	0	

(a)

		Current day (J)						Following day (J+1)					
Positions		21	22	23	24	25	26	27	28	29	30	
Classes			3	1	4	5	1						
O P T I O N	1/2		1	0	0	0	0	0	0	0	0	0	
	2/5		1	1	0	1	1	0	0	0	0	0	
	1/3		0	0	0	0	0	0	0	0	0	0	
	3/5		0	0	1	0	0	0	0	0	0	0	
	2/3		1	0	0	1	0	0	0	0	0	0	

(b)

Table 2. Computation of the solution shown in Table 1

The ROADEF Challenge 2005 uses a weighted sum to evaluate the solution Y :

$$F(Y)=w_1*obj_1+w_2*obj_2+w_3*obj_3 \tag{1}$$

where obj_1 , obj_2 and obj_3 correspond respectively to the value of Y on each objective, for the given priority ordering. The weights w_1 , w_2 and w_3 are respectively 1000000, 1000 and 1 (Nguyen & Cung, 2005). Renault’s factory configuration gives three possible priority orderings: HPO-COLOR-LPO, HPO-LPO-COLOR and COLOR-HPO-LPO. For a more detailed description of this problem, the reader is referred to (Nguyen & Cung, 2005) and (Solnon *et al.*, 2008).

3. Looking for compromise solutions for the ICSP using GISMOO

The algorithm used in this chapter to solve the ICSP from the Pareto domination viewpoint is GISMOO. This algorithm was introduced by Zinflou *et al.* (2011) to solve the classical benchmarks in combinatory optimization with discrete and continuous variables. GISMOO has an original iterative process in two phases: a Genetic phase and an Immune phase. The new solutions (also called descendants) are obtained by offspring creation from the classical genetic operators and by clone creation by the principle of clonal selection in artificial immune systems. Figure 2 presents the general two-phase operation of GISMOO. Even though this is a generic multi-objective algorithm, the description presented in this chapter is specific to the ICSP.

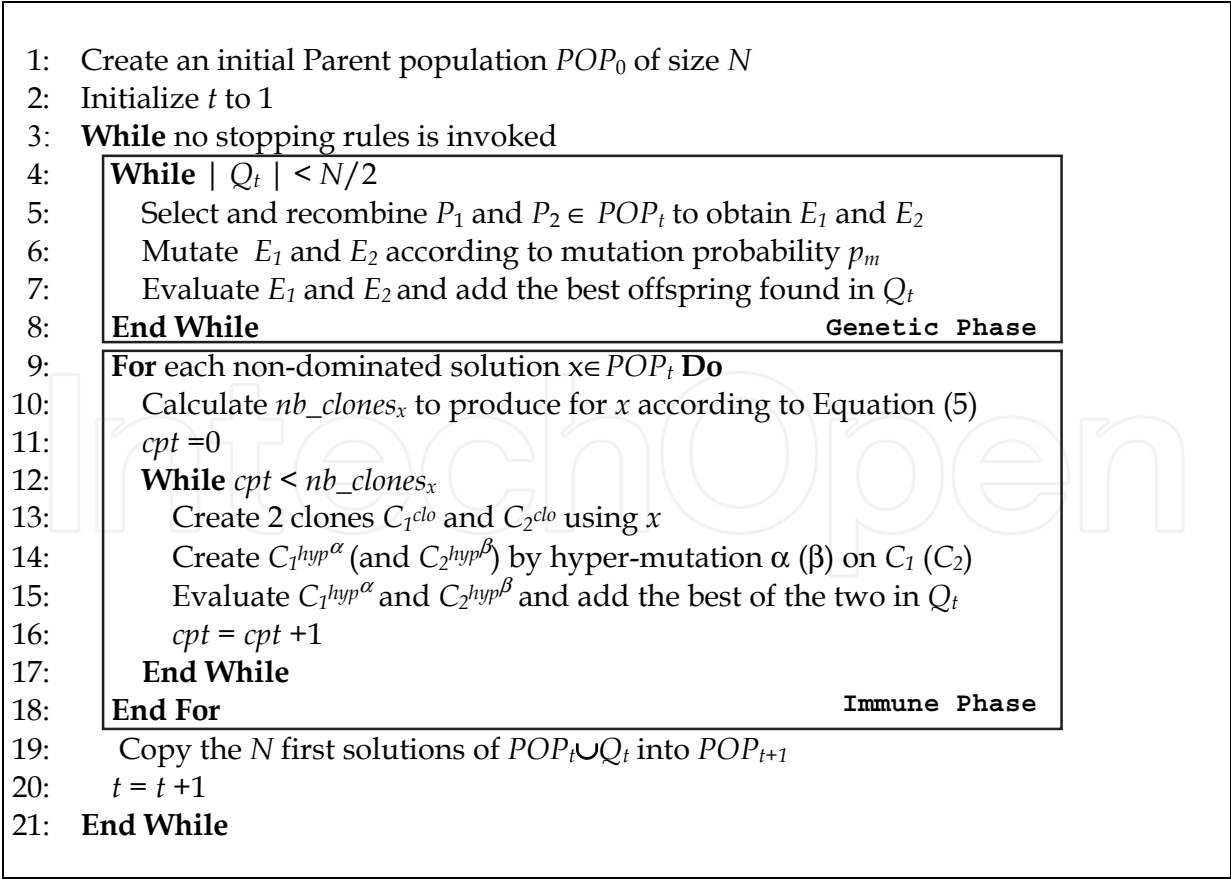


Fig. 2. Outline of GISMOO procedure

The main loop of GISMOO (lines 3-21) begins with the Genetic phase (lines 4-8) and generates $N/2$ offspring. This phase has the classical GA operations: selection, crossover and mutation. Notice that the selection procedure used in GISMOO is a binary tournament selection. In addition, even if two offspring are created during the recombination, only the best of the two is added to the Descendant population Q . It is important to mention that no crossover probability is needed in the Genetic phase of GISMOO, as the number of offspring to generate is related to the Parent population (POP) size. However, a mutation probability (p_m) is used to determine whether the generated offspring will mutate or not (line 6). Thereafter, the Immune phase (lines 9-18) adds $N/2$ solutions to the Descendant population Q . The number of clones produced from a non-dominated solution is proportional to an isolation factor defined in Section 3.1. In this way, a more isolated solution will generate a greater number of clones. After the Genetic and Immune phases, an elitist population replacement is made, in order to keep the N best solutions of the combined Parent and Descendant populations (line 19).

3.1 Performance assignment

Before making a formal presentation of the components of GISMOO that solves the ICSP, it is important to carefully explain how solution performance is assigned. One of the main difficulties in solving multi-objective optimization problems by a Pareto EA is finding a quality metric ordering the solutions in a population. In fact, the quality of a solution depends on the evaluation of several contradictory and often incommensurable objectives. One of the mechanisms most often used by Pareto EA's is expressing the quality of the solution as a function of two factors: a *dominance factor* and an *isolation factor*. The first factor measures the solution's degree of dominance in Pareto sense, and the second evaluates the density of solutions around a given solution. Even if the EA's share this performance assignment mechanism, each one has its own definition.

GISMOO's dominance factor, similar to that in PMS^{MO} (Zinflou *et al.*, 2008b), is calculated in two steps. First, a force $S(x)$, the number of solutions dominated by x , is assigned to each solution x in the Parent population (POP), combined with the Descendent population Q . A feasible solution x *dominates* a feasible solution y iff there is an objective $i \in Z$ for which $f_i(x) < f_i(y)$ and such that for all other objectives $j \in Z$ ($j \neq i$), $f_j(x) \leq f_j(y)$, where f_i is the objective function for i and Z is the number of objectives to minimize. According to the force S , the dominance factor of an individual x , noted $R^+(x)$, is determined using the following equation.

$$R^+(x) = \begin{cases} \frac{S(x)}{1 + 2 * S(x)} & \text{si } \sum_{y \in POP_t \cup Q_t, y > x} S(y) = 0 \\ \sum_{y \in POP_t \cup Q_t, y > x} S(y) & \end{cases} \quad (2)$$

The above calculation thus establishes a dominance relation between solutions based on actual objectives as well as dominated solutions in the search space. Even though GISMOO's calculation is similar to that in PMS^{MO} , the considered populations in the two algorithms are not the same.

GISMOO's isolation factor is inspired by the spacing metric sp introduced by Schott (1995), which evaluates the distance $Dist(x)$ between an individual x and its closest neighbour y (with $y \neq x$), as indicated in Equation (3).

$$Dist(x) = \min_{y \in POP \cup Q} \left[\sqrt{(f_1(x) - f_1(y))^2 + \dots + (f_Z(x) - f_Z(y))^2} \right] \quad (3)$$

We note that GISMOO does not add the dominance and isolation factors. Rather, it assigns a better performance to the solutions with a high dominance factor, and in case of tie, the equality is broken using the isolation factor.

3.2 Representation of the chromosome

Instead of a classical bit string representation that seems poorly adapted to this type of problem, our representation of a chromosome is composed of two vectors of length Nb_cars , corresponding to the cars' option class and colour, as already indicated in Table 1(b).

3.3 Creating of the initial population

In our proposed implementation, the initial population of solutions is generated in two ways: 70% randomly and 30% with a greedy heuristic based on the notion of interest. Two greedy heuristics are used here: *greedy_color* and *greedy_ratio*. These two heuristics were first presented by Zinflou *et al.* (2008a), to solve the ICSP lexicographically. Schematically, *greedy_color* minimizes the number of colour changes, whereas *greedy_ratio* minimizes the number of capacity conflicts for HPO's. For more detailed information, see the above reference. Note that at each iteration, there is a random choice between the two heuristics.

3.4 Genetic phase

3.4.1 Crossover operator

To solve the ICSP, the recombination operator we use is the $NCPX^{MO}$ introduced in (Zinflou *et al.*, 2008a). Schematically, this operator tries to minimize the number of car classes to reposition in the sequence, by using the information linked to the conflictual positions in the parent sequence, based on the notion of *Total Weighted Interest (TWI)*, which calculates whether a car of class v and colour $Colors$ should be in position i in the sequence, according to the following equation:

$$TWI_{v,Colors,i} = I_{v,i,HPO} * w_{HPO} + I_{v,i,COLOR} * w_{COLOR} + I_{v,i,LPO} * w_{LPO} \quad (4)$$

where w_{HPO} , w_{COLOR} and w_{LPO} correspond to the weights given to each objective (1000000, 1000 and 1 according to the objective priorities), and $I_{v,i,HPO}$, $I_{v,i,COLOR}$ and $I_{v,i,LPO}$ correspond to the "interest" in placing the car class v in position i for each objective. However, contrary to the original $NCPX^{MO}$, we do not use the TWI notion, but rather the concept of Pareto Interest (PI). Instead of weighting and adding the interest values for each objective, we compare them from a Pareto viewpoint. In this way, the PI allows us to find out if it is good to put a car of class v and colour $Colors$ in position i , according to how it dominates (or not) the other candidate classes. A tie is broken by using an extension of a strategy introduced in (Gottlieb *et al.*, 2003), which favours a better distribution of car classes in the sequence under construction. For more details on these subjects, see (Zinflou *et al.*, 2008a).

3.4.2 Selection

Several selection strategies are possible for an EA solving the ICSP in the Pareto sense. We chose the binary tournament selection because its implementation and execution costs are

low, and because it has already been shown to work for the theoretical car sequencing problem (Zinflou *et al.*, 2007) and the ICSP from the lexicographic viewpoint (Zinflou *et al.*, 2008a). The tournament games are decided by the dominance factor of the participants, and a tie is broken by using the isolation factor.

3.4.3 Mutation operators

The two mutation operators used by GISMOO to solve the ICSP are the swap operator and random exchange. These operators are often used in the literature to explore the neighbourhood of a solution in a local search method to solve the ICSP.

3.5 Immune phase

In GISMOO, the Descendant population (Q) is subdivided into two parts of equal size $N/2$. The subpopulation of offspring is generated with the selection, recombination and mutation operators, and the subpopulation of clones is generated according to the clonal selection principle introduced by De Castro and Timmis (De Castro & Timmis, 2002). To create the clones, the first step is to sort the current Parent population (POP_t) into fronts, using the same principle as the NSGAII (Deb, 2000). The non-dominated individuals of POP_t , those located in the first front, are then selected as the antibody population to be cloned. In GISMOO, the number of clones produced for each antibody is not constant, but rather proportional to its isolation factor: the individuals farther from their neighbours generate more clones. With this technique, we seek to identify and emphasize non-dominated solutions in isolated regions. The number of clones, $nb_clones(x)$, produced for each antibody x is given by Equation (5).

$$nb_clones(x) = round \left[\left(Dist(x) * \frac{N}{2} \right) / \sum_{x=1}^{|Front_1|} Dist(x) \right] \quad (5)$$

where $|Front_1|$ is the number of non-dominated individuals of the current population and $Dist(x)$ is the isolation factor of x defined in Equation (3). The function *round* rounds off the argument to the nearest integer.

Once the number of clones is determined for an individual x , two clones (copies of x) are produced and then hyper-mutated by the respective application of the α and β mutation operators (C_1^{clo} , C_2^{clo}). In the ICSP context, the operator α corresponds to a swap mutation, and β to a random exchange. No new parameters, therefore, need to be created. After the evaluation, the two mutated clones are compared, and the dominant one (in the Pareto sense) is retained. If no dominance relation can be established, one of the clones is chosen at random. In each case, the selected clone is added to the current population Q . For each individual x in $Front_1$, this process is repeated until the number of clones produced attains $nb_clones(x)$.

3.6 Replacement

GISMOO is an elitist algorithm: to conserve the best individuals in the current population, its replacement strategy is deterministic of type $(\lambda + \mu)$ where λ is the Parent population size and μ the Descendant population size. In our approach, $\lambda = \mu = N$.

3.7 Managing elitism

Like any elitist algorithm, GISMOO has mechanisms that retain non-dominated individuals while it is searching for solutions. It uses an archive A to conserve non-dominated solutions during its iterations. However, the individuals in A do not participate in the selection process; only the non-dominated individuals currently in the population do so. It is important to note that the size of A is not fixed or limited by some maximum. Finally, to ensure that the best individuals in the population are conserved in the current population, GISMOO uses the elitist replacement procedure described in the previous paragraph.

4. Numerical experiments

Until now, the ICSP has generally been approached lexicographically or by lumping together the different goals. To our knowledge, the only work that treats the ICSP from a purely multi-objective viewpoint is that of Zinflou *et al.* (2009) and of de Oliveira dos Reis (2007). However, the latter work is limited to instances of fewer than 60 cars. The experiments of Zinflou *et al.* compare the performances of PMS^{MO} to that of NSGAII, using the ROADEF 2005 instances. We use the same two algorithms to analyze GISMOO's performance from the Pareto viewpoint. The version of GISMOO solving the ICSP and used in this chapter was implemented in C++ with Visual Studio .net 2008. The computer used for the numerical experiments is a Dell model with a Pentium Xeon 3.6 GHz processor with 1GB of RAM, operating under Windows XP.

In our implementation, the main data structures are shared by all the algorithms. We thus obtain a common basis for comparing and equitably evaluating the performances of the different algorithms. In all the numerical experiments of this section, each instance of the ICSP was solved 30 times by each algorithm on the same computer. We used the following parameters:

- The size (N) of the populations was empirically fixed at 100 individuals for each instance and each algorithm.
- The mutation probability (p_m) was fixed at 0.06 for the three algorithms. Note that GISMOO does not need a crossover probability. For the other two algorithms, the crossover probability was 1.
- In the context of the ROADEF Challenge 2005, a time limit of 600 seconds on a PC Pentium4/1.6GHz/win2000/1GB RAM was fixed. Because of the computer used in our experiments, as well as the experimental conditions of the Challenge, the time limit for the algorithm was fixed at 350 seconds.
- The maximum size of the local archive for PMS^{MO} was fixed at 100 individuals.

The value of the different parameters discussed here was fixed in accordance with the numerical experiments in (Zinflou *et al.*, 2009). We also mention that the goal of this chapter is not to directly compare the performance of NSGAII to that of PMS^{MO} , but rather to analyze the performances of GISMOO in solving industrial problems. For a direct comparison between NSGAII and PMS^{MO} the reader can consult (Zinflou *et al.*, 2009).

4.1 Test problems

The numerical experiments presented in this chapter were carried out on the three samples furnished by the Renault company for the ROADEF Challenge 2005. The first sample (Set A) had 8 data sets for the scheduling of 334 to 1314 cars with 6 to 22 options, making 36 to 287 car classes with 11 to 24 different colours. This sample was used to evaluate the teams

in the qualifying phase and select the 18 teams for the following round. The second sample (Set B) had 15 instances, each composed of 65 to 1270 cars with 4 to 25 options, 11 to 339 classes and 4 to 20 colours. This sample was used to select the 12 finalist teams in the Challenge. The last sample (Set X) had 19 instances with 65 to 1319 cars, 5 to 26 options, 10 to 328 classes and 5 to 20 colours. This sample was used in the final evaluation of the 12 finalists, in order to select the winning team.

4.2 Performance metrics

Evaluating the performance of a multi-objective algorithm is often an arduous task. Several metrics evaluating such algorithms are mentioned in the literature. The goal of this series of numerical experiments is to compare the performance of GISMOO with that of NSGAII and PMS^{MO} in solving the ICSP while respecting the usual standards in multi-objective optimization. To attain this goal, we use the three metrics introduced by Zitzler & Thiele (1999): the hyper-volume H , the coverage of two sets metric C and the covering difference D .

Schematically, the hyper-volume metric H estimates the “volume under” the “surface” formed by the points of a given solution set. In particular, when the problem has two objectives, this metric corresponds to calculating an area, and for three objectives, a volume is calculated. Formally, the size of the dominated space H is defined in the following way. Let $X = \{x_1, x_2, \dots, x_s\}$ be a set of S feasible solutions. The function H gives the volume enclosed by the union of the polytopes p_1, p_2, \dots, p_s where each p_i is formed by the intersections of the following hyper-planes arising out of x_i , along with the axes: for each axis in the objective space, there exists a hyper-plane perpendicular to the axis and passing through the point $(f_1(x_i), f_2(x_i), \dots, f_z(x_i))$. In the two-dimensional case, each p_i represents a rectangle defined by the points $(0, 0)$ and $(f_1(x_i), f_2(x_i))$.

The C metric is used to represent the relative spread of solutions between two sets of solution vectors, U and V . The value $C(U, V)$ corresponds to the percentage of solutions of V that are weakly dominated by at least one solution of U . A point x *weakly dominates* a point y iff $f_z(x) \leq f_z(y)$ for all z in Z . The C metric is not symmetric: $C(U, V) \neq 1 - C(V, U)$, so it is necessary to consider both values, $C(U, V)$ and $C(V, U)$, to obtain a reliable measure of the two compromise surfaces.

Finally, the D metric is defined in the following way. Let $U, V \subseteq X$ be two sets of decision vectors. The function D is defined by $D(U, V) = H(U+V) - H(V)$ and gives the size of the space weakly dominated by U but not weakly dominated by V in the objective space.

5. Results and discussion

Tables 3 to 5 present the results obtained by the three algorithms, for the three ICSP test data sets, as measured by the H metric. Each row of these tables gives the name of the instance, the number of cars to schedule (Nb_cars), and the average results obtained with H for each algorithm. The best results are shaded in grey. We have indicated by (*) the instances for which the three algorithms found exactly the same set of solutions in each execution.

From the results of Tables 3 to 5, we observe that GISMOO has the best results for all the instances of all the data sets. That is, the size of the space dominated by GISMOO is always greater than or equal to that of the other two algorithms. In fact, when we compare GISMOO with NSGAII, we find that GISMOO is superior in 38 instances and equal in 4

instances. GISMOO has the same score with PMS^{MO}. Between NSGAII and PMS^{MO}, the results are divided and the scores are often very close.

Instance	Nb_cars	NSGAII	PMS ^{MO}	GISMOO
022_3_4	485	1.16 E+8	1.17 E+8	1.22 E+8
024_38_3	1260	6.05 E+6	6.47 E+6	8.7 E+6
024_38_5	1315	1.58 E+7	1.58 E+7	1.93 E+7
025_38_1	1004	1.41 E+8	1.65 E+8	1.76 E+8
039_38_4_ch1	954	7.41 E+6	7.40 E+6	8.30 E+6
048_39_1	600	5.27 E+7	5.7 E+7	6.16 E+7
064_38_2_ch1	875	3.46 E+7	3.48 E+7	3.94 E+7
064_38_2_ch2	335	1.45 E+7	1.52 E+7	1.65 E+7

Table 3. Average hyper-volumes *H* of NSGAII, PMS^{MO} and GISMOO for the instances of Set A of the ICSP

Instance	Nb_cars	NSGAII	PMS ^{MO}	GISMOO
022_S22_J1	526	1.54 E+6	1.86 E+6	2.03 E+6
023_S23_J3	1110	6.46 E+6	6.98 E+5	7.26 E+6
024_V2_S22_J1	1270	7.6 E+7	8.71 E+7	1.04 E+8
025_S22_J3	1161	8.04 E+6	9.44 E+6	1.02 E+7
028_ch1_S22_J2	365	3.41 E+6	3.35 E+6	5.43 E+6
028_ch2_S23_J3	65	8.56 E+3	9.13 E+3	9.14 E+3
029_S21_J6	730	2.93 E+7	2.89 E+7	3.56 E+7
035_ch1_S22_J3	128	8.64 E+7	8.79 E+7	9.14 E+7
035_ch2_S22_J3	269	1.39 E+8	1.45 E+8	1.56 E+8
039_ch1_S22_J4	1231	1.24 E+8	1.37 E+8	1.42 E+8
039_ch3_S22_J4	1000	5.86 E+6	1.06 E+7	2.01 E+7
048_ch1_S22_J3	591	6.61 E+7	7.2 E+7	7.41 E+7
048_ch2_S22_J3	546	9.94 E+7	1.01 E+8	1.05 E+8
064_ch1_S22_J3	825	4.24 E+7	4.14 E+7	5.12 E+7
064_ch2_S22_J4	412	8.04 E+7	8.11 E+7	9.15 E+7

Table 4. Average hyper-volumes *H* of NSGAII, PMS^{MO} and GISMOO for the instances of Set B of the ICSP

Although the hyper-volume, by definition, gives a good idea of the size of the space dominated by a solution set, this metric does not let us compare two solution sets with each other. To do that, we can use the set coverage metric *C*. Figure 3 presents the results of the three algorithms according to the *C* metric. In this figure, each box corresponds to the comparison of the solutions of algorithm *U* (row) with algorithm *V* (column). The value *C*(*U*,*V*) indicates the percentage of elements of *V* dominated by at least one element of *U*. In each box, the scale goes from 0 at the bottom to 1 at the top. Each box has 8 graphs corresponding to the 8 instances of Set A. Each boldface horizontal bar indicates the mean of the *C* measures for the 30 executions, and the vertical bars indicate the spread between the maxima and minima of the executions.

Instance	Nb_cars	NSGAII	PMS ^{MO}	GISMOO
022_S49_J2	704	1.17 E+8	1.18 E+8	1.2 E+8
023_S49_J2	1260	5.46 E+7	5.53 E+7	6.59 E+7
024_S49_J2	1319	1.02 E+8	1.02 E+8	1.1 E+8
025_S49_J1	996	2.30 E+8	2.36 E+8	2.53 E+8
028_CH1_S50_J4	325	1.51 E+8	1.55 E+8	1.66 E+8
028_CH2_S51_J1*	65	1.00 E+3	1.00 E+3	1.00 E+3
029_S49_J5	780	1.54 E+8	1.34 E+8	1.72 E+8
034_VP_S51_J1_J2_J3	921	1.45 E+8	1.5 E+8	1.54 E+8
034_VU_S51_J1_J2_J3	231	9.05 E+7	9.11 E+7	1.00 E+8
035_CH1_S50_J4*	90	9.85 E+6	9.85 E+6	9.85 E+6
035_CH2_S50_J4	376	4.36 E+5	5.00 E+5	6.09 E+5
039_CH1_S49_J1*	1247	2.5 E+8	2.5 E+8	2.5 E+8
039_CH3_S49_J1	1037	1.59 E+8	1.67 E+8	1.81 E+8
048_CH1_S50_J4	519	7.32 E+7	7.36 E+7	8.71 E+7
048_CH2_S49_J5	459	8.5 E+7	9.03 E+7	9.62 E+7
064_CH1_S49_J1	875	3.53 E+7	3.53 E+7	4.19 E+7
064_CH2_S49_J4	273	9.76 E+7	9.76 E+7	9.77 E+7
655_CH1_S51_J2_J3_J4*	264	1.00 E+9	1.00 E+9	1.00 E+9
655_CH2_S52_J1_J2_S01_J1	219	2.08 E+6	2.17 E+6	2.37 E+6

Table 5. Average of the hyper-volumes *H* of NSGAII, PMS^{MO} and GISMOO for the instances of Set X of the ICSP

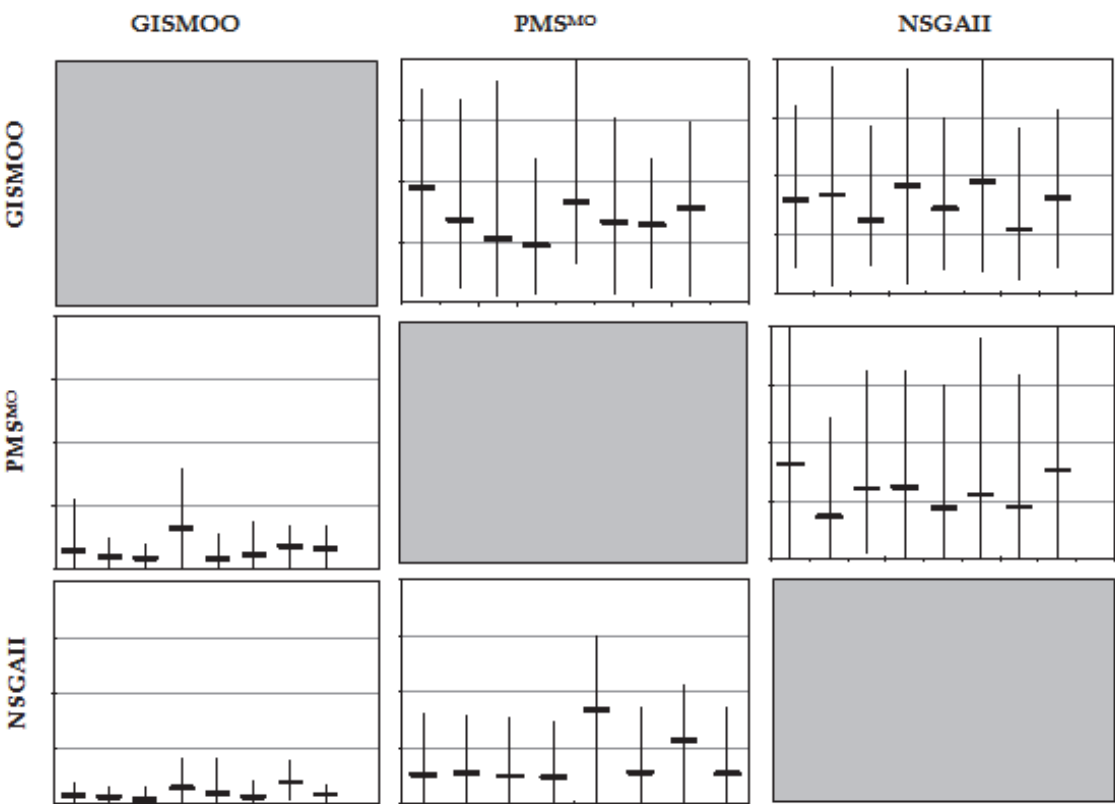


Fig. 3. Average coverage *C* of GISMOO, PMS^{MO} and NSGAII for the instances of Set A

In a way similar to that of Figure 3, Figure 4 presents the results of the three algorithms according to the C metric, for the 15 instances of Set B. Each box has 15 corresponding graphs. The boldface horizontal bar indicates the mean of the C measures for the 30 executions, and the vertical bars indicate the spread between the maxima and minima of the executions.

Finally, Figure 5 presents the results of the three algorithms according to the C metric, for 15 of the 19 instances of Set X. The results for the 4 other instances (028_CH2_S51_J1, 035_CH1_S50_J4, 039_CH1_S49_J1, 655_CH1_S51_J2_J3_J4) are exactly the same for the three algorithms, so that they are not presented here, in order to make the figure easier to read.

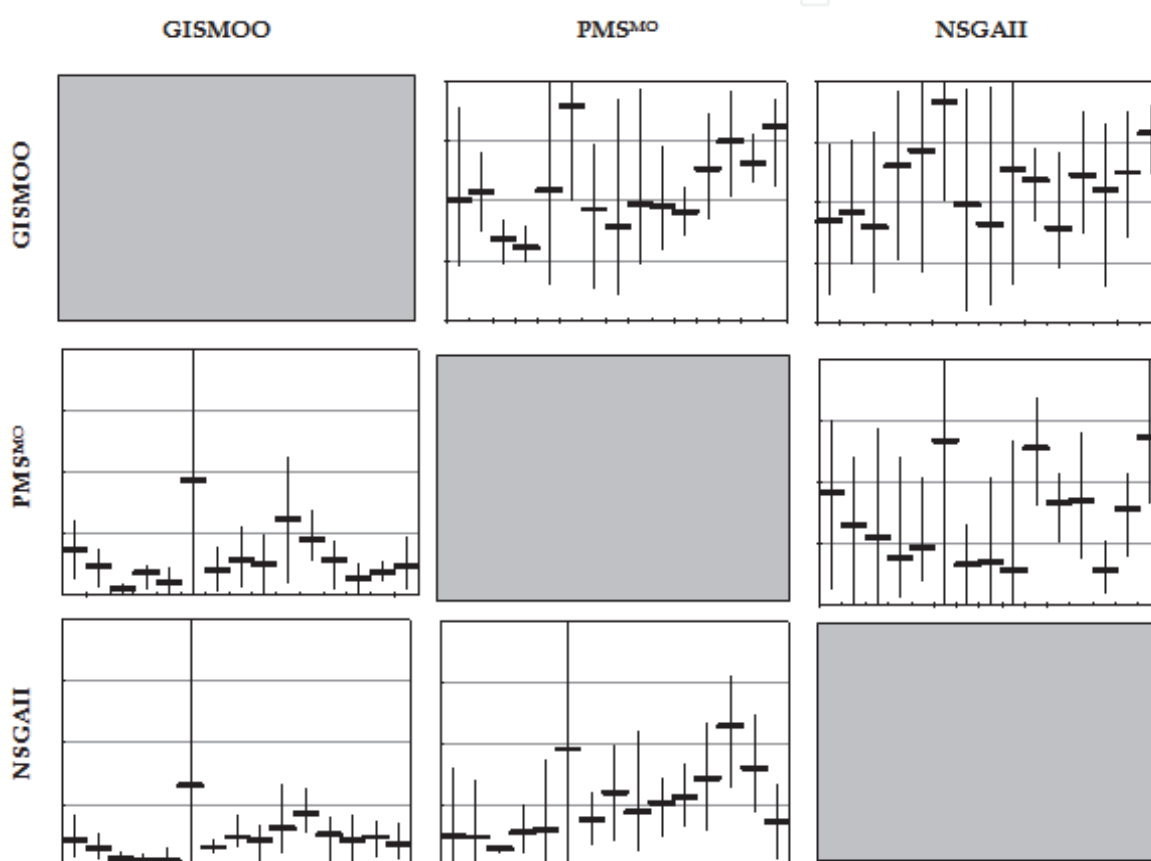


Fig. 4. Average coverage C of GISMOO, PMS^{MO} and NSGAII for the instances of Set B

The results shown are, once again, clearly in favour of GISMOO for all the instances. Indeed, the values of $C(NSGAII, GISMOO)$ are always less than those of $C(GISMOO, NSGAII)$. Note that for many instances, the ratio between the results for the two algorithms is greater than 10; this means that almost all the solutions found by GISMOO are non-dominated or dominate those found by NSGAII. If GISMOO is compared with PMS^{MO} , it is seen that the values of $C(PMS^{MO}, GISMOO)$ are always less than those of $C(GISMOO, PMS^{MO})$, with a ratio greater than 10 for most instances, making almost all the solutions found by GISMOO non-dominated or dominating those found by PMS^{MO} . The results found by the C metric confirm, therefore, the results found by the hyper-volume.

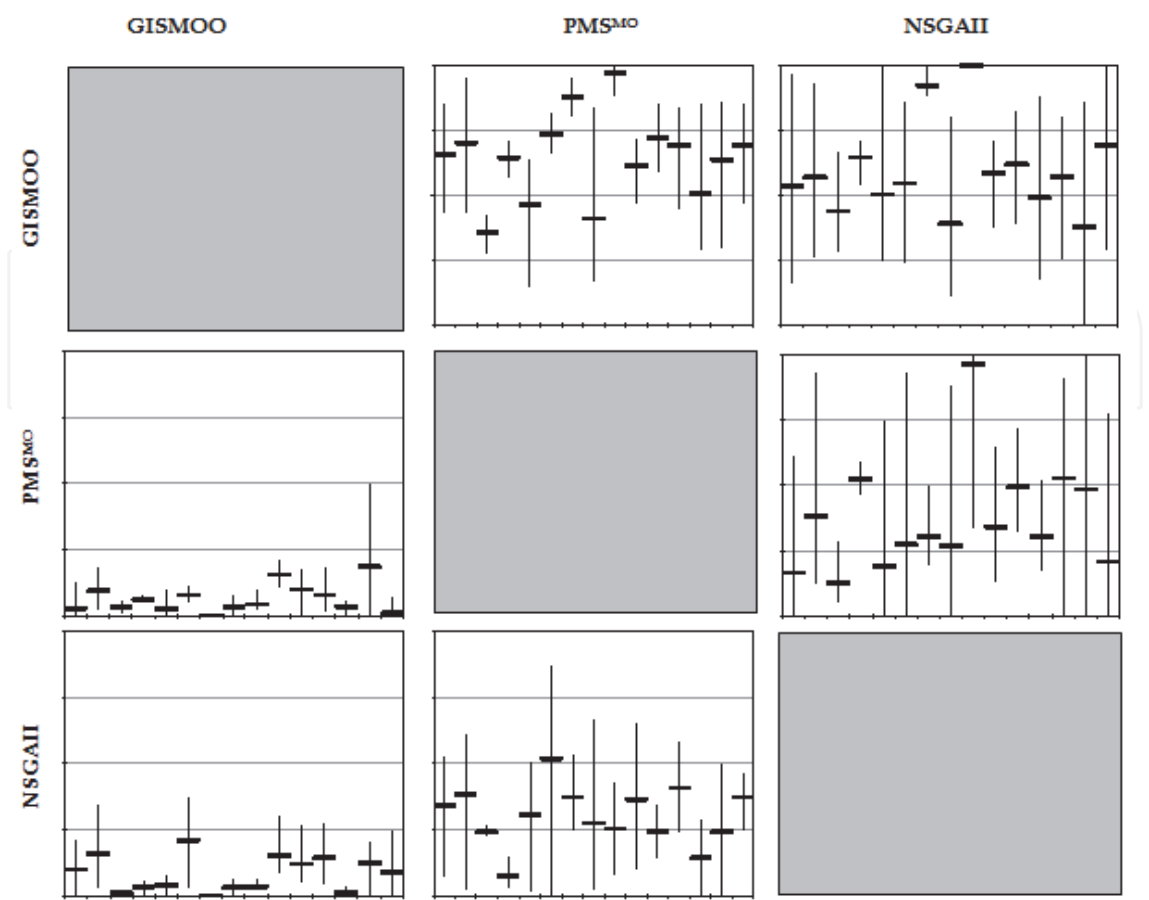


Fig. 5. Average coverage C of GismoO, PMSMO and NSGAII for the instances of Set X

Along with the hyper-volume and the covering, we also compared the three algorithms with the D metric that measures the difference in the covering by the solutions of two algorithms at a time. This metric gives an idea of the size of the solution space dominated by the solutions of one algorithm but not the other. Tables 6 to 11 resume respectively the results this metric obtained between GismoO and NSGAII and between GismoO and PMSMO, for each data set tested. Each row of these tables gives the instance name, the number of cars to schedule and, for each algorithm, the mean results obtained with the D metric after 30 independent executions. The best results are indicated by grey shading.

Instance	Nb_cars	NSGAII	GismoO
022_3_4	485	3.09 E+6	9.10 E+6
024_38_3	1260	1.16 E+5	1.19 E+5
024_38_5	1315	1.06 E+8	1.09 E+8
025_38_1	1004	8.24 E+8	8.59 E+8
039_38_4_ch1	954	4.2 E+6	5.09 E+6
048_39_1	600	1.88 E+8	1.97 E+8
064_38_2_ch1	875	2.11 E+6	2.15 E+6
064_38_2_ch2	335	8.4 E+6	1.05 E+7

Table 6. Average coverage difference D between NSGAII and GismoO for the instances of Set A

Instance	Nb_cars	PMS ^{MO}	GISMOO
022_3_4	485	3.09 E+6	7.88 E+6
024_38_3	1260	1.16 E+5	1.19 E+5
024_38_5	1315	1.06 E+8	1.09 E+8
025_38_1	1004	8.25 E+8	8.35 E+8
039_38_4_ch1	954	4.2 E+6	5.1 E+6
048_39_1	600	1.88 E+8	1.93 E+8
064_38_2_ch1	875	2.11 E+6	2.15 +6
064_38_2_ch2	335	8.5 E+6	9.82 E+6

Table 7. Average coverage difference *D* between PMS^{MO} and GISMOO for the instances of Set A

Instance	Nb_cars	NSGAII	GISMOO
022_S22_J1	526	1.97 E+6	2.46 E+6
023_S23_J3	1110	2.77 E+6	2.85 E+6
024_V2_S22_J1	1270	3 E+6	3.05 E+6
025_S22_J3	1161	2.4 E+6	2.77 E+6
028_ch1_S22_J2	365	1.96 E+6	2.16 E+6
028_ch2_S23_J3	65	8.66 E+2	1.44 E+3
029_S21_J6	730	5.89 E+6	5.96 E+6
035_ch1_S22_J3	128	3.36 E+7	3.86 E+7
035_ch2_S22_J3	269	3.44 E+7	3.61 E+7
039_ch1_S22_J4	1231	1.08 E+8	1.26 E+8
039_ch3_S22_J4	1000	1.05 E+5	1.19 E+7
048_ch1_S22_J3	591	1 E+6	1.09 E+6
048_ch2_S22_J3	546	1.95 E+7	2.99 +7
064_ch1_S22_J3	825	7.38 E+6	8.26 E+6
064_ch2_S22_J4	412	3.35 E+7	4.46 E+7

Table 8. Average coverage difference *D* between NSGAII and GISMOO for the instances of Set B

An examination of the six tables shows, once again, that the results are clearly in favour of GISMOO. The comparison with NSGAII (Tables 6, 8 and 10) shows that GISMOO has a larger covering difference for all instances except for four in Set X where the two algorithms have identical results. The comparison with PMS^{MO} (Tables 7, 9 and 11) shows the same advantage for GISMOO. These results confirm those obtained by the hyper-volume, which can be explained by the fact that the *D* metric calculations are based on those of the hyper-volume. With the hyper-volume results, the *D* metric results lead one to conclude that the solutions found by GISMOO have a better distribution than those found by NSGAII and PMS^{MO}.

Instance	Nb_cars	PMS ^{MO}	GISMOO
022_S22_J1	526	1.98 E+6	2.14 E+6
023_S23_J3	1110	2.77 E+6	2.80 E+6
024_V2_S22_J1	1270	3.02 E+6	3.05 E+6
025_S22_J3	1161	2.6 E+6	2.76 E+6
028_ch1_S22_J2	365	1.96 E+6	2.17 E+6
028_ch2_S23_J3	65	8.7 E+2	8.9 E+2
029_S21_J6	730	2.88 E+6	5.97 E+6
035_ch1_S22_J3	128	3.37 E+7	3.71 E+7
035_ch2_S22_J3	269	3.44 E+7	3.55 E+7
039_ch1_S22_J4	1231	1.09 E+8	1.13 E+8
039_ch3_S22_J4	1000	1.04 E+6	1.14 E+7
048_ch1_S22_J3	591	1.01 E +6	1.03 E +6
048_ch2_S22_J3	546	1.95 E+7	2.99 E+7
064_ch1_S22_J3	825	7.36 E+6	8.36 E+6
064_ch2_S22_J4	412	3.38 E+7	4.39 E+7

Table 9. Average coverage difference *D* between PMS^{MO} and GISMOO for the instances of Set B

Instance	Nb_cars	NSGAII	GISMOO
022_S49_J2	704	4.8 E+6	7.82 E+6
023_S49_J2	1260	5.97 E+6	7.04 E+6
024_S49_J2	1319	1.4 E+7	1.48 E+7
025_S49_J1	996	5.97 E+7	6.2 E+7
028_CH1_S50_J4	325	2.09 E+7	2.24 E+7
028_CH2_S51_J1*	65	0	0
029_S49_J5	780	7.84 E+7	9.62 E+7
034_VP_S51_J1_J2_J3	921	1.96 E+7	2.05 E+7
034_VU_S51_J1_J2_J3	231	2.51 E+7	3.45 E+7
035_CH1_S50_J4*	90	0	0
035_CH2_S50_J4	376	3.91 E+5	5.64 E+5
039_CH1_S49_J1*	1247	0	0
039_CH3_S49_J1	1037	1.94 E+7	2.16 E+7
048_CH1_S50_J4	519	2.88 E+7	3.02 E+7
048_CH2_S49_J5	459	1.54 E+7	1.65 E+7
064_CH1_S49_J1	875	8.31 E+6	8.97 E+6
064_CH2_S49_J4	273	2.34 E+6	2.37 E+6
655_CH1_S51_J2_J3_J4*	264	0	0
655_CH2_S52_J1_J2_S01_J1	219	2.63 E+5	2.92 E+5

Table 10. Average coverage difference *D* between NSGAII and GISMOO for the instances of Set X

Instance	Nb_cars	PMS ^{MO}	GISMOO
022_S49_J2	704	4.82 E+6	7.21 E+6
023_S49_J2	1260	5.91 E+6	6.97 E+6
024_S49_J2	1319	1.4 E+7	1.48 E+7
025_S49_J1	996	5.97 E+7	6.14 E+7
028_CH1_S50_J4	325	2.1 E+7	2.2 E+7
028_CH2_S51_J1*	65	0	0
029_S49_J5	780	7.85 E+7	1.16 E+8
034_VP_S51_J1_J2_J3	921	1.96 E+7	2 E+7
034_VU_S51_J1_J2_J3	231	2.5 E+7	3.39 E+7
035_CH1_S50_J4*	90	0	0
035_CH2_S50_J4	376	3.91 E+5	5.00 E+5
039_CH1_S49_J1*	1247	0	0
039_CH3_S49_J1	1037	1.94 E+7	2.08 E+7
048_CH1_S50_J4	519	2.88 E+7	3.01 E+7
048_CH2_S49_J5	459	1.54 E+7	1.6 E+7
064_CH1_S49_J1	875	8.31 E+6	8.97 E+6
064_CH2_S49_J4	273	2.34 E+6	2.37 E+6
655_CH1_S51_J2_J3_J4*	264	0	0
655_CH2_S52_J1_J2_S01_J1	219	2.63 E+5	2.83 E+5

Table 11. Average coverage difference *D* between PMS^{MO} and GISMOO for the instances of Set X

6. Comparison from a decision-making viewpoint

Besides the performance metrics used above, we also compare the three algorithms from a decision-making viewpoint. To graphically illustrate the results obtained, the comparison is done only with the HPO and COLOR objectives of the ICSP. We compare GISMOO with NSGAII and PMS^{MO}, and also with the results of the best team in the ROADEF Challenge 2005 (BEST_ROADEF). Remember that these latter results were obtained with a lexicographic ordering of the objectives. Moreover, the BEST_ROADEF results were obtained by optimizing the objectives in the order HPO-COLOR-LPO or COLOR-HPO-LPO. Finding these extreme solutions requires two distinct executions of the algorithm and thus requires a global execution time which is double the calculation time allocated to the three other algorithms.

Figure 6 presents a visual comparison of GISMOO, NSGAII, PMS^{MO} and BEST_ROADEF for executions of instance 022_3_4 (in Set A), for which the HPO constraints are “easy” to satisfy, according to Renault. This graphical representation confirms the results for the metrics in Section 5. It is clear that GISMOO’s Pareto set, for this instance, dominates those of NSGAII and PMS^{MO}. Indeed, the curve for GISMOO is definitely lower than those of NSGAII and PMS^{MO}. We recall here that the ICSP is a minimization problem. When GISMOO’s results are compared with BEST_ROADEF’s, we note that a single execution of GISMOO allows us to obtain the same solutions as two distinct executions of the algorithm used by the challenge’s winning team. As well, we see that GISMOO shows us several

alternative solutions ignored by the lexicographic treatment imposed by the Challenge rules. By giving too much importance to one objective, the lexicographic approach used by the ROADEF 2005 teams makes their algorithm converge towards an overly restricted zone of the search space.

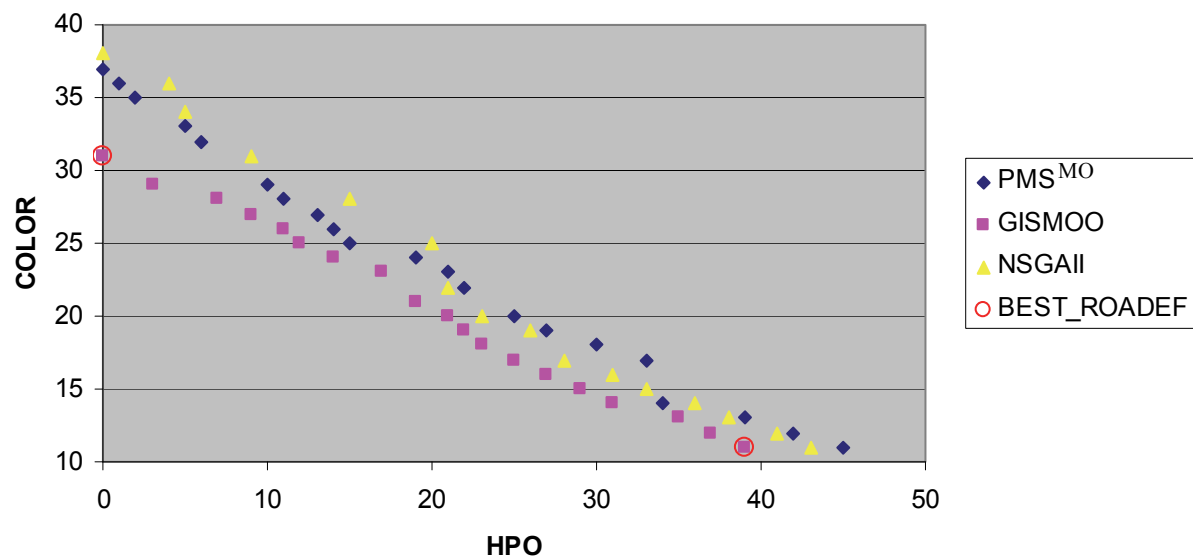


Fig. 6. Graphical performance comparison of GISM00, PMS^{MO}, NSGAI and BEST_ROADEF for the 011_3_4 instance

From a decision-making viewpoint, GISM00’s solution set offers greater latitude to a manager by presenting him with 19 alternative solutions, instead of the two extreme solutions proposed by BEST_ROADEF. For example, a “COLOR-oriented” manager could slightly lessen the importance of that objective and save two or even four HPO conflicts. In this case, the number of colour changes would increase from 11 to 13, but the number of HPO conflicts would decrease from 39 to 35. On the other hand, a manager oriented towards HPO conflict minimization would probably be interested to see the effect of lessening the importance of that objective on the number of purges. Thanks to the various solutions presented by GISM00, he would see that the number would diminish roughly in the same proportion. In fact, by permitting three more HPO conflicts (3) than the extreme solution (0), he would save three COLOR purges (28 instead of 31). GISM00’s solution set would allow still another manager with no preference between HPO and COLOR, to choose a balanced solution with about as many HPO conflicts (21) as colour changes (20).

It is important to note, however, that compromise solution sets cannot always be generated. Some of the ICSP instances proposed by Renault are such that all four algorithms give a unique solution optimizing all the objectives at the same time. This is the case for the instance 655_CH1_S51_J2_J3_J4 in Set X, as Table 12 shows. Each row of the table indicates, for each algorithm, the number of HPO conflicts, the number of colour changes and the number of LPO conflicts. The analysis of these results shows that the four algorithms, in all of their executions, give exactly the same solution.

Figure 7 presents a visual comparison of GISM00, NSGAI, PMS^{MO} and BEST_ROADEF for executions on the 035_ch2_S22_J3 (from Set B), which gives a problem for which the HPO constraints are “hard” to satisfy, according to Renault. As in the example presented in Figure 6, the Pareto set proposed by GISM00 clearly dominates those proposed by NSGAI and

PMS^{MO}. However, we note that the deviation between GISMOO and the two other Pareto algorithms is not as great as that observed by the 022_3_4 instance. This situation can be explained by the fact that the 035_ch2_S22_J3 instance has only 269 cars to schedule, whereas the 022_3_4 instance has 485. Nevertheless, we note that neither NSGAII nor PMS^{MO} can obtain the extreme HPO solution obtained by BEST_ROADEF. The best solutions found by NSGAII and PMS^{MO}, with HPO as the major goal, give 448 and 438 HPO conflicts, while GISMOO's best solution gives only 385 HPO conflicts. We can suppose that the difference between GISMOO and the other two Pareto algorithms would be even larger for a larger instance with HPO constraints that are "hard" to satisfy. Along with the two extreme solutions, GISMOO offers 70 other compromise solutions to a manager. However, contrary to what was observed for instance 022_3_4, we note that there is a large difference between the extreme HPO solution and the other solutions offered by GISMOO. It is possible to explain this difference by the difficulty in satisfying the HPO constraints for this instance.

GISMOO			PMS ^{MO}			NSGAII			BEST_ROADEF		
HPO	COLOR	LPO	HPO	COLOR	LPO	HPO	COLOR	LPO	HPO	COLOR	LPO
0	30	0	0	30	0	0	30	0	0	30	0

Table 12. Solution of GISMOO, PMS^{MO}, NSGAII and BEST_ROADEF for the 655_CH1_S51_J2_J3_J4 instance

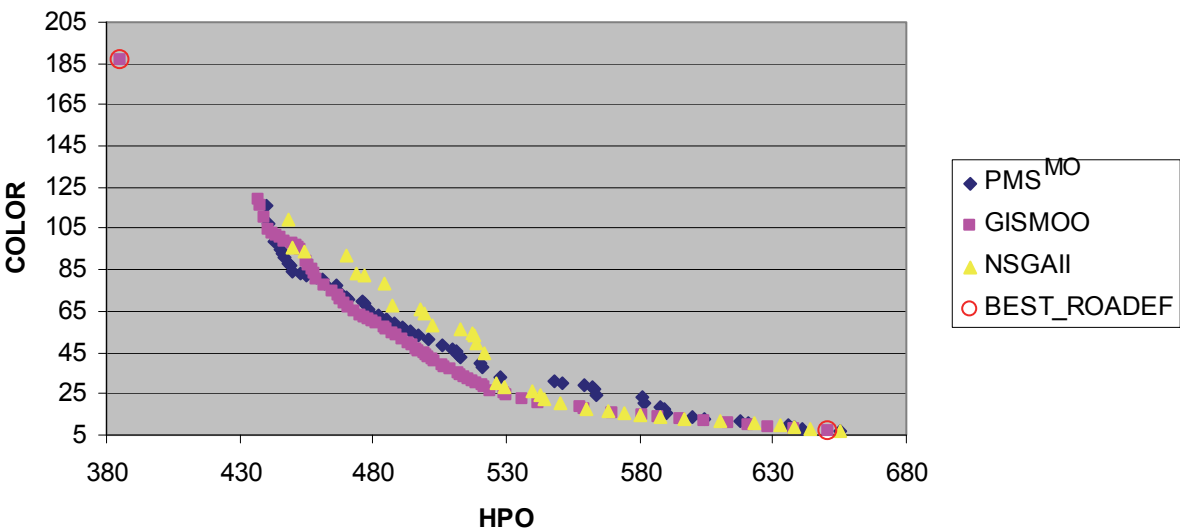


Fig. 7. Graphical performance comparison of GISMOO, PMS^{MO}, NSGAII and BEST_ROADEF for instance 035_ch2_S22_J3

Figure 8 presents a visual comparison of GISMOO, NSGAII, PMS^{MO} and BEST_ROADEF for another instance (048_ch2_S49_J5) for which the HPO constraints are "hard" to satisfy, but which have 546 cars to schedule. This graphical representation confirms the suppositions made for the preceding figure. We observe that GISMOO's solution set clearly dominates NSGAII's and PMS^{MO}'s solution sets: the difference between the curves is clearly larger for this instance than for instance 035_ch2_S22_J3 which has only about half as many cars. Figure 8 also shows that BEST_ROADEF's two solutions dominate GISMOO's solutions for instance

048_ch2_S49_J5. However, a closer look at the solutions reveals exactly the same value on the main objective. In fact, there are 3 HPO conflicts and 93 colour changes for BEST_ROADEF with HPO as the main objective, versus 3 HPO conflicts and 135 colour changes for GISMOO. With COLOR as the main objective, BEST_ROADEF obtains a solution with 58 colour changes and 282 HPO conflicts, versus 58 colour changes and 420 HPO conflicts for GISMOO. We recall that GISMOO’s execution time is about half that of BEST_ROADEF. If we give GISMOO and ROADEF the same time, the performance difference is considerably lessened, as Figure 9 shows. The extreme solutions of GISMOO are almost identical to those of BEST_ROADEF: 3 HPO conflicts and 94 colour changes for BEST_ROADEF with HPO as the main objective, versus 3 conflicts and 93 changes for GISMOO, and with COLOR as the main objective, 58 changes and 284 conflicts for GISMOO versus 58 changes and 283 conflicts for BEST_ROADEF. In addition, GISMOO offers 7 compromise solutions to a decision maker.

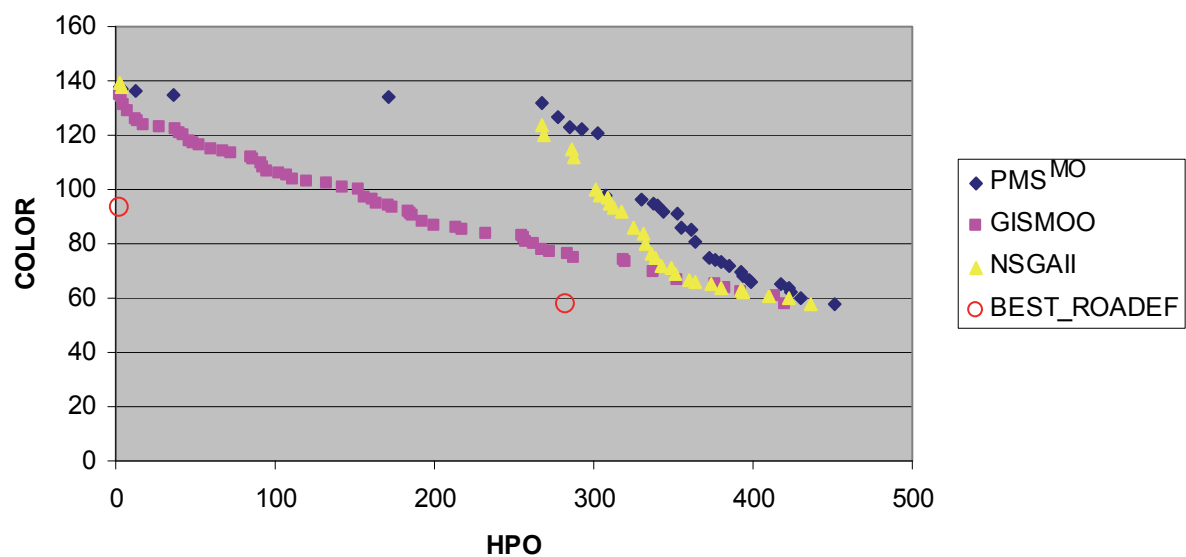


Fig. 8. Graphical performance comparison of GISMOO, PMS^{MO}, NSGAI and BEST_ROADEF for instance 048_ch2_S49_J5

7. Conclusion

In this chapter we have presented an evolutionary Pareto algorithm, GISMOO, to solve the multi-objective industrial car sequencing problem. The biggest difference between GISMOO and other multi-objective EAs mainly lies in the way in which the immune metaphor is used in a Pareto EA to identify and emphasize the solutions located in less crowded regions of the search space. Even if EAs are known to be well suited for multi-objective optimization in Pareto sense, few researchers and industrials decided to use this category of algorithms to solve the ICSP. This situation may be explained by the fact that the ICSP is generally considered as a problem with several goals lexicographically ordered and not from the Pareto viewpoint. However, the lexicographical treatment of the objectives is such that it can eliminate several “interesting” solutions for the manufacturer. Indeed, the relaxation of the importance granted to the main objective can highlight other attractive solutions. One original effect of this use of GISMOO is that a Pareto algorithm considers the objectives of the problem integrally, without ordering them or giving them weights. To our knowledge, this problem has generally been considered from a one-goal viewpoint, or by

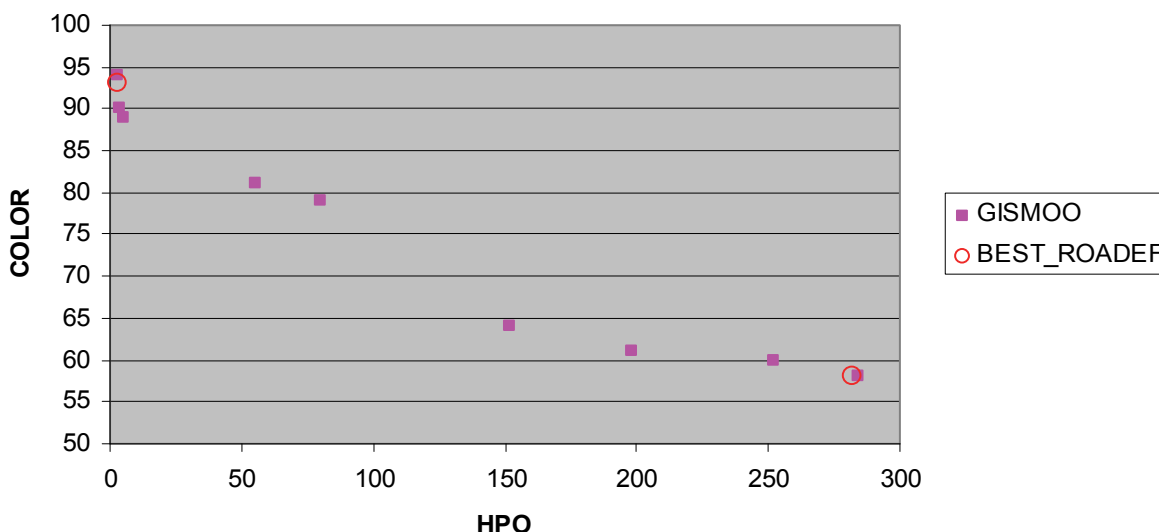


Fig. 9. Graphical performance comparison of GISMOO and BEST_ROADEF for instance 048_ch2_S49_J5, when the same execution time is allocated to the two algorithms

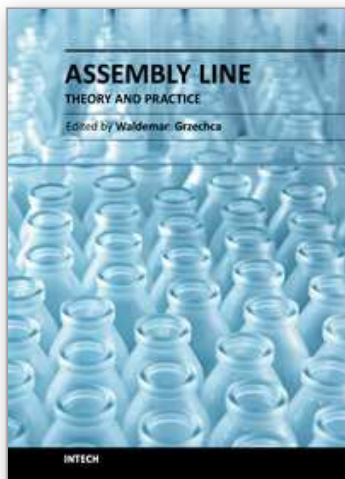
ordering the goals lexicographically, or by testing only small-size instances. The results obtained by GISMOO for a variety of test cases of the ICSP have showcased compromise solutions that were not obtainable by a lexicographic treatment. Instead of a single solution in function of an *a priori* ordering of goals, GISMOO offers the decision maker a panoply of solutions from which he can choose according to his own preferences. Another positive effect of using GISMOO is to bridge the gap between theoretical approaches and practical situations for this type of industrial problem. Indeed, the experimental results confirm the excellent performance of our algorithm in coping with a real-life situation.

In future work, we will seek to extend the application field of GISMOO to others multiple-objective problems such as other industrial problems or continuous test problems with or without constraints.

8. References

- Benoist, T. (2008). Soft car sequencing with colors: Lower bounds and optimality proofs. *European Journal of Operational Research*, Vol. 191, No. 3, pp. 957-971.
- Briant, O., D. Naddef & G. Mounié (2008). Greedy approach and multi-criteria simulated annealing for the car sequencing problem. *European Journal of Operational Research*, Vol. 191, No. 3, pp. 993-1003.
- Cordeau, J. F., G. Laporte & F. Pasin (2008). Iterated Tabu Search for the Car Sequencing Problem. *European Journal of Operational Research*, Vol. 191, No.3, pp.945-956.
- De Castro, L. N. and J. Timmis (2002). *Artificial Immune Systems: A New Computational Intelligence Approach*. London.
- de Oliveira dos Reis, R. J. (2007). *Solving the Car Sequencing Problem from a Multiobjective Perspective* Lisbonne, Universidade Técnica de Lisboa. Master's thesis.
- Deb, K. (2000). A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multiobjective Optimization : NSGA II. *Parallel problem Solving form Nature – PPSN VI*, M. Schoenauer et al. (Eds), Springer Lecture Notes in Computer Science, p. 849-858.
- Estellon, B., F. Gardi & K. Nouioua (2008). Two local search approaches for solving real-life car sequencing problem. *European Journal of Operational Research*, Vol. 191, No. 3, pp.928-944.

- Francisci, D. (2002). *Algorithmes évolutionnaires et optimisation multi-objectifs en data mining*, Laboratoire informatique, signaux et systèmes de Sophia Antipolis UMR 6070.
- Gottlieb, J., M. Puchta & C. Solnon (2003). A study of greedy, local search and ant colony optimization approaches for car sequencing problems. *Computers Science*: pp.246-257.
- Knowles, J. D. & D. W. Corne (2000a). M-PAES : A Memetic Algorithm for Multiobjective Optimization. In *Proceedings of the 2000 Congress on Evolutionary Computation*, p. 325-332.
- Knowles, J. D. & D. W. Corne (2000b). The Pareto-Envelope based Selection Algorithm for Multiobjective Optimization. In *Proceedings of the Sixth International Conference on Parallel Problem Solving from Nature (PPSN VI)*, pp. 839-848, Berlin.
- Nguyen, A. & V.-D. Cung (2005). Le problème du Car Sequencing Renault et le challenge ROADEF' 2005, *Proceedings of Journées Francophones de Programmation par Contraintes*, 3-10.
- Ribeiro, C. C., D. Aloise, T.F. Noronha, C. Rocha & S. Urrutia (2008). An efficient implementation of a VNS/ILS heuristic for a real-life car sequencing problem. *European Journal of Operational Research*, Vol. 191 No. 3, pp. 596-611.
- Schott (1995). *Fault tolerant design using single and multicriteria genetic algorithm optimization*. Department of Aeronautics and Astronautics, Massachusetts Institute of Technology. Master's thesis.
- Solnon, C., V.-d. Cung, A. Nguyen & C. Artigues (2008). The car sequencing problem: overview of state-of-the-art methods and industrial case-study of the ROADEF'2005 challenge problem. *European Journal of Operational Research*, Vol 191, No. 3, pp.912-927.
- Zinflou, A., C. Gagné & M. Gravel (2007). Crossover operators for the car-sequencing problem. *Seventh European Conference on Evolutionary Computation in Combinatorial Optimisation (EvoCOP 2007)*, LNCS 4446, C. Cotta and J. van Hemert (Eds.), Springer-Verlag Berlin Heidelberg, pp. 229-239.
- Zinflou, A., C. Gagné & M. Gravel (2008a). Design of an efficient genetic algorithm to solve the industrial car sequencing problem. *Advances in Evolutionary Algorithms*, Advances in Evolutionary Algorithms, I-Tech Education and Publishing, Vienna, Austria, Witold Kosinski Editor, Chapter 19, pp. 377-400, November 2008, ISBN 953761911-7.
- Zinflou, A., C. Gagné, M. Gravel & W.L. Price (2008b). Pareto memetic algorithm for multiple objective optimization with an industrial application. *Journal of Heuristics* Vol. 14, No. 4, pp. 313-333.
- Zinflou, A., C. Gagné & M. Gravel (2009). Solving the industrial car sequencing problem in a Pareto sense. *Proceedings of the 12th International Workshop on Nature Inspired Distributed Computing*, Rome, Italy.
- Zinflou, A., C. Gagné & M. Gravel (2011). GISMOO: A New Hybrid Genetic/Immune Strategy for Multiple Objective Optimization. (submitted).
- Zitzler, E., M. Laumanns & L. Thiele (2001). *SPEA2 : Improving the Strength Pareto Evolutionary Algorithm*, Technical Report 103, Computer Engineering and Communication Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich, Gloriastrasse 35, CH-8092 Zurich, Switzerland.
- Zitzler, E. & L. Thiele (1999). Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation*, Vol.3, pp. 257-271.



Assembly Line - Theory and Practice

Edited by Prof. Waldemar Grzechca

ISBN 978-953-307-995-0

Hard cover, 250 pages

Publisher InTech

Published online 17, August, 2011

Published in print edition August, 2011

An assembly line is a manufacturing process in which parts are added to a product in a sequential manner using optimally planned logistics to create a finished product in the fastest possible way. It is a flow-oriented production system where the productive units performing the operations, referred to as stations, are aligned in a serial manner. The present edited book is a collection of 12 chapters written by experts and well-known professionals of the field. The volume is organized in three parts according to the last research works in assembly line subject. The first part of the book is devoted to the assembly line balancing problem. It includes chapters dealing with different problems of ALBP. In the second part of the book some optimization problems in assembly line structure are considered. In many situations there are several contradictory goals that have to be satisfied simultaneously. The third part of the book deals with testing problems in assembly line. This section gives an overview on new trends, techniques and methodologies for testing the quality of a product at the end of the assembling line.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Arnaud Zinflou and Caroline Gagne (2011). Tackling the Industrial Car Sequencing Problem Using GISMOO Algorithm, Assembly Line - Theory and Practice, Prof. Waldemar Grzechca (Ed.), ISBN: 978-953-307-995-0, InTech, Available from: <http://www.intechopen.com/books/assembly-line-theory-and-practice/tackling-the-industrial-car-sequencing-problem-using-gismoo-algorithm>

INTech
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2011 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](https://creativecommons.org/licenses/by-nc-sa/3.0/), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen