Application of Strategies on NSGAII for Searching of Optimal Solutions to the Car Sequencing Problem

Aplicación de Estrategias Sobre NSGAII Para Búsqueda de Soluciones Optimas al Problema de Secuenciación de Vehículos

GUZMÁN-SANDOVAL, Alonso Eric[1]†*, PUGA-SOBERANES, Héctor José[1], SORIA-ALCARAZ Jorge Alberto[2] and CARPIO-VALADEZ, Juan Martín[1]

[1]Tecnológico Nacional de México/ Instituto Tecnológico de León, México.
[2]Universidad de Guanajuato, División de Ciencias Económico-Administrativas

ID 1st Author: *Alonso Eric, Guzmán-Sandoval* / **ORC ID:** 0000-0002-7204-8850, **CVU CONACYT ID:** 1076451

ID 1st Co-author: *Héctor José, Puga-Soberanes* / **ORC ID:** 0000-0002-1315-1436

ID 2nd Co-author: *Jorge Alberto, Soria-Alcaraz* / **ORC ID:** 0000-0002-8602-6150

ID 3rd Co-author: *Juan Martín, Carpio-Valadez* / **ORC ID:** 0000-0002-1191-2676

DOI: 10.35429/EJDRC.2022.14.8.1.13 Received March 26, 2022; Accepted June 30, 2022

Abstract

One of the main conflicts in a car production plant is to deliver the orders received daily in a timely manner, which are not uniform and involve a large amount of human and material resources. The car sequencing problem is a NP-Hard problem that consists of finding the sequence of cars that minimizes the number of constraint violations in an assembly line. The problem can be approached from a mono-objective or multi-objective point of view. The objective of this paper is to treat a case study of this problem, presented at ROADEF 2005, from the multi-objective Pareto approach, taking the NSGAII algorithm as a basis for a proposal scheme and verifying its feasibility. A systematic and general improvement of the quality of the final Pareto fronts is verified, and the results of the implementation of a strategy scheme that consists of the initialization of the population guided by local search, and specialized crossover and mutation operators are reported. These results allow us to give continuity to the generation of an optimization proposal for the vehicle sequencing problem.

Evolutionary Algorithms, Multi-Objective Optimization, Car Sequencing

Resumen

Uno de los principales conflictos en una planta de producción de automóviles es entregar en tiempo y forma los pedidos recibidos diariamente, los cuales no son uniformes e involucran una gran cantidad de recursos humanos y materiales. El problema de secuenciación de automóviles es un problema de complejidad NP-Duro que consiste en encontrar la secuencia de automóviles que minimice el número de violaciones de restricciones de una línea de ensamblaje. El problema se puede abordar desde un punto de vista monobjetivo o multiobjetivo. En este trabajo se tiene como objetivo tratar un caso de estudio de este problema, presentado en el ROADEF 2005, desde el enfoque multiobjetivo de Pareto, tomando como base para un esquema de propuestas el algoritmo NSGAII y verificar su factibilidad. Se constata una mejora sistemática y general de la calidad de los frentes de Pareto finales, y se reportan los resultados de la implementación de un esquema de estrategias que consta de la inicialización de población guiada por búsqueda local, y operadores de cruza y de mutación especializados. Estos resultados permiten dar continuidad a la generación de una propuesta de optimización al problema de secuenciación de vehículos.

Algoritmos Evolutivos, Optimización Multiobjetivo, Secuenciación de Automóviles

Citation: GUZMÁN-SANDOVAL, Alonso Eric, PUGA-SOBERANES, Héctor José, SORIA-ALCARAZ Jorge Alberto and CARPIO-VALADEZ, Juan Martín. Application of Strategies on NSGAII for Searching of Optimal Solutions to the Car Sequencing Problem. ECORFAN Journal-Democratic Republic of Congo. 2022, 8-14: 1-13

* Author's Correspondence (e-mail: aeguzsando@gmail.com)
† Researcher contributing first author.

Introduction

The *car sequencing problem* (CSP) was presented by (Parello & Kabat, 1986) as a constraint satisfaction problem in automotive factory workshops. Usually, it is described as the ordering of a sequence of vehicles within an assembly line, where a series of installation options must be assembled to each vehicle with the objective of minimizing the number of violations to restrictions in all assembly stations. Since 1990, numerous works have been reported the application of metaheuristics with a mono-objective tackling, using strategies based on constraint logic programming (Guerre-Chaley, Frein, & Bouffard-Varcelli, 1995), genetic algorithms (Warmick & Tsang, 1995) and taboo search (Zufferey, 2016), among others.

In 2005 a case study based on the CSP was presented at the ROADEF challenge. In this variation, the constraints of the original CSP are considered and identified as smoothing constraints, moreover adding a third variable related to color changes. Thus the case study tackles three optimization objectives listed right away: *color changes, high-priority overloads, and low-priority overloads*. Several proposals were presented in the challenge, highlighting *greedy strategies, local search, genetic algorithms and simulated annealing*, among others, which apply a *lexicographic multiobjetive approach*. The work of (Nguyen & Cung, 2005) resumes the best techniques applied to this case study and their results in the challenge.

Generally, performing a deep review of the state of the art, we find out that lexicographic approximations have been applied in most works related to the *CSP*. Such is the work presented by (Marquez-Sanchez & Puga-Soberanes, 2019) that exposes an exhaustive study of the *CSP*, as well as a lexicographic strategy based on differential evolution with a chromosomic repairer. Based on the results and observations of the previous work, (Velazquez & Puga-Soberanes, 2020) presents a feasibility study for the application of ~~a~~ *estimation of distribution algorithm* (EDA) with ~~a~~ lexicographic approach, concluding that it is a potential strategy to find feasible and optimal solutions to the problem.

However, in the state of the art exists several works that explore the problem from the multi-objective sense of *Pareto*, characterized by treating the variables of a multi-objective optimization problem simultaneously and independently. The work of (Zinflou, 2008) tackles the *CSP* with various techniques (including the lexicographic and Pareto sense), although its main contribution is the formulation and implementation of a new multiobjetive evolutive algorithm called PMS^{MO}, also applying specialized heuristics for the generation of initial population guided by greedy heuristics and recombination and mutation operators for the individuals of the population.

On the other side, the work of (Chutima & Olarnviwatchai, 2018) proposes a new case study extended from the ROADEF 2005 case, presenting additional objectives and beign tackled by a novel multi-objective EDA with a Pareto sense called COIN.

In order to provide continuity to the CSP study and the applicaton of evolutive techniques with a Pareto approach, this paper proposes a scheme of strategies to obtain optimal solutions to the problem and analyzes its effects on the performance obtained, taking as a basis the *NSGAII* algorithm (Deb, 2002) and using instances of the ROADEF 2005 challenge. We first present a formal definition and description of the characteristics of the selected performance function model for the CSP. Next, the methodology used will be described, including the strategic scheme and the details of the instances used. Finally, the results obtained from the implementation of the proposed strategies and their corresponding statistical analysis are reported to form several conclusions.

Problem description

According to (Parello & Kabat, 1986), the approach of the classic CSP indicates the search of a sequence of cars that minimizes the total number of violations to the installation constraints in the workstations of an assembly line. In the ROADEF variant, the constraints are divided in two types: soft smoothing and hard smoothing. In addition, a third variable related to the minimization of color changes is introduced.

ISSN 2414-4924

The established assembly constraints in each station are denoted by the ratio N/P, where N is the maximum number of cars that the station can serve without generate a volation and P is the total number of cars that always occupy the station. In synthesis, the violations (or overloads) in the station are counted only as the number of cars requiring service from the station that exceeds N.

Regarding the color changes that occur through the assembly line, these are counted taking care that there are no sub-sequences of cars of the same color that exceed the constraint of a maximum number of consecutive cars of the same color, denoted by B.

In this CSP model, a collection of cars S is the union of a set of cars D_P, that was not serviced from the previous day, with the set of cars D of the current day planned on the assembly line:

$$S = D_P \cup D = \{s_1, s_2, \dots, s_n\}, \quad (1)$$

Each car $s \in S$ has an associated vector of installation requirements taken from the set $O = HO \cup LO$, identifying high-priority restrictions by set HO, related to critical installation options in the assembly line as they require a strong load of work; and low priority constraints denoted by the set LO, related to installation options, considered as non-critical:

$$O = HO \cup LO = \{o_1, o_2, o_3, \dots o_L\} \quad (2)$$

In addition, a set of colors is established, described as:

$$C = \{c_1, c_2, \dots, c_M\}, \quad (3)$$

Of the mentioned sets, an association function links for each car $s \in S$ a color from the set C:

$$c: S \to C, \quad (4)$$

and a requirement function for a vector of options, defined as:

$$r: S \times O \to \{0,1\}$$

$$(s_i, o_j) \to r_{ij}, \quad (5)$$

$$r_{ij} = \begin{cases} 1 & if\ o_j\ is\ installed\ on\ s_i, \\ 0 & otherwise, \end{cases}$$

where s_i represents the ith car of a sequence of S into a station and o_j the j-th installation option required by the car.

Given a sequence of all the elements of S, randomly ordered, expressed as a vector $\vec{s} = [s_1, s_2, \dots, s_n]$, the cost function is defined as:

$$F(\vec{s}) = (F_1(\vec{s}), F_2(\vec{s}), F_3(\vec{s})), \quad (6)$$

where $F_1, F_2, F_3 \in \{FC, FH, FL\}$ and FC, FH and FL correspond to the cost functions of color changes, high priority constraints and low priority constraints respectively. The FC function only counts color changes of cars given in the sequence, while the FH and FL functions count constraint violations using the generic expression:

$$\sum_{i=1}^{|O|} FE_i(\vec{s}), \quad (7)$$

where $FE_i(\vec{s})$ represents the cost function applied to the i-th workstation and it can be replaced by any of the FH_i or FL_i functions for high or low priority constraints respectively.

To calculate the overloads, the sliding window mechanism proposed by (Bolat & Yano, 1992) was used, assuming that the windows W_k have a size equal to k. The size of each window depends on the ratio constraints N_i/P_i, wich is unique in the ith position of the sequence. The size of the first set of windows and the middle windows is always equal to the denominator P_i,whereas the size of the last windows is adjusted by:

$$NewP_i = P_i - 1, \quad (8)$$

$$k = N_i + 1, \quad (9)$$

where $NewP_i$ denotes only the size of the first window of the set of final windows. By assumption, in the last windows the value of P_i will exceed the total size of the vector $\vec{s}$, so this value should decrease as it progresses until it reaches a value equal to k, which is the value of the last window of the set of final windows. In this way, a slip of W_k (in a sequence of k consecutive vehicles) calculates the overloads through the expression:

$$FE_i(\vec{s}, W_k) = max\{\left(\sum_{i=1}^{|E|} E_i(s)\ in\ W_k\right) - N_i\}, \quad (10)$$

where E represents the count of occupied cars in a station, evaluating the frtness function on the position of the car s. Since it is substraction, the value of the overloads on the ratio N_i/P_i could be $FE(S, W_k) < 0$, in which case the number of overloads in the slip automatically becomes 0.

With the given information about the model, the fitness evaluation function for each workstation is calculated with the expression:

$$FE_i(\bar{s}) = \sum_t^v FE_i(\vec{s}, W_{P_i}) + \sum_k^{NewP_i} FE_i(\bar{s}, W_{NewP_i}), \quad (11)$$

Next, v and t represent the limits of the final windows where v is constant and t is variable, such that $t \to v$. $|c'|$ corresponds to the total number of cars from the given starting position in the sequence, while $\#n$ is the number of cars in each window and c corresponds to the starting position of the first window. The size of the window is reduced after counting by the window until $t = v$. This is done with the following expressions:

$$t = |c'| - \#n, \quad (12)$$

$$v = c - NewPi \quad (13)$$

Finally, the problem consists of minimizing the cost function (6), where equations (7) and (11) are applied in the objective functions of constrains for FH, FL and taking into account the third objective function FC that only counts the color changes given in the sequence without violating a constraint B. To manage the information of the problem instances, the cars with the same requirement were grouped into classes cl_i.

Options				**Classes (cl_i)**					
O	**Prior.**	N_i	P_i	**1**	**2**	**3**	**4**	**5**	**6**
o_1	1	1	2	0	1	1	0	0	0
o_2	1	2	5	1	0	1	0	1	1
o_3	0	1	3	0	1	0	0	0	0
o_4	0	3	5	0	0	0	1	0	1
o_5	0	2	3	0	1	1	0	1	0
Cars: $\|S\| = 25$				5	5	4	4	3	4
#Colors(C)			c_1	2	1	1	2	1	1
			c_2	1	1	0	2	1	1
			c_3	1	3	2	0	0	2
			c_4	1	0	1	0	1	0

Table 1 Generic form of a CSP instance with a sequence of 25 cars *(Zinflou, Design of an Efficient Genetic Algorithm to Solve the Industrial Car Sequencing Problem, 2008).*

Table 1 illustrates the case of an instance that has a sequence of 25 cars cataloged in 6 classes, defined by the shape of the requirements vector. The priority of each option of the set O is indicated with 1 for high and 0 for low, as well as the ratio N_i/P_i for each one. In the classes section, the respective requirement vectors for each one are shown, while in the *Cars* row, the number of cars $|S|$ is indicated, followed by the number corresponding to each class cl_i. Finally, at the bottom of the table, the set of colors and the number of cars per class painted of a specific color are indicated; for example, for class 1, two cars will be painted in the color c_1, and the remaining three cars will be painted by colors c_2, c_3 and c_4 respectively.

Proposed methodology

The research methodology initially consisted of the implementation of the NSGAII algorithm as the basis for a framework of proposed strategies to tackle the CSP, previously establishing an adequate individual coding and a population initialization mechanism that helps to enhance the search process from the beginning. Subsequently, the improvement strategies were defined to guide the search of optimal solutions in the solution space, specifically the crossover and mutation operators for searching intensification and solution diversification. In the design of experiments, five different variants of NSGAII were established to systematically test each proposal, and finally we established a statistical analysis of the results of the experimentation based on the measurement of *hypervolume* (Zitzler, Laumanns, & Thiele, 2001).

a. NSGAII

Formally, the version implemented in this work was the standard generational *NSGAII* proposed by (Deb, 2002), characterized by the generation of two populations: one of the parents and the other of offsprings, denoted as P and Q respectively, which are combined in the set R, maintaining the same size in each iteration of the evolutionary process. In the algorithm a method called *non-dominated sort* is used to catalog each *non-dominated front* found. In addition, a density estimator called *Crowding Distance* (*CD*) is used to generate diversity in the solutions of non-dominated fronts.

Both methods are the main criteria for the selection of individuals from the population that will enter into a evolutionary component executed by a generic genetic algorithm.

The general procedure, illustrated in Figure 1, starts with a population R that contains the populations of parents P_t and children Q_t (empty in the first generation), with t corresponding to a generation of the evolutionary process.

The population R is ordered (via Fast Non-dominated Sort function) according to the different fronts of non-dominated solutions found (denoted as F) and the density of individuals of each element is estimated (via Crowding Distance density estimation function). For a new population P_{t+1}, only the elements belonging to low-level non-dominated fronts are taken (the best in terms of minimization) and, in the case of a tie, the one with the lowest density is taken. The non-dominated solutions ordered on different fronts become the input of a genetic algorithm (GA), resulting in the population of children Q_{t+1}.

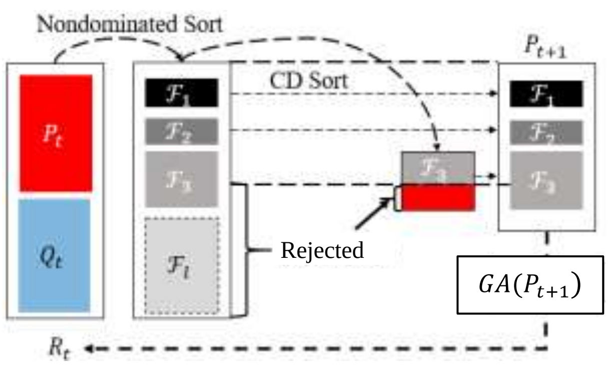


Figure 1 Scheme of the general procedure of NSGAII *(Deb, 2002)*

b. Initial population generation

In the process of generating the initial population, a random sample vector $\vec{s}_0$ of all cars is taken from the collection of cars S. This sequence is segmented into subsequences of approximately the same size, in such a way that the restriction B is not violated. In our case, the size of each subsequence is equal to the minimum value of B, beign generalized to all the instances used. After the segmentation The result is a sequence of subsequences denoted by $\hat{s}_0$.

Given the segmentation, the process perform two main steps based on guided local search (GLS) and random generation. In the first step, the fitness of the initial solution is previously calculated, denoted as F_0, and generates the first half of the initial population creating a search neighborhood with a population of empty cadidate receptors y with the same size of $\hat{s}_0$. To create those candidate receptors we seek to perturb the sequence $\hat{s}_0$. by randomly taking half of its subsequences and placing them one by one in their corresponding positions.

When a subsequence is placed there is a 10% chance of shuffling its elements, the rest of the subsequences of $\hat{s}_0$ are stored in a list denoted by L. To occupy the remaining positions of y we seek to place as many elements of the list L as possible in empty positions of y in such a way they do not generate violations in ratio constraints. The rest of the empty positions of $y$are randomly occupied with the rest of the elements of L; this process is repeated as many times to complete the neighborhood NH.

Once completed the neighborhood, the fitness of each $y \in NH$ is evaluated and compared with F_0, if its fitness is greater it is accepted into the intial population P. The process is repeated until $|P| = \lceil N/2 \rceil$. Finally, in the second step, corresponding to the second half of the initial population, the sequence $\vec{s}_0$ is taken and permuted to generate a new individual, repeating the process until $|P| = N$. The generation procedure is illustrated in detail in Figure 2.

Input: $P \leftarrow \emptyset, N, \vec{s}$
$\hat{s}_0 \leftarrow$ segment($\vec{s}$)
Evaluate fitness of initial solution $F_0 \leftarrow F(\vec{s})$
$NH \leftarrow \emptyset$ Generate empty search neighborhood
While $P \leq N/2$
For $i = 1, i \leq N$
Generate receptor y with size $|\hat{s}_0|$
$L \leftarrow \emptyset$
For $j = 1, j < |\hat{s}_0|/2$
$pos_{rand} \leftarrow$ random position of $\hat{s}_0$
If $y[pos_{rand}] = \emptyset$, entonces
$y[pos_{rand}] \leftarrow \hat{s}_0[pos_{rand}]$
$\alpha \leftarrow$ random number between $[0, 1]$
If $\alpha < 0.1$, the shuffle $y[pos_{rand}]$
$L \leftarrow \hat{s}_0 - \hat{s}_0[pos_{rand}]$
For $j = 1, j < |\hat{s}_0|$
If $y[j] = \emptyset$
$l_{rand} \leftarrow$ Take random position from L
If $L[l_{rand}]$ in $y[j]$ does not conflict, then
$y[j] \leftarrow L[l_{rand}]$
$L \leftarrow L - L[l\_rand]$
Locate remaining elements from L into empty spaces of y
$NH \leftarrow NH \cup Y_i$
For each $y \in NH$
Si $F(y) < F_0$
$P \leftarrow P \cup y$
$i \leftarrow N/2 + 1$
While $i \leq N$
$y \leftarrow$ Shuffle $\vec{s}$
$P \leftarrow P \cup Y_i$
$i = i + 1$
Return P

Figure 2 Algorithm 1 Guided Local Search Based Initialization Proposal Procedure.

Figure 3 shows an example of the procedure to generate an individual for the initial population, composed of a sequence of 15 cars. The process begins by taking the initial sequence, which in the first step is segmented into five subsequences. In the second step the subsequences seq_1 and seq_3 are randomly selected and placed in their original positions into the receiver y, meanwhile the rest of subsequences are saved into the wait list L.

In the third step, of all the subsequences evaluated in each free position, the only one that was possible to place in the second position of the receiver y was the subsequence seq_4, because when it was evaluated at that position it did not cause any conflict of the ratio constraints violations.

Likewise, under the probability of 10%, this subsequence was shuffled after being placed. The wait list L keeps storing the subsequences seq_2 and seq_5 to finally begin placed in the remaining random empty positions in the fourth step.

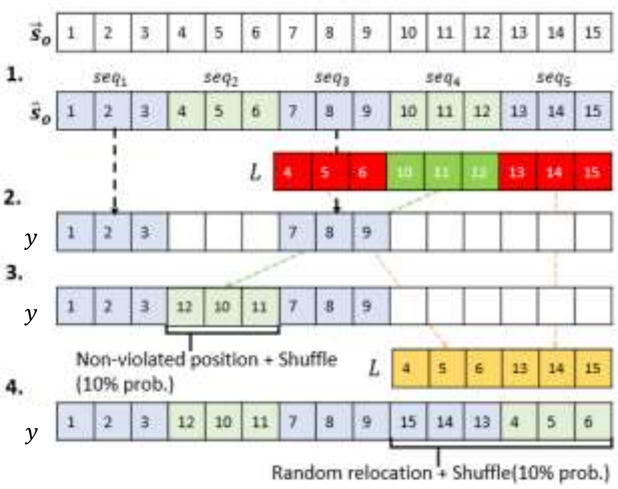


Figure 3 Schematic of the procedure of generation of individuals for the search neighborhood.

c. Recombination

The recombination operator proposal is based on the order-based crossover (*OX2*), described by (Syswerda, 1991), applying CSP-specific proportion constraint and color change satisfaction functions, which makes it an ad hoc recombination method specifically formulated for this problem.

The first step of the proposed operator, illustrated in Figure 4, consists of the selection of an individual$Parent_1$; then, sections of consecutive elements and their positions in $Parent_1$ that do not violate hard or soft constraints are located and stored with their original positions inside the descendant H, of size $|Parent_1|$, and those elements that generate conflict are stored in a wait list L.

The same evaluation process is performed on $Parent_2$, looking up for the original positions, but excluding those that are already inside or whose original positions are already occupied in H. The elements of L are inserted into H starting from the first empty position and going forward evaluating that they do not conflict at any position or else they remain in L. The remaining elements are inserted in the free positions of H starting from the first empty position found.

ISSN 2414-4924

The entire process is repeated starting with the indvidual $Parent_2$ for a second descendant.

Input: $Parent_1, Parent_2$
1. Create empty receptor H with size $|Parent_1|$
2. $L = \emptyset$
3. For $i \leftarrow 1, i \leq |H|$
4. If $check\_conflicts(Padre_1[i]) = False$, then
5. $H[i] \leftarrow Parent_1[i]$ (Takes original position in H)
6. Else, $L \leftarrow L \cup Parent_1[i]$ (Stored in L)
7. For $i \leftarrow 1, i \leq |H|$
8. If $Parent_2[i] \notin H \wedge H[i] = \emptyset$, then
9. Repeat steps 4-6 with $parent_2$
10. For $i \leftarrow 1, i \leq |H|$
11. If $H[i] = \emptyset$, then
12. $H[i] \leftarrow L[i]$ (Locates $L[i]$ in first empty position)
13. $L = L - L[i]$
14. Return H

Figure 4 Algorithm 2. Ad Hoc Recombination Operator Proposal Procedure for CSP.

Figure 5 ilustrates a basic example of the described procedure for two solutions consisting of 8 cars. A conflict is indicated with the value of 1 in the third and fourth rows, which corresponds to conflicts of color and installation proportions.

$Parent_1$

auto	0	3	4	7	5	1	6	2
Color	1	1	2	3	2	2	2	4
C_{cc}	0	0	1	1	0	0	0	1
C_o	0	1	0	0	1	0	0	0

$H^{[1]}$ $L_1 = \{2,3,4,5,7\}$

auto	0					1	6	
Color	1					2	2	
C_{cc}	0					0	0	
C_o	0					0	0	

$Parent_2$

auto	3	7	4	1	5	6	0	2
Color	1	3	2	2	2	2	1	4
C_{cc}	1	1	0	0	0	0	1	1
C_o	0	0	0	1	0	0	0	0

$H^{[2]}$ $L = \{2,3,7\}$

auto	0		4		5	1	6	
Color	1		2		2	2	2	
C_{cc}	0		0		0	0	0	
C_o	0		0		0	0	0	

$H^{[3]}$

auto	0	3	4	7	5	1	6	2
Color	1	1	2	3	2	2	2	3
C_{cc}	0	0	1	1	1	0	0	1
C_o	0	0	0	0	0	0	0	1

Figure 5 Scheme of the recombination procedure

For $Parent_1$, it is not observed conflicts in positions 0, 5 and 6, corresponding to the cars 0, 1 and 6. These take their original positions in $H^{[1]}$ and a wait list L is filled with the cars 2, 3, 4, 5 and 7. For $Parent_2$, it cannot be seen any indicated conflict in the positions for the cars 4, 5 and 6, but the car 6 is already present in position 7 of $H^{[1]}$. Therefore, only the cars 4 and 5 are placed in their original positions in $H^{[2]}$. Finally, the car 3 have only one color conflict, however if it is placed in position 2 of $H^{[3]}$, this conflict is canceled. The remaining elements of L are taken in order (i.e. according to their original order in $Parent_2$), starting with the car 7, which is placed in position 4, inevitably generating a color conflict, and the car 2 in the last position, also generating a conflict of color and proportion. Thus, the contrast between the amount of conflicts of the two parents with $H^{[3]}$ is appreciated, decreasing considerably in proportion violations, while in color changes the same amount of $Parent_2$ remains.

d. Mutation

There are mutation operators in the state of the art built specifically for various task scheduling problems, including the CSP, some of them are mentioned by (Chutima & Olarnviwatchai, 2018) and (Zinflou, Gagné, & Gravel, 2013). In this work, two techniques were implemented seeking the balanced intensification of the search for optimal solutions, applying it after the recombination process. Both methods, denoted by M_A and M_B, are part of an unique mutation operator denoted by M, establishing a selection threshold ($\tau = 0.5$). When comparing a random value θ with τ, it is established the mutation method for the individual H. Given this idea, the function of this operator is defined as:

$$M(H) = \begin{cases} M_A & si\ \theta < \tau, \\ M_B & otherwise, \end{cases} \quad (14)$$

where the $M(H)$ corresponds to the function of the mutation operator over H, M_A corresponds to the method "batch swap with invertion" (*BSWI*) and M_B corresponds to the method "invertion with swapping "(*IWS*). The M_A-BSWI method (see Figure 6) seeks to intensify the search for solutions that prioritizes the objective of color changes.

This works by taking the child H, dividing it into several parts of the same size (which, as in the initialization proposal, is generalized to the minimum value of B all the instances used), resulting in a series of subsequences denoted by H'. Next, two of the subsequences are randomly chosen; if a random value α is less than a established threshold equal to 0.5, the subsequence is cut in half, generating two parts, and their positions are swapped. Otherwise, if α is greater than the threshold, the previous process is repeated, but in the second subsequence. Finally, the position of both subsequences is inverted and H' is returned to the original form of H by a "flattening" function.

Input: H, W
$H' \leftarrow$ segment H in W parts
$a \leftarrow rand(0, |H'|)$
$b \leftarrow rand(0, |H'|)$
$\alpha \leftarrow rand(0, 1)$
If $\alpha < 0.5$ then
    $swap(H'[a])$
Si no
    $swap(H'[b])$
$Invert\_Position(H'[a], H'[b])$
$H \leftarrow flatten(H')$
Return H

Figure 6 Algorithm 2. A_M-BSWI mutation method

For the second method, M_B-IWS, the objectives of constraint smoothing on workstations are prioritized. This method works by setting two exchange points a and b, chosen randomly. According to the order of the points, a cut is made to take a section of the solution that goes from point a to point b and inverts the resulting parts. Finally, two points of said section are randomly chosen, having a 50% probability of exchanging their positions, repeating the process several times. The procedure is illustrated in Figure 7.

Input: H
$a \leftarrow rand(|H|/2)$
$b \leftarrow rand(a, |H|)$
$Invert(H[a{:}b])$
For $i = 0, i < W$
    $pos_1 \leftarrow rand(H[a{:}b])$
    $pos_2 \leftarrow rand(H[a{:}b])$
    $\alpha \leftarrow rand(0,1)$
    If $\alpha < 0.5$
        $swap(H[pos_1], H[pos_2])$
Return H

Figure 7 Algorithm 3. B_M-IWS mutation method

e. Experimental design

Five NSGAII test variants, identified as NSGA_A, NSGA_B, NSGA_C, NSGA_D, and NSGA_E, were established for the experimental stage, each one of them associated with different heuristics. The primary target was to compare the performance of NSGAII variants by observing the effect over the quality of the Pareto fronts obtained. In the variants where the proposed mutation methods, BSWI and RWI, are applied simultaneously with the same probability of execution, they are identified as the heuristic $M_A \backslash M_B$, considering them a single function when applying equation (14). Something similar occurs with the random initialization (RI) and the guided local search initialization (GLSI), as they are part of the same strategy to generate two parts of the initial population.

In addition to the proposed crossover method, the heuristic OX2 (order-based crossover) was also chosen, which, according to (Syswerda, 1991), gets satisfactory results in task scheduling problems. The variant NSGA_A has random initialization (RI), OX2 crossover operator and simple swapping mutation (SM). For the variants NSGA_B, NSGA_C and NSGA_D respectively, the AM+BM mutation proposals, the initialization proposal (GLSI+RI) and the proposed crossover operator (AHX) were taken individually. Finally, in the variant NSGA_E, the three proposals were applied as a strategic framework for optimization. Table 2 shows the heuristics applied in each one of the variants.

Method	$NSGA_A$	$NSGA_B$	$NSGA_C$	$NSGA_D$	$NSGA_E$
RI	✓	✓		✓	
GLS/R			✓		✓
OX2	✓	✓	✓		
AHX				✓	✓
SM	✓		✓	✓	
MA/MB		✓			✓

Table 2. Heuristics used in the variants of NSGAII.

f. Application of the proposals and test instances

The test instances used were taken from the *Constraint Satisfaction Problem Library web repository* (*www.csplib.org*) and they are the same used in the ROADEF 2005 challenge; these are divided in three groups: *A, B* and *X*.

Originally, each instance has a hierarchical order of objectives, which was assigned for techniques with a lexicographical approach, so it is not necessary from a Pareto approach.

According to (Nguyen & Cung, 2005), *group A* consists of 16 instances that were used for the qualifying stage of the challenge and to calibrate and refine the procedures of the proposals presented. *Group B*, consisting of 45 instances, was used in a second stage of the challenge; and *group X*, composed of 19 instances, was used to evaluate the final performance of the proposals and establish a winner. In the case of this work, a total of 15 instances were selected from all the groups: three from *group A*, eight from *group B* and four from *group X.*

Table 3 illustrates the information regarding the selected instances. In the second and third columns the number of cars for the previous day and the current day are indicated correspondingly. Next, the number of classes is indicated by $|Cl|$, then, the number of colors of the set by C and the value of the restriction B, finally the number of options of high and low priority, denoted by $|HO|$ and $|LO|$ respectively.

Instance	$\|D_p\|$	$\|D\|$	$\|Cl\|$	$\|C\|$	B	$\|HO\|$	$\|LO\|$
A							
022_3_4	15	484	18	12	145	3	6
048_39_1	18	600	158	12	10	5	12
064_38_2	30	874	25	14	15	7	2
B							
022_S22_J1	14	426	20	15	500	2	7
028_ch1_S22	20	265	91	20	15	4	3
029_S21_J6	42	731	12	12	15	4	3
035_ch2_S22	23	270	8	9	150	3	2
039_ch1_S22	31	1232	86	12	15	2	9
048_ch1_S22	310	592	181	14	10	6	19
048_ch2_S22	48	547	112	12	10	7	16
064_ch1_S22	28	826	46	14	15	11	3
X							
023_S49_J2	18	1261	79	13	40	5	7
024_S49_J2	18	1320	106	15	10	7	11
025_S49_J1	74	997	208	20	60	6	14
034_VP_S51	20	232	7	30	400	6	2

Table 3 Selected test instances and their information

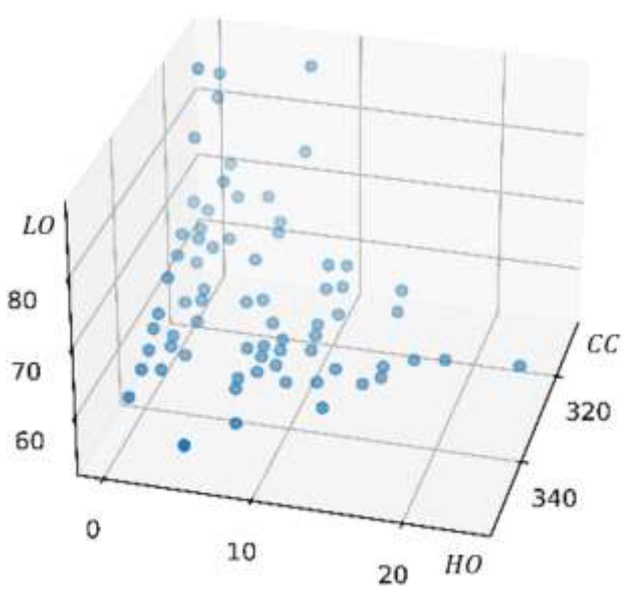
The test parameters of the experiments are illustrated in Table 4. The values for crossover and mutation rates use similar values used by works based on evolutionary computation, such as (Zinflou, Gagné, & Gravel, 2013) and (Marquez-Sanchez & Puga-Soberanes, 2019).

Parameter	Value
Populaton size	100
Archive size	100
Generations	1000
Crossover rate	0.9
Mutaton rate	0.35

Table 4 Parameters used by variants.

Archive size is used as a maximum number of non-dominated solutions for the last Pareto front found and uses the same value as *Population size* parameter. Both parameters used high values since we sought to guide a search that would yield the largest number of final solutions to compare results between the test variants of NSGAII with the different instances. The number of generations is high because we observed that in some instances (e.g 039_ch1_S22 and 023_S49_J2) the convergence with the different variants was slow.

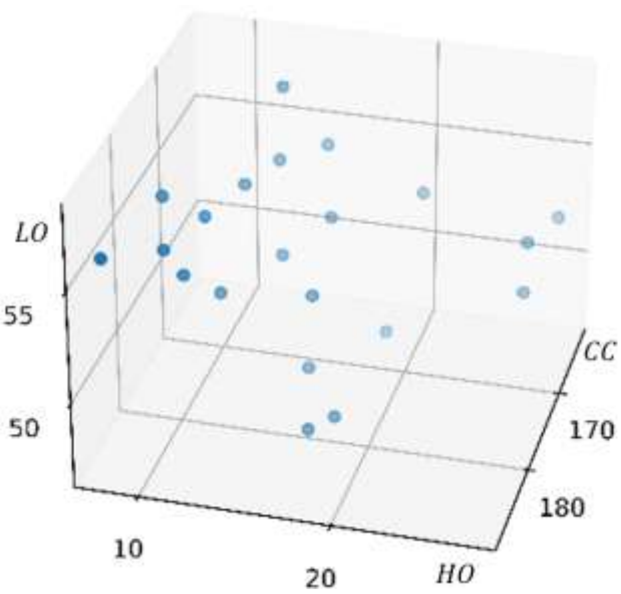
At the end of an execution of any variant, the Pareto front of non- dominated solutions was identified as a sequence of cars with a corresponding associated list of plottable points. Graphic 1 shows the graph of the set of solutions that form the front generated by the three objective functions in the *NSGA_A* variant, using the control instance 022_3_4 of *group A*. *Graphic 2* shows the graph of the set of Pareto solutions for the *NSGA_E* variant, with the same instance.



Graphic 1 Final set of Pareto solutions returned in a typical execution of the NSGA_A variant.

ISSN 2414-4924

June 2022 Vol.8 No.14 1-13



Graphic 2 Final set of Pareto solutions returned in a typical execution of the NSGA_E variant.

Results

A summary of the mean and standard deviation of the hypervolume metric from the results of 36 experiments for each group of Instances corresponding to each NSGAII, variant is shown in Tables 5-7. This metric represents a quality or performance value to the final Pareto fronts obtained in each experiment. Those instances from each group whose results are representative are highlighted in bold.

Instancia	$NSGA_A$	$NSGA_B$	$NSGA_C$	$NSGA_D$	$NSGA_E$
022_3_4	(2.59E+6 5.20E+5)	(1.45E+6 3.21E+5)	(2.13E+62.38 E+5)	2.46E+6; 2.29E+5)	(5.45E+61.53E +5)
048_39_1	(8.79E+6 1.86E+6)	(8.94E+6 1.13E+6)	(2.99E+72.50 E+6)	(3.39E+7 2.71E+6)	(3.60E+71.06E +6)
064_38_2	(1.19E+72.34E +6)	(5.09E+6 2.87E+5)	(1.04E+73.81 E+5)	(1.10E+7 4.40E+5)	(3.02E+71.13E +6)

Table 5. Summary of hypervolume metrics for selected instances in group A.

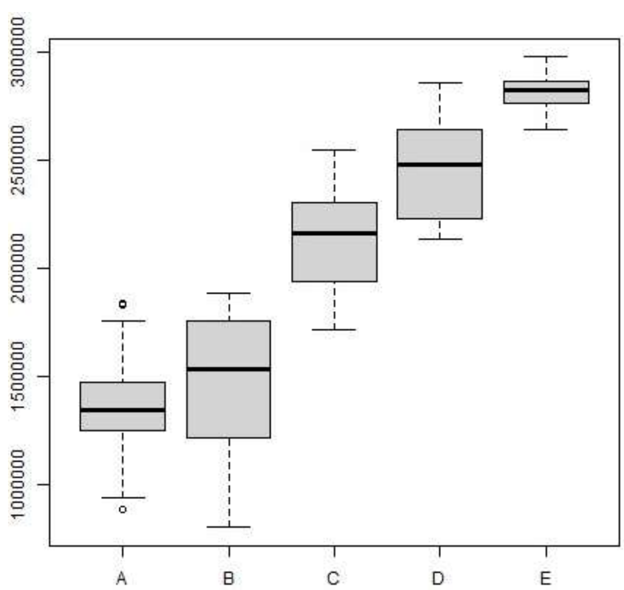
Instancia	$NSGA_A$	$NSGA_B$	$NSGA_C$	$NSGA_D$	$NSGA_E$
022_S22_J1	(1.39E+5; 1.06E+4)	(1.41E+5; 6.16E+3)	(2.17E+6; 1.22E+5)	(1.94E+6; 8.29E+4)	(2.90E+6; 8.14E+4)
028_ch1_S22	(2.90e+06; 5.95e+05)	(3.29e+06; 3.94e+05)	(3.65e+06; 3.93e+05)	(3.52e+06; 2.03e+05)	(4.55e+06; 2.82e+05)
029_S21_J6	(3.54e+06; 6.95e+05)	(3.78e+06; 6.95e+05)	(6.18e+06; 7.28e+05)	(6.54e+06; 9.38e+05)	(7.47e+06; 7.71e+05)
035_ch2_S22	(4.21e+06; 2.09e+05)	(4.31e+06; 2.11e+05)	(5.16e+06; 1.15e+05)	(4.87e+06; 4.24e+04)	(5.34e+06; 2.54e+05)
039_ch1_S22	(5.53e+06; 1.62e+06)	(5.58e+06; 2.04e+05)	(1.69e+07; 1.16e+06)	(1.64e+07; 5.99e+05)	(2.43e+07; 3.70e+06)
048_ch1_S22	(7.23e+06; 1.96e+06)	(7.33e+06; 9.49e+05)	(2.65e+07; 1.30e+06)	(2.80e+07; 4.93e+05)	(3.33e+07; 7.00e+06)
048_ch2_S22	(2.89e+07; 5.19e+06)	(2.45e+07; 2.37e+06)	(4.54e+07; 4.39e+06)	(4.72e+07; 1.74e+06)	(5.42e+07; 2.16e+06)
064_ch1_S22	(1.40e+07; 2.84e+06)	(1.29e+07; 2.03e+06)	(4.12e+07; 4.62e+06)	(4.58e+07; 3.13e+06)	(5.20e+07; 3.03e+06)

Table 6 Summary of hypervolume metrics for selected instances in group B

Instancia	SGA_A	$NSGA_B$	$NSGA_C$	$NSGA_D$	$NSGA_E$
023_S49_J2	(1.56e+074.02e+06)	(1.56e+07; 6.52e+05)	(7.39e+07; 1.40e+06)	(6.28e+07; 3.31e+06)	(7.67e+07; 2.56e+06)
024_S49_J2	(3.89E+071.19E+07)	(3.84E+07; 7.12E+06)	(1.43E+08; 1.79E+06)	(1.63E+08; 3.57E+06)	(1,96E+08; 5.19E+06)
025_S49_J1	(5.21e+079.78e+06)	(5.87e+07; 1.33e+07)	(7.35e+07; 8.55e+06)	(7.72e+07; 8.19e+06)	(9.93e+07; 5.24e+06)
034_VP_S51	(4.84e+077.39e+06)	(4.80e+07; 1.03e+07)	(1.28e+08; 1.63e+07)	(1.37e+08; 1.25e+07)	(1.58e+08; 8.55e+06)

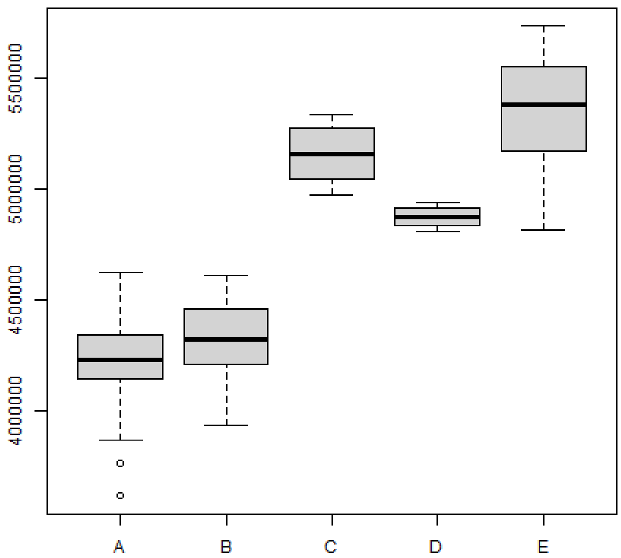
Table 7 Summary of hypervolume metrics for selected instances in group X.

The results show that NSGA_A and NSGA_B have means close to each other with respect to the other configurations, and that NSGA_C and NSGA_D have higher means than the other variants. This indicates an improvement as the effect of the initialization and recombination proposals, respectively. In the case of the NSGA_E, which integrates the three proposals (initialization, crossover and mutation), in general, higher mean values are observed, with respect to the NSGA_C and NSGA_D, showing an improvement in the generation of Pareto fronts from the point of view of the Hypervolume metric. Graphics 3-5 give a visual perspective, showing the box plots of the representative instances 022_3_4, 35_ch2_S22 and 34_VP_S51 marked in bold in Tables 5-7.

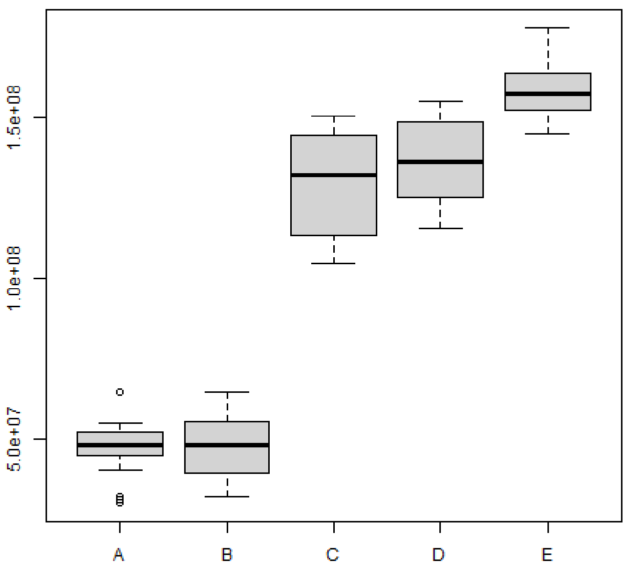


Graphic 3 Variants box plots, corresponding to instance 022_3_4.

ISSN 2414-4924



Graphic 4. Variant box plots, corresponding to instance 35_ch2_S22



Graphic 5. Variant box plots, corresponding to instance 34_VP_S51

In some instances, such as 35_ch2_S22 and 022_3_4, it is observed that the dispersion of the hypervolume values of variants NSGA_B, NSGA_C and NSGA_D is greater than the other configurations, this may be due to the fact that, individually, the proposals overexploited the search in different regions of the solution space, yielding disparate results in each experiment.

For a more rigorous analysis of the performance of each variant and its results with each of the instances, analysis of variance (ANOVA) and pairwise comparison were applied, with a significance level of 0.05, whose results are shown in Tables 8, 9.

Pairwise test yields *P* metric values between the groups {NSGA_A,NSGA_B} and {NSGA_C ,NSGA_D} that indicates no statistically significant diference, being insufficient evidence of the difference and superiority of a variant over the other. However, between the groups {NSGA_A, NSGA_C, NSGA_D} and NSGA_B, NSGA_C, NSGA_D} a significant difference is shown, suggesting that at least the group {NSGA_C, NSGA_D} is superior to the group {NSGA_A, NSGA_B}. On the other hand, the difference between the group {NSGA_A, NSGA_B, NSGA_C, NSGA_D} and the NSGA_E variant shows *P* values below the level of significance, which suggests the significant difference and superiority of NSGA_E.

In the ANOVA test, the sample population is the total of the experiments executed with all the variants. Then, the population was divided by each test variant. All the hypervolume values of each experiment for each variant were taken and their means were calculated and compared, showing that they are statistically different. In other words, a significant difference is observed between NSGA_A and NSGA_E, and also between the groups {NSGA_B, NSGA_C} and {NSGA_B, NSGA_D}, for all instances. In this context, two hypotheses are proposed: a null hypothesis that suggests no difference between the groups of configurations and means; and an alternative hypothesis suggesting significative difference between variants and means. The results of the *F* metric (less than 0.05) support the alternative hypothesis.

	Df	Sum Sq	Mean Sq	value	Pr(>F)
Algoritmo	4	2,86E+22	7,14E+21	560.7	< 2.2e-16
Instancia	373	2,47E+23	6,61E+20	51.9	< 2.2e-16
Residuals	4347	5,54E+22	1,27E+19	-	-
-	-	-	-	-	-

Table 8 ANOVA test results

Alg.	A	B	C	D
B	1	-	-	-
C	< 2e-16	< 2e-16	-	-
D	< 2e-16	< 2e-16	1	-
E	< 2e-16	< 2e-16	1.3e-13	1.9e-11

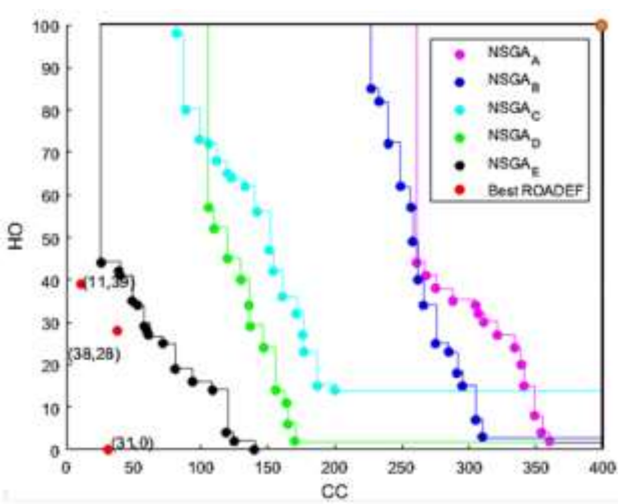
Tabla 9 Pairwise Comparison test results

ISSN 2414-4924

Graphic 6 shows the graph of the final Pareto fronts generated by the different test variants, taking into account the objectives of color changes (CC) and overloads in high priority options (HO) and the composition of their hypervolumes, only in the case of instance 022_3_4. The systematic improvement of the proposals is more distinguishable if we observed the size of the "areas" formed from a maximum reference point (orange point) towards all the points of each front. The larger the “area”, the greater the hypervolume metric, and greater the performance of the variant.

Despite the fact that it was sought to intensify the search in an adequate and balanced way, the solutions which obtained higher ranges in HO objective (approx. [0.100]) suggest that, the mechanism used could be improved by defining more restrictions in the proposals that would give a better focus to the search, however this could have negatively affected the complexity of the algorithm and its computational performance.

On the other hand, one of the observed advantages of the Pareto approach, although the difference in the extreme solutions of the fronts of said variants is wide, the greater variety of solutions offered by the decision maker of the algorithm provides several alternatives. close to, or even better (depending on the objective to be optimized), to the final solutions of the best qualified techniques in the ROADEF 2005 and other techniques with a lexicographical approach in the state of the art (Nguyen & Cung, 2005), marked with red dots.



Graphic 6. Layout of solutions of each NSGA II variant and the best proposals of ROADEF 2005, tackling instance 022_3_4.

The opposite nature of the objectives of the problem was also observed, clearly influencing the results obtained and hindering the generation of optimal Pareto fronts. Initially, the smoothing objectives (FH, FL) are seen to benefit more in solutions composed of very small groups of cars of the same color. The opposite occurs in the objective of color changes, with large groups of cars of the same color being more beneficial, as long as constraint B is not violated.

Likewise, those N/P ratio constraints where the numerator N is close to the denominator P, generate a smaller number of violations in the workstations. These two facts were taken into account for the formulation of the proposals.

Conclusions

This work adress the car sequencing problem from a multi-objective Pareto sense. As far as is known, in the state of the art there are just a few works that tackles this problem in this way, most of the researchers use multiobjective lexicographical approximations.

Initialization, crossover and mutation strategies were presented specifically formulated for the CSP, that is distinguished by the definition of its constraints and the contradictory nature between smoothing and color constraints. The framework, constituted by the proposals, was applied over NSGAII, allowing us to establish test variants that made possible the measurement of the impact on the general improvement and the quality of the optimal solutions obtained, both for each strategy individually and as a whole. The experiments used instances of the ROADEF 2005 and the hypervolume metric to measure the quality of the results.

The statistical tests show that there is a significant difference between the algorithm used without any proposals and the same one applying the proposals; the results show that there is a systematic improvement, being the NSGA_E variant, which includes all the proposals, the one that generally presents the better performance. Our research gives viability and leaves open the process of continuity and improvement of Pareto fronts with the proposal.

This work contributes to the research process to find optimal Pareto fronts to the multi-objective approach of the CSP. Further work is being carried out tackling the CSP according with a feasibility study of a multi-objective Pareto approach in a decomposition scheme. The study for the improvement of Pareto optimal fronts in the car sequencing problem has repercussions both in the optimization of automobile production in a timely manner and in the optimization of human and material resources.

Thanks

This work has been fundend by the National Comitte of Science and Technology CONACYT of Mexico, grant number 1076451 (E. Guzmán). The authors thank the CONACYT, the National Technological Institute of Mexico and the Technological Institute of León for the support provided for the conduction of this research.

References

Bolat, A., & Yano, C. A. (1992). Scheduling algorithms to minimize utility work at a single station on a paced assembly line. *Production Planning & Control*, 393-405. doi:https://doi.org/10.1080/09537289208919409

Chutima, P., & Olarnviwatchai, S. (2018). A multi-objective car sequencing problem on two-sided assembly lines. *Journal of Intelligent Manufacturing*(29), 1617 – 1636. doi:https://www.doi.org/10.1007/s10845-016-1201-6

Deb, K. (2002). A Fast and Elitist Multiobjective Genetic Algorithm: NSGAII. *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION*, 182 - 197. doi:https://www.doi.org/10.1109/4235.996017

Guerre-Chaley, F., Frein, Y., & Bouffard-Varcelli, R. (1995). An Efficient Procedure for Solving a Car Sequencing Problem. *Procedings of INRIA/IEEE Symposium on Emerging Technologies and Factory Automation ETFA'95 IEEE*, *2*, 385-393. doi:https://www.doi.org/10.1109/ETFA.1995.496679

Marquez-Sanchez, E. M., & Puga-Soberanes, H. J. (2019). Algoritmo de evolución diferencial con reparador cromosómico aplicado a un problema de secuenciación de vehículos. *Research in Computer Science*, 912-927. doi:https://doi.org/10.13053/rcs-148-8-21

Nguyen, A., & Cung, V. (2005). Le problème du Car Sequencing Renault et le challenge ROADEF' 2005,. *Proceedings of Journées Francophones de Programmation par*, 3-10.

Parello, D. B., & Kabat, W. C. (1986). Job-shop scheduling using automated reasoning: A case study of the car-sequencing problem. *Journal of Automated Reasoning*, 1- 42. doi:https://doi.org/10.1007/BF00246021

Syswerda, G. (1991). Schedule optimization using genetic algorithms. *Handbook of Genetic Algorithms*, 332-349.

Velazquez, J. C., & Puga-Soberanes, H. J. (2020). Estudio de factibilidad para la aplicación de algoritmos de estimación de distribución al problema de secuenciación de vehículos. *Revista de Ingeniería Industrial*, 31-42.

Warmick, T., & Tsang, E. (1995). Tackling Car Sequensing Problems Using a Generic Genetic Algorithm. *Evolutionary Computation*, 267-298. doi:https://doi.org/10.1162/evco.1995.3.3.267

Zinflou, A. (2008). *Design of an Efficient Genetic Algorithm to Solve the Industrial Car Sequencing Problem.* Vienna, Austria: Intech Education and Plublishing. doi:https://doi.org/10.5772/6122

Zinflou, A., Gagné, C., & Gravel, M. (2013). *Pareto Solution Strategies for the Insdustrial Car Sequencing Problem.* doi: https://doi.org/10.1002/9781118731598.ch10

Zitzler, E., Laumanns, M., & Thiele, L. (2001). *SPEA2: Improving the strength Pareto evolutionary algorithm.* TIK-report.

Zufferey, N. (2016). Tabu Search Approaches for Two Car Sequencing Problems with Smoothing Constraints. *Metaheuristics for Production Systems*, 167 - 190. doi:https://doi.org/10.1007/978-3-319-23350-5_8