UNIVERSIDAD NACIONAL DE COLOMBIA
SEDE MEDELLÍN
FACULTAD DE MINAS

DYNA
http://dyna.medellin.unal.edu.co/

A hybrid genetic algorithm for ROADEF'05-like complex production problems

Mariano Frutos [a], Ana Carolina Olivera [b] & Fernando Tohmé [c]

[a] *Department of Engineering, Universidad Nacional del Sur and CONICET, Bahía Blanca, Argentina. mfrutos@uns.edu.ar*
[b] *Department of Exact and Natural Sciences, Universidad Nacional de la Patagonia Austral and CONICET, Caleta Olivia, Argentina. aco@cs.uns.edu.ar*
[c] *Department of Economics, Universidad Nacional del Sur and CONICET, Bahía Blanca, Argentina. ftohme@criba.edu.ar*

Received: April 25th, de 2014. Received in revised form: October 30th, 2014. Accepted: November 13th, 2014

Abstract
In this work, we present a hybrid technique that combines a Genetic Algorithm with meta-heuristics to solve a problem in RENAULT France's production plants. The method starts with an initial solution obtained by means of a GRASP (Greedy Randomized Adaptive Search Procedure) used as an input for a Genetic Algorithm complemented by a Simulated Annealing procedure of population improvement. We establish a comparison point among the different techniques used in the method. Their performances are evaluated as well as that of the entire method. The conclusion is that hybrid methods have clear advantages for the treatment of production planning problems.

Keywords: multi-objective optimization, hybrid algorithms, car sequencing.

Algoritmo genético híbrido para problemas complejos de producción tipo ROADEF'05

Resumen
En este trabajo se presenta una técnica híbrida que combina un Algoritmo Genético con meta-heurísticas para la resolución de un problema en las plantas productivas de RENAULT Francia. El método comienza con una solución inicial por medio de GRASP (Greedy Randomized Adaptive Search Procedure), que es utilizada como entrada por un Algoritmo Genético complementado por un procedimiento de Simulated Annealing para mejorar las poblaciones. Se establece un punto de comparación entre las diferentes técnicas. El desempeño de las mismas es evaluado así como el de todo el método. La conclusión es que los métodos híbridos tienen claras ventajas para el tratamiento de problemas de planificación de la producción.

Palabras clave: optimización multi-objetivo, algoritmos híbridos, secuenciamiento de vehículos.

1 Introduction

Scheduling and programming problems at the core of the SFROAD (Société Française de Recherche Opérationnelle et d'Aide à la Décision) 2005 focus on the problems that have arisen at the RENAULT, France production plants. They present ROADEF'05 with the challenge of finding a solution to a real world extension of the classical car sequencing problem [1], the goal of which is to schedule cars along an assembly line while satisfying several capacity constraints. The particular problem addressed by SFROAD differs from the standard one since, besides capacity constraints imposed by the assembly shop, it introduces paint batching constraints involving the minimization of the consumption of solvents in the paint shop. Our analysis focuses on the phase of arrangement of daily sequences in production problems [2]. Here it is necessary to take into account different, even conflicting factors. The main goal is to develop a hybrid technique to tackle this problem and evaluate its performance compared to other traditionally used methods. Usual techniques intended to find approximate optimal solutions to similar problems are, greedy search [3], GRASP [4], GISMOO Algorithm [5], local search [6,7], hybrid variable neighborhood search [8], among others. The Hybrid Genetic Algorithm (HGA) presented in this work amalgamates constructive procedures like the Greedy Randomized

DOI: http://dx.doi.org/10.15446/dyna.v82n190.43137

Adaptive Search Procedure (GRASP) [9], and Genetic Algorithms (GAs) [10] with search methods like Simulated Annealing (SA) [11].

The paper is structured as follows: First, the ROADEF’05 problem is introduced. Then, the proposed Hybrid Genetic Algorithm (HGA) is described in detail. Then, comparisons between the HGA and other methods are presented. Finally, we analyze the results of running the HGA and present the conclusions.

2. ROADEF'05

The problem consists in determining the scheduling order of vehicles in a production day that best satisfies the assembly line and paint shop requirements [1]. The paint shop goal is to minimize the consumption of paint solvent. Therefore, it requires grouping vehicles according to their colors as well as minimizing the number of spray gun washes, i.e. to schedule the longest paint color batches that are possible. Paint color batches have a limitation on the upper batch size due to the need for frequent washing of the spray guns even when there is no need for paint color changes. This limitation constitutes a hard constraint.

In order to lighten the workload in the assembly line, vehicles that require special assembling operations have to be evenly distributed throughout the total processed cars. These vehicles are considered to be “hard to assemble”. There are two classes of ratio constraints, high priority level and low priority level ones. High priority level ratio constraints ensue from car characteristics that require heavy workloads in the assembly line. Low priority level ratio constraints, instead, result from car features that cause small inconveniences in the production process.

Given the heterogeneities involved in the problem, multi-objective optimization constitutes a natural approach to the problem [12-14]. The objectives are, from the highest to the lowest priority level with no compensation between them: (a) the minimization of paint color changes (eq. 1); (b) minimization of the number of violations of high priority level ratio constraints (eq. 2); (c) the minimization of violations of low priority level ratio constraints (eq. 3).

$$f(a) = \sum_{i=1}^{N-1} (NPCC)_i,\ \forall i = 1,\dots,(N\text{-}1) \tag{1}$$

$$f(b) = \sum_{i=1}^{N-n+1} (NVHPRC)_i,\ \forall i = 1,\dots,(N\text{-}n+1) \tag{2}$$

$$f(c) = \sum_{i=1}^{N-n+1} (NVLPRC)_i,\ \forall i = 1,\dots,(N\text{-}n+1) \tag{3}$$

Table 1.
Specifications summary.

Scenario	Order	HPRC	LPRC	Batches	N
Low level of difficulty					
022_3_4_EP_RAF_ENP	b, a, c	3	6	450	499
024_38_3_EP_RAF_ENP	b, a, c	5	8	10	1274
025_38_1_EP_ENP_RAF	b, c, a	4	18	10	1232
025_38_1_EP_RAF_ENP	b, a, c	4	18	10	1232
039_38_4_EP_RAF_ch1	b, a	5	-	20	981
039_38_4_RAF_EP_ch1	a, b	5	-	15	981
048_39_1_EP_ENP_RAF	b, c, a	5	12	10	618
048_39_1_EP_RAF_ENP	b, a, c	5	12	10	618
Medium level of difficulty					
Scenario	**Order**	**HPRC**	**LPRC**	**Batches**	**N**
022_EP_ENP_RAF_S22_J1	b, c, a	2	7	500	540
022_EP_RAF_ENP_S22_J1	b, a, c	2	7	500	540
024_V2_EP_RAF_ENP_S22_J1	b, a, c	6	7	10	1319
024_V2_RAF_EP_ENP_S22_J1	a, b, c	6	7	10	1319
029_EP_RAF_ENP_S21_J6	b, a, c	4	3	15	773
029_RAF_EP_ENP_S21_J6	a, b, c	4	3	15	773
048_ch1_EP_RAF_ENP_S22_J3	b, a, c	6	19	10	902
048_ch1_RAF_EP_ENP_S22_J3	a, b, c	6	19	10	902
High level of difficulty					
Scenario	**Order**	**HPRC**	**LPRC**	**Batches**	**N**
022_RAF_EP_ENP_S49_J2	a, b, c	3	9	200	718
023_EP_RAF_ENP_S49_J2	b, a, c	5	7	40	1279
024_EP_RAF_ENP_S49_J2	b, a, c	7	11	10	1338
025_EP_ENP_RAF_S49_J1	b, c, a	6	14	60	1071
029_EP_RAF_ENP_S49_J5	b, a, c	4	3	60	822
035_CH2_RAF_EP_S50_J4	a, b	2	-	1000	377
039_CH1_EP_RAF_ENP_S49_J1	b, a, c	1	11	20	1543
039_CH3_EP_RAF_ENP_S49_J1	b, a, c	2	10	20	1283

Source: Compiled by author.

Here NPCCi is the number of paint color changes in the sequence i, NVHPRCi and NVLPRCi are the number of violations of high priority ratio constrains and low priority ratio constrains, respectively in sequence i. On the other hand, N is the number of sequences and n the number of sub-sequences. The fitness (f) is defined for each of the three objectives. In what follows, we assume that the values of α, β and δ are given (α=1.000.000, β=1.000 y δ=1) and that the full model is captured by eq. (4) and eq. (5). In eq. (4), objective one (*obj one*) corresponds with (a) or (b), objective two (*obj two*) is (a), (b) or (c), and objective three (*obj three*) is (a) or (c). Of course, the "or" in the definition of the objectives are exclusive, i.e. only one of the alternatives will be the case. In this way eq. (4) is directly related to the possible scenarios that frame the problem. Eq. 5 restricts the number of vehicles of the same color in each sub-sequence.

$$\mathrm{Min}: \alpha * f(\text{obj one}) + \beta * f(\text{obj two}) + \delta * f(\text{obj three}) \qquad (4)$$

s.t.

$$(\mathrm{LEPC})_j \leq (\mathrm{LEPC})_{max}, \; \forall\, j = 1,\ldots, S \qquad (5)$$

Here $LEPC_j$ is the amount of cars of the same color in the sub-sequence j, $LEPC_{max}$ is the maximum allowable number of cars of the same color and S is the number of sub-sequences of equal color.

Table 1 specifies the level of complexity, scenario, number of high priority level constraints (HPRC), number of low priority level constraints (LPRC), limit of paint color batches (batches) and number of vehicles in a production day (N). The priorities of the objectives (a), (b) and (c) are shown in the Order column.

3. Hybrid Genetic Algorithm

The Hybrid Genetic Algorithm (HGA) presented here has two main stages, each with a different clear objective: the first one constructs an initial solution set with GRASP [9,15], while the second one, using it, follows the evolution of the population by means of the Genetic Algorithm (GA) [10,16] combined with Simulated Annealing (SA) [11]. In the GA stage, the SA module is introduced to improve the children from one generation to the other. Fig. 1 shows the layout of the Hybrid Genetic Algorithm.

3.1. Greedy Randomized Adaptive Search Procedures

The GRASP algorithm is, in turn, structured in two phases: a constructive one whose product is a good but not necessarily locally optimal solution; and a local search procedure that examines solution neighborhoods until a local optimum is found. The procedure begins by taking a random vehicle as the first element in the sequence. While not all vehicles are in the sequence, the closest match for place i is chosen. Each element of the candidate list is assigned a probability to be chosen. These probabilities are weighed with respect to a partial fitness value.

```
Algorithm Hybrid Genetic Algorithm
Input: TimeLimit: Real;
       N: interner;
OutPut:
Var:   Time: Real;
       i,j: interger;
Time :- Now;
GRASP(Population(0));
for i = 1 to N do
    Pop (0).Ind (i).Fitness();
end for
i :- 1;
while Time < TimeLimit do
        Parents :- Pop.Select_Parents();
        Children :- Crossover(Parents);
        MutationChildren :- Mutation(Children);
        SimulatedAnnealing(MutationChildren);
        for j = 1 to Number_of_Child(MutationChildren) do
            MutationChildren(j).Fitness();
        end for
        Population(i) :- NewPopulation(MutationChildren,Pop (i-1));
        i :- i+1;
end while
end Algorithm
```

Figure 1. Hybrid Genetic Algorithm.
Source: Compiled by author.

```
Algorithm GRASP
Input: cars:listofcars;
       n,p: integer;
OutPut: listofindividuals;
Var: S, Sj, Sbest, listcandidates: listofcars;
     i,j,k : integer;
V0 :- aleatorycar(cars);
for k = 1 to n do
carsaux:-cars;
    while S.cars < cars.count do
        listcandidates.add(bestcarcandidate(carsaux));
        carsaux.del(bestcarcandidate(carsaux));
        Vi :- choose_prob(listofcandidates);
        S.addcar(Vi);
    end while
    for j = 0 to m do
        Sj :- Permute(p,S);
        if Sj.fitness < S.fitness then
            Sbest :- Sj;
        end if
        S :- Sj;
    end for
listofindividuals.add(S);
end for
end Algorithm
```

Figure 2. GRASP Stage pseudo-code.
Source: Compiled by author.

When the sequence is complete, an n permutation of vehicle-pairs is repeated m times. The objective is to look for a local optimum in the neighborhood of the solution. The values of m and n are input parameters of the algorithm. In Fig. 2 the layout of the GRASP algorithm is observed. Tables 2 (Solution: 17 5 4 10 9 8 2 6 3) and 3 (Solution: 1 7 5 4 10 9 8 3 2 6) show the construction of a feasible and a non feasible solution with the GRASP.

3.2. Genetic Algorithm Stage

The genetic stage works on the basis of the individuals produced by the GRASP stage. Each individual is a list of

Table 2.
Feasible solution (GRASP).

Greedy Algorithm - Feasible Building									
→1	2	3	3	2	→9	3	→2	3	→3
	4	→5	→4	6		→8	3	→6	
	→7	8	6	8			6		
	8	10		→10					
	10								
1	7	5	4	10	9	8	2	6	3
Solution									

Source: Compiled by author.

Table 3.
Non-feasible solution (GRASP).

Greedy Algorithm - No Feasible Building									
→1	2	3	3	2	→9	3	→2	3	3
	4	→5	→4	6		→8	3	→6	↓
	→7	8	6	8			6		
	8	10		→10					
	10								
							↓		
1	7	5	4	10	9	8	3	2	6
Solution									

Source: Compiled by author.

integers that represents the order in sequence of production. An individual in the population is a chain of integers. Each integer represents a vehicle.

The chromosome of the individual indicates the order of production sequence, in one day of work, from left to right. The initial population is a set of solutions received from the GRASP stage. Ranking selection is used to choose the parents that will construct the next population. An empirical analysis allows us to conclude that a population of between 90 and 100 individuals constitute a large enough sample.

3.2.1. *Crossover*

The GA is implemented with one point crossover. The operator randomly chooses a point to cross the parents. The simple crossover is applied here. Table 4 shows the crossover between $Parent_1$ = (1 2 3 4 5 6 7 8 9 10) and $Parent_2$ = (8 1 4 7 10 3 9 2 6 5) to obtain $Child_1$ = (1 2 3 4 10 9 6 5 8 7) and $Child_2$ = (8 1 4 7 5 6 9 10 2 3).

As an example, let us start from parents $Parent_1$ = (1 2 3 4 / 5 6 7 8 9 10) and $Parent_2$ = (8 1 4 7 / 10 3 9 2 6 5), where the slashes are split points. First, we obtain $Child_1$ = (1 2 3 4 * * * * * *) and $Child_2$ = (8 1 4 7 * * * * * *), preserving the first sub sequence of the respective split point for $Parent_1$ and $Parent_2$. Then, starting from the split point, the vehicles in $Parent_2$ that are not in $Child_1$ are used to complete the vehicles in $Child_1$. In this case, the list of vehicles in $Parent_2$ starting from the second split point is: (10 3 9 2 6 5), but when the vehicles that belong already to $Child_1$ (i. e., 3 2) are eliminated, the sub-sequence becomes (10 9 6 5 8 7). These vehicles are added to $Child_1$, starting from the split point. When the end is reached, the remaining cars are added at the initial part of $Child_1$.

Table 4.
Crossover Operation.

$Parent_1$									
1	2	3	4	5	6	7	8	9	10
$Parent_2$									
8	1	4	7	10	3	9	2	6	5
$Parent_1$ - $Child_1$									
1	2	3	4	*	*	*	*	*	*
$Parent_2$ - $Child_1$									
8	*	*	7←	→10	*	9	*	6	5
$Child_1$									
1	2	3	4	10	9	6	5	8	7
$Parent_1$ - $Child_2$									
*	2	3←	*	→5	6	*	*	9	10
$Parent_2$ - $Child_2$									
8	1	4	7	*	*	*	*	*	*
$Child_2$									
8	1	4	7	5	6	9	10	2	3

Source: Compiled by author.

Table 5.
Mutation Operation.

$Child_1$									
1	2	→3←	4	10	9	6	→5←	8	7
Mutation									
1	2	→5←	4	10	9	6	→3←	8	7

Source: Compiled by author.

3.2.2. Mutation

The mutation operator generates one value for each vehicle in an individual solution. This value indicates whether the vehicle must change its position in the sequence. If this is the case, a new value is generated. The new number indicates the new position of the vehicle in the sequence. Table 5 ($Child_1$: 1 2 3 4 10 9 6 5 8 7, Mutation: 1 2 5 4 10 9 6 3 8 7) shows an example of a child mutation.

3.2.3. Simulated Annealing

The goal of a "simulated annealing phase" is to improve the quality of children between generations. The key factor consists in defining an initial parameter λ, the initial temperature (T_i), a cooling speed (ω), the number of iterations (M) for each temperature (T), and the final temperature (T_f). For all children ch in a generation we select ch for the initialization of S_a and given T_i, greater than T_f , Simulated Annealing runs through two nested cycles. The first cycle is associated with T. The second is related to M, which varies depending on the actual state of T and parameter ω. The second cycle generates the new sequence S_c. S_c is constructed taking into account the pair permutations of vehicles in S_a. If $f(S_c) < f(S_a)$, S_c replaces S_a. Otherwise, it is associated a probability of accepting to S_c. The objective is to escape from a local optimum. When the nested cycle terminates, T is actualized considering α and initial parameter λ. The algorithm returns S_a, the last sequence of vehicles found. In Fig. 3 we show the layout of SA.

Table 6.
Comparison between Optimal Solutions.

Scenarios	BKS	HGA	GA	SA	GRASP
Low level of difficulty					
	Fitness values				
022_3_4_EP_RAF_ENP	31001	31001	71003	100022	110189
024_38_3_EP_RAF_ENP	4249083	4279287	32183423	64321229	62089430
025_38_1_EP_ENP_RAF	99720	99720	570053	1875775	1998245
025_38_1_EP_RAF_ENP	231134	232472	421262	452889	1054716
039_38_4_EP_RAF_ch1	13129000	14141000	35217000	101112000	115505000
039_38_4_RAF_EP_ch1	68155000	68155000	68272000	68742000	70128000
048_39_1_EP_ENP_RAF	61290	61290	19980425	26452112	29504120
048_39_1_EP_RAF_ENP	174612	175615	18205759	27225132	31509447
Medium level of difficulty					
	Fitness values				
Scenarios	**BKS**	**HGA**	**GA**	**SA**	**GRASP**
022_EP_ENP_RAF_S22_J1	3109	3109	71003	15466	17470
022_EP_RAF_ENP_S22_J1	19144	23135	32183423	57138	58172
024_V2_EP_RAF_ENP_S22_J1	1074299068	1074299068	570053	1212448564	1325887451
024_V2_RAF_EP_ENP_S22_J1	134023158	134072444	421262	135253163	145007023
029_EP_RAF_ENP_S21_J6	35167170	35173150	35217000	35307150	42107995
029_RAF_EP_ENP_S21_J6	52711171	52711171	68272000	53877633	53906700
048_ch1_EP_RAF_ENP_S22_J3	161378	161378	19980425	252407	252407
048_ch1_RAF_EP_ENP_S22_J3	64115670	64115670	18205759	65460009	66455118
High level of difficulty					
	Fitness values				
Scenarios	**BKS**	**HGA**	**GA**	**SA**	**GRASP**
022_RAF_EP_ENP_S49_J2	12002003	12002003	12002003	13002670	13172128
023_EP_RAF_ENP_S49_J2	192466	203077	322115	361285	392312
024_EP_RAF_ENP_S49_J2	337006	337006	948330	70558508	73828623
025_EP_ENP_RAF_S49_J1	160407	160407	232634	380345	380345
029_EP_RAF_ENP_S49_J5	110298	110298	170228	172710	175816
035_CH2_RAF_EP_S50_J4	6056000	6056000	8940000	9056000	10458000
039_CH1_EP_RAF_ENP_S49_J1	69239	69239	190046765	161028	4778046
039_CH3_EP_RAF_ENP_S49_J1	231030	233611	499617602	315135	5367105

Source: Compiled by author.

4. Experiments

Preliminary essays lead to the adoption of the following parameters: Size of the Population: 250, Number of Generations: 500, Probability of Crossing: 0.80, Probability of Mutation: 0.01, Initial Temperature for SA: 850, Final Temperature for SA: 0.01, Cooling Factor for SA: 0.95, CPU: 3.00 GHZ, RAM: 4.00 GB. Each algorithm had 30 runs. Table 6 shows the best known solution for each problem (BKS) and the best results reached with each meta-heuristic [9-11].

In addition, the third column in Table 6 presents the best results per HGA. Table 7 shows the proportion of the 30 runs in which the best result was reached (Success (%)). Running times were always short of 300 seconds. Taking the average running time for HGA, GA took 21.2% less time than that, SA 38.5% less, while GRASP ran for 62.8% less time.

For the problems pertaining to low levels of difficulty, HGA reaches the best result (on average) 95.4% of runs, while GA, SA and GRASP reach the best results at 89,9%, 61,6% and 56,2%, of the runs, respectively. On problems pertaining to medium levels of difficulty, HGA achieves the best results at an average of 89.9% of its runs, while GA,

```
Algorithm Simulated Annealing
Input: MutationChildren
OutPut: MutationChildrenImprove
Var:  lambda, omega, T, T_F: real;
      S_a, S_c: solutions;
for each child do
    T :- Init_Temperature();
    T_F :- Final_Temperature();
    lambda :- Init_Lambda();
    omega :- Init_Omega();
    Sa :- Generate_Initial_Secuence(child);
    while T > T_F do
        M :- (1/T) + omega
        for i = 1 to M do
            Sc :- Generate_Secuence(S_a);
            if Q(S_c) < Q(S_a) then
                Sa :- Sc;
            else
                if z(0, 1) < e^(-(Q(Sc) - Q(Sa))/T) then
                    Sa :- Sc;
                end if
            end if
        end for
        T :- α(T);
    end while
    improveChild :-S_a;
end for
end Algorithm
```

Figure 3. Simulated Annealing pseudo-code.
Source: Compiled by author.

SA and GRASP achieve them at 84,9%, 59,9% and 55,4% of the runs, respectively. Finally, in terms of high levels of difficulty problems, HGA reaches optimum results at a n average of 78.7% of its runs, while GA, SA and GRASP achieve them at 74,9%, 57,9% and 54,9% of their corresponding runs.

Table 7.
Percentage of times the best result is reached.

Scenarios	Success (%)				
Low level of difficulty	BKS	HGA	GA	SA	GRASP
022_3_4_EP_RAF_ENP	100.00	100.00	100.00	56.66	50.00
024_38_3_EP_RAF_ENP	100.00	100.00	93.33	66.66	63.33
025_38_1_EP_ENP_RAF	100.00	100.00	93.33	63.33	60.00
025_38_1_EP_RAF_ENP	100.00	100.00	90.00	60.00	56.66
039_38_4_EP_RAF_ch1	100.00	100.00	83.33	63.33	43.33
039_38_4_RAF_EP_ch1	100.00	83.33	80.00	66.66	60.00
048_39_1_EP_ENP_RAF	100.00	100.00	100.00	60.00	63.33
048_39_1_EP_RAF_ENP	100.00	80.00	80.00	56.66	53.33
Medium level of difficulty	BKS	HGA	GA	SA	GRASP
022_EP_ENP_RAF_S22_J1	100.00	90.00	90.00	66.66	63.33
022_EP_RAF_ENP_S22_J1	100.00	83.33	80.00	63.33	60.00
024_V2_EP_RAF_ENP_S22_J1	100.00	90.00	90.00	66.66	63.33
024_V2_RAF_EP_ENP_S22_J1	100.00	93.33	90.00	53.33	43.33
029_EP_RAF_ENP_S21_J6	100.00	100.00	93.33	56.66	46.66
029_RAF_EP_ENP_S21_J6	100.00	80.00	73.33	56.66	53.33
048_ch1_EP_RAF_ENP_S22_J3	100.00	90.00	73.33	63.33	63.33
048_ch1_RAF_EP_ENP_S22_J3	100.00	93.33	90.00	53.33	50.00
High level of difficulty	BKS	HGA	GA	SA	GRASP
022_RAF_EP_ENP_S49_J2	100.00	76.67	60.00	53.33	53.33
023_EP_RAF_ENP_S49_J2	100.00	83.33	80.00	73.33	70.00
024_EP_RAF_ENP_S49_J2	100.00	76.66	76.66	66.66	60.00
025_EP_ENP_RAF_S49_J1	100.00	76.66	73.33	50.00	50.00
029_EP_RAF_ENP_S49_J5	100.00	80.00	80.00	66.66	60.00
035_CH2_RAF_EP_S50_J4	100.00	76.66	73.33	53.33	50.00
039_CH1_EP_RAF_ENP_S49_J1	100.00	83.33	80.00	53.33	53.33
039_CH3_EP_RAF_ENP_S49_J1	100.00	76.66	76.66	46.66	43.33

Source: Compiled by author.

5. Conclusions

In this work we presented a novel approach to the solution of ROADEF'05, combining different procedures. They were adapted on the basis of the structure, number of variables and complexity of each scenario. Individual analyses of the Greedy Randomized Adaptive Search Procedure (GRASP), a Genetic Algorithm (GA) and Simulated Annealing (SA) returned satisfactory results. Moreover, in most scenarios GRASP and SA generated solutions with similar features. However, GA converges to better quality results than SA and GRASP. It is interesting to note that GA reaches superior results when the initial individual population is obtained by means of the GRASP. In this context, the Hybrid Genetic Algorithm (HGA) efficiently amalgamates the desirable characteristics of the three meta-heuristics, GRASP, GAs and SA. The experiments sustain this claim since the results achieved with the hybrid technique are better than those obtained by each technique by itself.

Acknowledgments

This work was funded by two research grants: PICT-2011-0396 of the Fondo para la Investigación Científica y Tecnológica (FONCyT) of the Agencia Nacional de Promoción Científica y Tecnológica (AGENCIA) and PGI 24/J056 of the Universidad Nacional del Sur.

References

[1] Solnon, C., Cung, V.D., Nguyen A. and Artigues C., 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, 191 (3), pp. 912-927, 2008. DOI:10.1016/j.ejor.2007.04.033.

[2] Gagne, C. and Zinflou, A., An hybrid algorithm for the industrial car sequecing problem. Proceedings of IEEE Congress on Evolutionary Computation, 1, (1-8), 2012. DOI: 10.1109/CEC.2012.6256122.

[3] Briant, O., Naddef, D. and Mouni, G., Greedy approach and multi-criteria simulated annealing for the car sequencing problem. European Journal of Operational Research, 191 (3), pp. 993-1003, 2008. DOI:10.1016/j.ejor.2007.04.052.

[4] Bautista, J., Pereira, J. and Adenso-Díaz, B., A GRASP approach for the extended car sequencing problem. Journal of Scheduling, 11, pp. 3-16, 2008. DOI: 10.1007/s10951-007-0046-4.

[5] Zinflou, A. and Gagné, C., Tackling the industrial car sequencing problem using GISMOO algorithm. In assembly line - theory and practice, W. Grzechca, Ed., Ed: I-Tech Education and Publishing, pp. 85-106, 2011. DOI: 10.5772/21113.

[6] Estellon, B., Gardi, F. and Nouioua, K., Two local search approaches for solving real-life car sequencing problems. European Journal of Operational Research, 191 (3), pp. 928-944, 2008. DOI: 10.1016/j.ejor.2007.04.043.

[7] Gavranovic, H., Local search and suffix tree for car-sequencing problem with colors. European Journal of Operational Research, 191 (3), pp. 972-980, 2008. DOI: 10.1016/j.ejor.2007.04.051.

[8] Prandtstetter, M. and Raidl, G.R., An integer linear programming approach and a hybrid variable neighborhood search for the car sequencing problem. European Journal of Operational Research, 191 (3), pp. 1004-1022, 2008. DOI:10.1016/j.ejor.2007.04.044.

[9] Frutos, M., Olivera, A.C. y Casal, R., Estudio sobre distintas estrategias utilizadas para establecer secuencias detalladas en entornos productivos. Proceedings of XII International Conference on Industrial Engineering and Operations Management, XXVI Encuentro Nacional Engenharia de Produção 1, pp. 136, 2006.

[10] Olivera, A.C., Frutos, M. y Casal, R., Algoritmos Genéticos: Una estrategia clave para abordar problemas de secuenciamiento en gran escala. Proceedings of XXXIV Jornadas Argentinas de Informática e Investigación Operativa, pp 1-9, 2006.

[11] Olivera, A.C., Frutos, M. and Casal R., Métodos para determinar secuencias de producción en un ambiente productivo complejo. Proceedings of XIII Congreso Latino-Iberoamericano de Investigación Operativa, pp.1-6, 2006.

[12] Fonseca, C.M. and Fleming, P.J., Overview of evolutionary algorithms in multiobjective optimization. Evolutionary Computation, 3 (1), pp. 1-16, 1995. DOI: 10.1162/evco.1995.3.1.1.

[13] Hanne, T., Global multiobjective optimization using evolutionary algorithms. Journal of Heuristics, 6 (3), pp. 347-360, 2000. DOI: 10.1023/A:1009630531634.

[14] Ribeiro, C.C., Aloise, D., Noronha, T.F., Rocha, C. and Urrutia, S., A hybrid heuristic for a multi-objective real-life car sequencing problem with painting and assembly line constraints. European Journal of Operational Research, 191 (3), pp. 981-992, 2008. DOI: 10.1016/j.ejor.2007.04.034.

[15] Resende, M. and González Velarde, J.L,. GRASP: Procedimientos de búsqueda miopes aleatorizados y adaptativos. Revista Iberoamericana de Inteligencia Artificial, 19, pp. 61-76, 2003.

[16] Goldberg, D.E., Genetic algorithms in search, Optimization and machine learning. Addison Wesley Publishing Company, Inc, 1989.

M. Frutos, is an Assistant Researcher of CONICET (National Research Council of Argentina) and Teaching Assistant at the Department of Engineering of the Universidad Nacional del Sur, in Bahía Blanca, Argentina. He completed his undergraduate degree in Industrial Engineering and MSc and PhD in Engineering at his home university. His research focuses on scheduling problems in production and their treatment through metaheuristic methods. He has published in Operational Research, Annals of Operations Research, Dyna, and American Journal of Operations Research among others. He participates actively in the Operations Research community in Latin America.

A.C. Olivera, is an Assistant Researcher of CONICET (National Research Council of Argentina) stationed at the Department of Exact and Natural Sciences of the Universidad Nacional de la Patagonia Austral, Comodoro Rivadavia, Argentina. She has a PhD in Computer Science from the Universidad Nacional del Sur, Bahía Blanca, Argentina and held a post-doctoral position at the University of Malaga, Malaga, Spain. Her research focuses on urban traffic and production chain optimization problems using bio-inspired algorithms. She has published book chapters and several papers in indexed journals and proceedings of refereed international conferences.

F. Tohmé, is a Principal Researcher of CONICET (National Research Council of Argentina) and Full Professor at the Department of Economics of the Universidad Nacional del Sur, in Bahía Blanca, Argentina. A former Fulbright Scholar, he held visiting positions at U.C. Berkeley, Washington University in St. Louis and Endicott College, USA. He holds an undergraduate degree in Mathematics and a PhD in Economics. His research has focused on decision problems, game theory and optimization in the socio-economic settings. He has published in Theory and Decision, Mathematics of Social Sciences, Artificial Intelligence, Mathematical and Computational Modeling, and Annals of Operations Research among others.