FITNESS-DISTANCE ANALYSIS
OF A CAR SEQUENCING PROBLEM

Marek KUBIAK *, Andrzej JASZKIEWICZ, Paweł KOMINEK

Abstract. The paper describes a fitness–distance analysis of a car sequencing problem. It defines 5 similarity measures for solutions of the problem, describes computational experiments and provides values of determination coefficients between fitness and similarity, which are an indicator of fitness–distance correlation (or 'big valley'). The analysis reveals certain correlations of fitness and two types of similarity for 4 of 5 types of available instances. This results might motivate such designs of metaheuristics for these types of instances which would exploit the structure in fitness landscapes.

Keywords: similarity measure, fitness-distance correlation, metaheuristics, combinatorial optimisation

1 Introduction

Adaptation of a metaheuristic algorithm to a given combinatorial optimization problem is not an easy task. There are many frameworks for such algorithms: simulated annealing, tabu search, ant or evolutionary algorithms. Such variety of methods makes the problem of choice among them the first issue.

But even when a metaheuristic is already chosen it is not obvious how to adapt the given scheme to the problem at hand. This involves choosing or designing specific components for the algorithm (e.g. neighbourhood operators for a local search, cooling scheme for a simulated annealing or genetic operators for an evolutionary algorithm) so it becomes an efficient one.

Unfortunately, at the moment there are hardly any precise yet general enough design schemes for metaheuristics, which could help a designer to adapt their chosen algorithm to the problem without the need for exploration of many different options.

*Institute of Computing Science, Poznan University of Technology, Piotrowo 2, 60-965 Poznań, Poland, Marek.Kubiak@cs.put.poznan.pl
for operators or parameters. Hence, there is a need of elaborating guidelines for
designers of algorithms to help them in their task of adapting a metaheuristic to the
problem at hand.

Some recent research in this direction focused firstly on analysis of search spaces
(fitness landscapes) of combinatorial optimisation problems in order to acquire inform-
ration on such properties of landscapes which may have impact on optimisation. For
example, in [11] there are 3 interesting fitness landscape properties mentioned which
have such impact: random walk correlation length, size of attractor of local optima
and fitness–distance correlation (FDC).

The fitness–distance analysis (FDA, which is the theme of this paper and provides
values of FDC) reveals information about a structure of a landscape called ‘big valley’
or ‘global convexity’ ([2], [8]). This structure is a trend in which good solutions of the
considered problem are clustered in some small part of the whole search space, better
solutions being even closer to each other than worse ones and global optima some-
where in the centre of this cluster. This phenomenon was found in fitness landscapes
of many instances of combinatorial optimisation problems: binary benchmarks [8],
travelling salesman [2], quadratic assignment, graph bipartitioning, binary quadratic
programming [11], [12], flow–shop [14] and job–shop scheduling problems [15], set cov-
ering [3], capacitated vehicle routing [9], and a practical version of the vehicle routing
problem [6].

Although there are examples of instances of some problems which seem not to
reveal any ‘big valley’ ([8], [9], [11], [15]) or even exhibit an inverse trend ([1], [8]),
common existence of this feature in many problems inspired some researchers to ex-
ploit it during search by means of metaheuristics. These ideas included: iterated local
search [2], memetic algorithms with distance preserving crossover operators [6], [11],
[12], or defining a small core subproblem for a metaheuristic [3]. In all these cases very
efficient algorithms were designed based on results of fitness–distance analysis. Thus,
perhaps the FDA may become a method of acquiring knowledge about a problem
which could be later used to motivate the design of components of metaheuristics.
Indeed, in case of memetic algorithms some guidelines for such design were given e.g.
in [6] and [12].

Based on this premise the paper focuses on the fitness-distance analysis of a prac-
tical car sequencing problem: describes similarity measures necessary to perform the
analysis, the process of analysis itself and its results. These results may then be used
during adaptation of metaheuristics to the problem.

2 The car sequencing problem

The car sequencing problem (CSP) considered here was the subject of an open com-
putational contest: ROADEF Challenge 2005 [5]. The goal of the problem is to find
the best possible sequence for a given set of cars, which are then put in this order on
a production line (a paint shop and an assembly line).

All cars from a given set have to be manufactured. Each vehicle is described by a
unique identifier, a paint colour code and a vector of binary properties. This vector
defines which pieces of additional equipment have to or must not be mounted in the car during final assembly operations (e.g. sun-roof).

The only hard constraint in the CSP is imposed by technological characteristics of the paint shop: the final sequence must not contain blocks of cars with the same colour longer than the given limit. This limit is called the paint batch limit ($PBL$).

In each instance of the problem there is a set of soft constraints, so-called ratio constraints (RCs), which are means of expressing the preference of the assembly line. One RC is related to exactly one binary property of cars mentioned earlier. An RC is defined by a ratio $N/D$ and specifies that at most $N$ vehicles in each continuous block of length $D$ may have the related property set. If this ratio is exceeded, then the number of extra cars in each block is added to the number of violations of this RC (a penalty). There is a specific approach to computation of violations at the beginning and at the end of a sequence, but since it does not change the whole concept of RCs it is not presented here. The reader interested in these details is referred to the full description of the problem available at the contest website [5].

Very similar concepts of constraints on binary properties of cars were considered in [13] and [4] (so-called capacity and subinterval quantity constraints); the relevant penalty functions had different mathematical formulations, though.

An example of the computation of the number of violations is shown in figure 1, where one can see a sequence $s$ of 8 cars and 6 stages of the computation, one for each block of $D = 3$ cars. $N$ equal to 1 means that there is no penalty for one (or less) car with a „1” (the binary property set). If there is more than one such car in a block, then some violations occur. In this case the total number of violations is $V_{N_1}(s) = 4$.

The set all of RCs is divided into two subsets: high priority ratio constraints (HPRCs) and low priority ones (LPRCs). The sums of numbers of violations by a sequence $s$ of RCs in those two sets, $V_{N_{HPRCs}}(s)$ and $V_{N_{LPRCs}}(s)$, form two quantitative characteristics of the solution $s$.

**Figure 1**: Computation of violations $V_{N_1}$ of $RC_1$ for consecutive continuous subsequences of cars. Only one binary property of cars is shown.
Another such characteristic is computed based on colours of vehicles. It is the total number of paint colour changes between two consecutive cars in a sequence $s$, $PCC(s)$. This number expresses the preference of the paint shop to have as long batches of the same colour as possible (but not exceeding PBL).

Finally, the value of the objective function $f(s)$ for a sequence $s$ is a weighted sum of the 3 mentioned characteristics and should be minimized:

$$f(s) = w_{HPRCs} \cdot VN_{HPRCs}(s) + w_{LPRCs} \cdot VN_{LPRCs}(s) + w_{PCC} \cdot PCC(s)$$

The vector of weights $w = (w_{HPRCs}, w_{LPRCs}, w_{PCC})$ is constant in each instance of the CSP. There are only 3 possible values of this vector (see table 1).

Additionally, for some instances another piece of information is available, describing the predicted hardness of high priority RCs. It may say that HPRCs are expected to be easy or difficult to optimise.

The possible values of the vector of weights and the HPRCs difficulty parameter give rise to 5 main types of instances; they are listed in table 1 together with some basic description of the data sets. There are 80 instances in total (except 4 test instances) and all of them were provided by Renault, the contest partner. These instances form three disjoint sets: A (initial set of instances), B (made available after the initial stage) and X (final, unknown to competitors before the end of the Challenge).

### Table 1: Basic description of the types of instances of the CSP available at [5].

<table>
<thead>
<tr>
<th>inst. type</th>
<th>HPRCs difficulty</th>
<th>weights $w$</th>
<th># inst.</th>
<th>#cars (min-max)</th>
<th>#HPRCs</th>
<th>#LPRCs</th>
<th>PBL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>easy</td>
<td>$(10^6, 1, 10^3)$</td>
<td>25</td>
<td>65-1319</td>
<td>1-11</td>
<td>0-19</td>
<td>10-500</td>
</tr>
<tr>
<td>2</td>
<td>difficult</td>
<td>$(10^6, 1, 10^3)$</td>
<td>11</td>
<td>128-1315</td>
<td>2-9</td>
<td>0-16</td>
<td>10-150</td>
</tr>
<tr>
<td>3</td>
<td>easy</td>
<td>$(10^6, 10^3, 1)$</td>
<td>13</td>
<td>65-1231</td>
<td>1-11</td>
<td>1-19</td>
<td>10-500</td>
</tr>
<tr>
<td>4</td>
<td>difficult</td>
<td>$(10^6, 10^3, 1)$</td>
<td>9</td>
<td>128-1315</td>
<td>2-9</td>
<td>2-16</td>
<td>10-150</td>
</tr>
<tr>
<td>5</td>
<td>easy</td>
<td>$(10^3, 1, 10^6)$</td>
<td>22</td>
<td>65-1270</td>
<td>1-11</td>
<td>0-19</td>
<td>10-1000</td>
</tr>
</tbody>
</table>

3 Similarity measures for solutions of the problem

Similarity (or distance) measures are the key elements of any fitness–distance analysis. They express research intuition about the problem: what kind of information about solutions might be important from the point of view of the objective function.

At the initial stage of the challenge, the authors’ intuition indicated that existence/preservation of certain subsequences of cars in solutions may be crucial for good values of the objective. It can be clearly seen from the definition of this function, at least from the point of view of RCs, that vehicles with binary properties set should be put in each subsequence in sufficient density in order not to cause violations in
other parts of the same solution. This idea led to the definition of similarity in terms of common subsequences.

Another concept was to check if consecutive couples of cars are similar in good solutions. It was considered a simplified version of the idea of common subsequences, since it would not check triples or longer parts of solutions. This idea is reflected in the definition of similarity as common succession relations.

The last idea was to compare solutions with respect to positions of cars and this motivated the definition of similarity as common positions. The authors wanted to check if good solutions should preserve certain positions in sequences. The initial intuition was that this should not be the case; the authors expected positions of cars to be irrelevant to the objective function. Consequently, it was interesting to see if FDA could confirm this kind of ‘no relevance’ hypothesis.

### 3.1 Groups and weaker groups of vehicles

Although each car was given a unique identifier in each input set, the cars having the same paint colour code and the same binary properties are indiscernible from the point of view of the objective function (a similar observation was made in [4] in case of a slightly different problem). Therefore, in order to break symmetries between solutions of the CSP, the authors introduced the concept of groups of vehicles (also called ordinary groups), i.e. sets of vehicles with identical properties. Consequently, while computing similarity all solutions have identifiers of vehicles replaced with indices of the related groups.

Additionally, the authors relaxed the concept of ordinary groups and defined so-called weaker groups, i.e. sets of vehicles which have only the most important properties identical (those with the associated weight $w = 10^6$); other properties of vehicles within one weaker group may be completely different: colour or LPRCs in instances of type 1–4, and all RCs in instances of type 5. The idea behind weaker grouping came from an observation that for instances of type 2 and 4 (with HPRCs most important and difficult) the impact of colour and LPRCs on the values of the objective function might be minimal, due to the fact that $VN_{HPRC}(s)$ is very unlikely to get zeroed. Thus, these properties should not be considered at all during computations of similarity. Since indices of weaker groups are easily computable, they were also used with other types of instances.

### 3.2 Similarity as common positions: $sim_{cp}$

This similarity measure is based on the idea of Hamming distance: it compares solutions on the position–by–position basis. The value of similarity $sim_{cp}$ is the number of positions with the same indices of groups in the compared solutions (common positions, $cp$). In figure an example of such a comparison is given.
3.3 Similarity as common subsequences: \( \text{sim}_{cs} \) and \( \text{sim}_{cs\text{wg}} \)

The second measure evaluates similarity in the sense of common subsequences (\( cs \)). This concept, although independently developed, is similar to the one described in [10], where sets of common subsequences are computed to determine similarity of two genomes; it has a bit different mathematical formulation here.

First, all common subsequences (longer than 1) of two compared solutions are computed using a generalized suffix tree. Second, in each solution separately, a subset of maximal subsequences is found; these are subsequences which have the maximum length and are not wholly included in any other subsequence, though they may partially overlap each other. An illustration of a result of this process is shown in figure 3. One can see two solutions containing only two types of cars (indices 1 and 2). Beneath them maximal common subsequences are marked with solid lines. As mentioned before, sets of such subsequences might be different in the compared solutions; subsequences do partially overlap in some cases, as well.

\[
\text{sim}_{cs}(s_i, s_j) = 8
\]

\[
\text{sum}_{cs}(s_i, s_j) = 70
\]

**Figure 3:** Comparison of two sequences by \( \text{sim}_{cs} \) measure. Solid intervals indicate maximal common subsequences; dashed ones show their proper subsequences included in computation of similarity (see text).

When the common subsequences are found, their lengths are summed up with proper weights. In order to achieve this goal the length of each subsequence is increased by lengths of all its proper, shorter subsequences (at least 2 position long) which finish at the same position; the subsequences which are contained in the next maximal common subsequence are not added in order not to inflate the value of similarity. The purpose of weights is to give preference to single and very long sub-
sequences over many shorter but overlapping ones. In figure B these shorter sub-
sequences included in computation are indicated with dashed lines.

The sum of lengths of all subsequences found so far in both solutions (the solid
and dashed intervals in figure B) defines the value of $\text{sum}_{cs}(s_1, s_2)$. Finally, the value
of similarity is computed with the following formula:

$$\text{sim}_{cs}(s_1, s_2) = \frac{1}{2}(\sqrt{4 \cdot \text{sum}_{cs}(s_1, s_2)} + 9 - 1)$$

This expression ensures a sort of normalization of values of this measure: the minimum
is 1 (no common subsequences) and the maximum is equal to the length of a solution
(identical solutions compared). The value of $\text{sim}_{cs}$ is also equivalent to the length
of such a maximal subsequence which would be the only common one in the two
compared solutions.

The third measure, $\text{sim}_{cswg}$, is computed exactly in the same way as $\text{sim}_{cs}$, though
the compared sequences contain indices of weaker groups only.

### 3.4 Similarity as common succession relations: $\text{sim}_{csuc}$ and $\text{sim}_{csucwg}$

The fourth measure of similarity counts the number of common relations of succession
between indices of groups in compared solutions. A pair of indices of groups is in this
relation if the second index immediately succeeds the first one.

An example of computation of $\text{sim}_{csuc}$ may be seen in figure H. In the two pre-
sented solutions there are exactly 8 common succession relations (emphasized with
arcs under the pairs of indices). Note that the last pair in solution $s_1$, $(1, 2)$, is not a
common succession relation because it does not have its counterpart in $s_2$ (there are
too many such pairs in $s_1$).

The last measure of similarity, $\text{sim}_{csucwg}$, is identical in definition to the previous
one, $\text{sim}_{csuc}$, but it is computed on the basis of indices of weaker groups (just like
$\text{sim}_{cswg}$).

![Figure 4](image)

**Figure 4**: Comparison of two sequences by $\text{sim}_{csuc}$ measure. Arcs indicate common
sucession relations (see text).
4 Fitness–distance analysis of the car sequencing problem

4.1 The computational experiment

In order to compute the values of FDC for instances of the problem the following experiment was conducted:

1. A set of 500 pairs of random local optima was generated for each instance (1000 solutions in total); random solutions were used as starting points and then greedy local search was performed, using a one–vehicle–shift operator (one vehicle is removed from a sequence and put in some other feasible place).

2. In all sets of local optima and for each pair of solutions two values of the objective function were computed ($f(s_1), f(s_2)$); also values of all similarity measures were evaluated ($sim(s_1, s_2)$).

3. In each set of local optima two values of correlation were computed for each similarity: $r(f(s_1), sim(s_1, s_2))$ and $r(f(s_2), sim(s_1, s_2))$; the value of the linear determination coefficient was computed as: $r^2 = r^2(f(s_1), sim(s_1, s_2)) + r^2(f(s_2), sim(s_1, s_2))$.

The computed values of the determination coefficient should not be biased because no correlation between $f(s_1)$ and $f(s_2)$ was observed. Therefore, values of $r^2(f(s_1), sim(s_1, s_2))$ and $r^2(f(s_2), sim(s_1, s_2))$ can be added.

A comment on this FDA method should be given. This way of computing FDC is different from those proposed earlier, e.g. in [2], [6], [8], [9], [12], [14] and others. It is based on a 3 dimensional model of a fitness landscape, where pairs of solutions $(s_1, s_2)$ are subjects to measurement of values: $f(s_1), f(s_2), sim(s_1, s_2)$. The authors think, this way of analysis is more practical and statistically accurate than the ones used in the majority of the studies listed above. However, there is not enough space in this paper to describe the advantages and disadvantages of this and other approaches; these issues deserve a separate discussion.

4.2 Results and discussion

The computed values of the determination coefficients are shown in table 2.

At first glance, all the presented average values of $r^2$ seem to be small and insignificant. However, an interpretation of these values has to be made with the method of their evaluation in mind: these are values of two independent correlation coefficients squared and added. For instance, $r^2 = 0$, 18 is based on two correlations $r = 0$, 3 (in all cases both of the correlations $r(f(s_1), sim(s_1, s_2))$ and $r(f(s_2), sim(s_1, s_2))$ were almost equal).

In case of this study also the fitness-distance (FD) plots are not exactly comparable to the ones published in other papers on FDA. The basic plot resulting from this analysis is a 3D plot (see figure 4) having values of the objective function in two horizontal axes ($f(s_1)$ and $f(s_2)$) and values of similarity in the vertical one. In
Table 2: Values of determination coefficients for all similarity measures grouped by instance set and type.

<table>
<thead>
<tr>
<th>set</th>
<th>type</th>
<th>#inst.</th>
<th>sim_{cp} ( \text{avg}(r^2) )</th>
<th>sim_{cs} ( \text{avg}(r^2) )</th>
<th>sim_{cswg} ( \text{avg}(r^2) )</th>
<th>sim_{csuc} ( \text{avg}(r^2) )</th>
<th>sim_{csucwg} ( \text{avg}(r^2) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>4</td>
<td>0.62%</td>
<td>34.90%</td>
<td>10.03%</td>
<td>22.29%</td>
<td>1.37%</td>
</tr>
<tr>
<td>A</td>
<td>2</td>
<td>4</td>
<td>0.27%</td>
<td>1.20%</td>
<td>24.08%</td>
<td>0.78%</td>
<td>0.49%</td>
</tr>
<tr>
<td>A</td>
<td>3</td>
<td>1</td>
<td>0.08%</td>
<td>0.11%</td>
<td>0.10%</td>
<td>0.09%</td>
<td>2.28%</td>
</tr>
<tr>
<td>A</td>
<td>4</td>
<td>3</td>
<td>0.69%</td>
<td>0.99%</td>
<td>34.62%</td>
<td>0.53%</td>
<td>1.11%</td>
</tr>
<tr>
<td>A</td>
<td>5</td>
<td>4</td>
<td>0.31%</td>
<td>29.13%</td>
<td>44.27%</td>
<td>16.35%</td>
<td>24.10%</td>
</tr>
<tr>
<td>B</td>
<td>1</td>
<td>9</td>
<td>0.57%</td>
<td>19.89%</td>
<td>0.51%</td>
<td>9.02%</td>
<td>0.60%</td>
</tr>
<tr>
<td>B</td>
<td>2</td>
<td>6</td>
<td>0.62%</td>
<td>8.18%</td>
<td>4.66%</td>
<td>4.75%</td>
<td>0.46%</td>
</tr>
<tr>
<td>B</td>
<td>3</td>
<td>9</td>
<td>0.93%</td>
<td>8.77%</td>
<td>0.72%</td>
<td>3.32%</td>
<td>0.36%</td>
</tr>
<tr>
<td>B</td>
<td>4</td>
<td>6</td>
<td>0.75%</td>
<td>3.28%</td>
<td>6.46%</td>
<td>1.30%</td>
<td>0.27%</td>
</tr>
<tr>
<td>B</td>
<td>5</td>
<td>15</td>
<td>0.50%</td>
<td>17.99%</td>
<td>39.45%</td>
<td>8.26%</td>
<td>5.14%</td>
</tr>
<tr>
<td>X</td>
<td>1</td>
<td>12</td>
<td>0.48%</td>
<td>29.98%</td>
<td>3.85%</td>
<td>16.94%</td>
<td>0.45%</td>
</tr>
<tr>
<td>X</td>
<td>2</td>
<td>1</td>
<td>0.12%</td>
<td>0.43%</td>
<td>1.19%</td>
<td>0.64%</td>
<td>0.07%</td>
</tr>
<tr>
<td>X</td>
<td>3</td>
<td>3</td>
<td>0.88%</td>
<td>0.71%</td>
<td>1.25%</td>
<td>0.60%</td>
<td>0.69%</td>
</tr>
<tr>
<td>X</td>
<td>4</td>
<td>0</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>X</td>
<td>5</td>
<td>3</td>
<td>0.86%</td>
<td>46.32%</td>
<td>54.45%</td>
<td>31.33%</td>
<td>41.29%</td>
</tr>
<tr>
<td>A</td>
<td>(all)</td>
<td>16</td>
<td>0.44%</td>
<td>16.50%</td>
<td>26.09%</td>
<td>9.96%</td>
<td>6.84%</td>
</tr>
<tr>
<td>B</td>
<td>(all)</td>
<td>45</td>
<td>0.65%</td>
<td>13.26%</td>
<td>14.88%</td>
<td>6.03%</td>
<td>2.00%</td>
</tr>
<tr>
<td>X</td>
<td>(all)</td>
<td>19</td>
<td>0.59%</td>
<td>26.38%</td>
<td>11.29%</td>
<td>15.78%</td>
<td>6.92%</td>
</tr>
<tr>
<td>(all)</td>
<td>1</td>
<td>25</td>
<td>0.54%</td>
<td>27.13%</td>
<td>3.64%</td>
<td>14.95%</td>
<td>0.65%</td>
</tr>
<tr>
<td>(all)</td>
<td>2</td>
<td>11</td>
<td>0.45%</td>
<td>4.94%</td>
<td>11.41%</td>
<td>2.93%</td>
<td>0.43%</td>
</tr>
<tr>
<td>(all)</td>
<td>3</td>
<td>13</td>
<td>0.85%</td>
<td>6.25%</td>
<td>0.79%</td>
<td>2.44%</td>
<td>0.58%</td>
</tr>
<tr>
<td>(all)</td>
<td>4</td>
<td>9</td>
<td>0.73%</td>
<td>2.52%</td>
<td>15.85%</td>
<td>1.04%</td>
<td>0.55%</td>
</tr>
<tr>
<td>(all)</td>
<td>5</td>
<td>22</td>
<td>0.52%</td>
<td>23.88%</td>
<td>42.37%</td>
<td>12.87%</td>
<td>13.52%</td>
</tr>
<tr>
<td>(all)</td>
<td>(all)</td>
<td>80</td>
<td>0.59%</td>
<td>17.02%</td>
<td>16.27%</td>
<td>9.13%</td>
<td>4.14%</td>
</tr>
</tbody>
</table>

An interactive environment (such as GNU Octave) these plots can be easily rotated, allowing a more detailed insight into the possible relationships between variables than in the case of static plots.

Two dimensional FD plots presented here (see figures 6, 7) result from cuts through clouds of points like in figure 5 along the plane \( f(s_2) = f(s_1) \). More precisely, in order not to have the plots with few points even pairs of solutions which are close to the plane are shown (\(|f(s_2) - f(s_1)| \leq \epsilon \cdot f(s_1)\), with \( \epsilon \leq 10\% \)).

First observation is that \( \text{sim}_{cp} \) is not related to quality of solutions at all: values of \( r^2 \) in all instances are virtually 0. Moreover, all FD plots looked like the one shown on the left side of figure 6, where the objective function is not correlated with.
Figure 5: Examples of 3D fitness–distance plots for an instance with 485 cars and $sim_{cs}$.

$sim_{cp}$. These facts strongly confirm the initial hypothesis about the insignificance of positions of vehicles in the CSP.

In case of $sim_{cs}$ there is better news for instances from set 1 and 5: on average $r^2 > 0.2$; instances of type 2, 3 and 4 generally do not exhibit correlation of quality and $sim_{cs}$.

For a reader with experience in statistics values of $r^2 \approx 0.2$ might seem insignificant. Therefore, the authors stress once more the fact that these are sums of two squared values of independent correlations. For example, Jones and Forrest (in [8]) based some of their conclusions on values $r \approx 0.15$ (they classified problems with $r \geq 0.15$ as straightforward for optimisation with genetic algorithms). It may be clearly seen that a threshold of $r^2 \approx 0.2$ used by the authors is more significant that the one used in the cited work, because it corresponds to both correlations $r \approx 0.31$.

The reader should also remember the fact that in this experiment there are two sources of variability of similarity (two independently drawn local optima) as opposed to only one source of variability in the previous studies, where one solution was compared always with a fixed point (a nearest global optimum, like in [2], [3], [8], [11], [14]) or a large set of solutions (like in [2], [6], [9]). Moreover, the values of similarity are not averaged here, but are raw values. Therefore, it should not be surprising that correlations do not indicate strong relationships and variability of similarity in FD plots is considerable; the picture of the search space of the CSP is not beautified here.

The relationship between the objective function and $sim_{cs}$ may also be noted in the 3D plots shown in figure 5. Both plots show the same set of pairs of local optima for an instance of the problem, but from a different angle. The one on the left shows the cloud of points from the back (the points related to pairs of best solutions lay above the far corner of the base plane); this cloud has almost triangular shape, slightly elevated in its far end. The plot on the right shows the cloud from a side (the points related to pairs of best solutions are on the left now). The elevation of the set of points is now more clearly seen: better solutions tend to be more similar to each other than worse ones. This kind of beneficial elevation may also be seen in figure 6.
As said above, this positive relationship is visible only for types 1 and 5 of instances. Type 1 contains HPRCs which are easy to optimise (see table 1) and are usually zeroed, and colour as the second important component. Type 5 has the colour component as the most important. This fact suggests that the importance of colour promotes certain subsequences of vehicles in good solutions. Conversely, in other types of instances the component of colour is not that important in the objective function and \( \text{sim}_{cs} \) does not relate well with it.

The measure \( \text{sim}_{csuc} \) exhibits similar relationship with \( f(s) \) as \( \text{sim}_{cs} \) does, but this relationship is always weaker than in case of common subsequences. The authors’ guess is that the simple relation of succession between indices of groups is not that important to the objective as the existence of longer common subsequences; certain pairs of consecutive vehicles may appear simply by chance also in worse local optima.

When indices of ordinary groups are replaced with those of weaker groups and \( \text{sim}_{cswg} \) is used, then larger values of \( r^2 \) appear for instances of type 2, 4, 5 from set A, and only of type 5 for sets B, X. In FD plots the positive relationship between \( f(s) \) and \( \text{sim}_{cswg} \) was also observed (see figure 7, the plot on the right). From the results for set A there can be derived the conclusion that when less important properties of cars are overlooked then a significant relationship between quality and subsequence of vehicles is revealed. Most surprisingly, this phenomenon seems to vanish in set B. (It is hard to determine its status in set X, since there is only one instance of types 2 and 4 available in this set).

The results of experiments with \( \text{sim}_{cswg} \) and type 5 of instances indicate the strongest correlation. This is not surprising, though; an index of a weaker group in this case represents simply the colour of a car and in the CSP optimisation of colour only is easy (may be done by means of an algorithm polynomial in time). Thus, it appears that similar subsequences of colour are important for the objective function but colour does not require high optimisation effort.

The comparison of \( r^2 \) for \( \text{sim}_{cswg} \) and \( \text{sim}_{csucw} \) indicates a similar relationship as in the case of \( \text{sim}_{cs} \) and \( \text{sim}_{csuc} \): the first one is always greater than the second one if only values of the determination are significantly large (\( r^2 > 0, 2 \)).

![Figure 6](image_url): Examples of fitness–distance plots for instances with 1161 and 376 cars. Lines of linear regression are also shown.
The type 3 of instances, where the colour is always the least important component of the objective function and there are usually many RCs, did not reveal any relationship between similarity and quality in these FD experiments. Only 2 instances in 13 of this type revealed significant correlation between quality and similarity sim\textsubscript{cs} (see figure 7, the left plot). It appears as though the importance and large number of binary properties related to RCs negatively influences the relationship between the objective function and this kind of similarity.

To summarise, the correlation between quality and similarity of solutions depends not only on the type of similarity but also on the type of an instance. The authors deem that of the similarity measures defined in this work only sim\textsubscript{cs} and its variant, sim\textsubscript{cswg}, deserve attention.

5 Conclusions

The study on the CSP described in this paper revealed existence of correlations between quality of solutions and similarity between them in certain sets of instances, if similarity is understood as common subsequences of types of vehicles. Unfortunately, the values of the linear determination coefficients are not as high as the authors would like them to be, and the relationships in the fitness–distance plots are not as strong, as well. But despite that fact the authors believe that even if such correlations are small they help in optimisation. Therefore, exploitation of the idea of common subsequences in a metaheuristic algorithm for the CSP would be beneficial in case of types 1, 2, 4 and 5 of instances.

In this place a remark on the CSP should be made; its first description (for the first stage of the ROADEF Challenge 2005) stated that weights of components of the objective function could only be 10\textsuperscript{4}, 10\textsuperscript{2} or 1. The values of correlations computed for this first formulation of the CSP were higher than after the weights were changed to those values presented in this paper (see table I).

Nevertheless, the authors attempted at defining such genetic operators for a memetic
algorithm which would exploit the revealed correlation of quality and common subsequences. These operators were:

- two crossover operators preserving in an offspring all common subsequences identified in parents (they differed in the level of randomness introduced in the offspring),
- a mutation operator randomly reinserting vehicles in a short subsequence of the mutated solution (at most 5% of the sequence length).

The operators were implemented by the authors in a memetic algorithm. Some initial results of its performance were reported in [7]. Here it just may be said that the presence of these genetic operators in the memetic algorithm was not negligible. They significantly improved the performance of the algorithm compared to a multiple start local search run with the same time limit as stated in the Challenge description (600 seconds).

What is more important, the results of this study (and others of this kind) do not have to be followed by an implementation of a memetic algorithm. Fitness-distance correlations may also be exploited in other types of metaheuristics.

Acknowledgements

The work of the authors was supported by Polish State Committee for Scientific Research, KBN, through research grant no. 3 T11F 004 26.

References


