On the hardness of the Quadratic Assignment Problem with meta-heuristics

Eric Angel
Université de Paris Sud
L.R.I, CNRS-URA 410
Centre d'Orsay, 91405 Orsay, France
e_mail: angel@lri.fr

and

Vassilis Zissimopoulos
Université Paris Nord
LIPN, CNRS-URA 1507
93430 Villetaneuse, France
e_mail: vz@lipn.univ-paris13.fr

Abstract

Meta-heuristics are a powerful way to approximately solve hard combinatorial optimization problems. However, for a problem, the quality of results can vary considerably from one instance to another. Understanding such a behaviour is important from a theoretical point of view, but also has practical applications such as for the generation of instances during the evaluation stage of a heuristic.

In this paper we propose a new complexity measure for the Quadratic Assignment Problem in the context of meta-heuristics based on local search, e.g. simulated annealing. We show how the ruggedness coefficient previously introduced by the authors, in conjunction with the well known concept of dominance, provides important features of the search space explored during a local search algorithm, and gives a rather precise idea of the complexity of an instance for these heuristics. We comment previous experimental studies concerning tabu search methods and genetic algorithms with local search in the light of our complexity measure. New computational results with simulated annealing are presented.

Keywords- quadratic assignment problem, local search, complexity measures.

1 Introduction

Given two $n \times n$ matrices $F = (f_{ij})$ and $D = (d_{ij})$, the Quadratic Assignment Problem (QAP) can be stated as follows:

$$\min_{\pi} \sum_{ij} f_{ij} d_{\pi(i)\pi(j)}.$$

The QAP is known to be NP-hard [10], and non approximable [17]. In the context of location theory, f_{ij} is the flow of materials from plant i to plant j, and d_{ij} represents the distance from location i to location j. The objective is to find an assignment of all plants to locations such that the sum of products distance $\times$ flow is minimized. We shall assume, as it is usually the case, that the matrices F and D are symmetric with a null diagonal, and the cost function to be minimized is written $C(\pi) = \frac{1}{2}\sum_{i,j=1}^{n} f_{ij} d_{\pi(i)\pi(j)}$.

The QAP is a computationally very hard combinatorial optimization problem, and instances with a size greater than 30 are practically not solvable to optimality. Fortunately it has been noticed that heuristics give remarkably good results, and specially those based on local search, e.g. simulated annealing and tabu search. For example, Maniezzo, Dorigo and Colorni have concluded in [21]: "[...] reflects a QAP property which was not yet put into evidence in the literature: any approach based on local search is bound to be very effective as a heuristic for QAP", and Connolly writes in [8]: "Simulated annealing is an extremely efficient heuristic for the QAP". These methods use the 2-exchange neighborhood which consists in exchanging the locations of two plants. More formally, given a permutation $\pi = (\pi(1), \ldots, \pi(i), \ldots, \pi(j), \ldots, \pi(n))$, its neighbors are the $n(n-1)/2$ permutations of the form $\pi' = (\pi(1), \ldots, \pi(j), \ldots, \pi(i), \ldots, \pi(n))$ for $1 \leq i < j \leq n$, obtained from π by a swap.

In [1] the authors have presented a theoretical performance guarantee result for a basic local search procedure, nonetheless the quality of solutions obtained with meta-heuristics can vary considerably from one instance to another. This leads us to the notion of a complexity measure for the QAP. Generally speaking, by this we mean that given an algorithm and an instance, to be able to characterize the difficulty of that instance, and to say if the algorithm is adapted or not for it. Of course, the interpretation of the difficulty is function of the method used, and the aim pursued. For example, it can be a prediction of the computational time needed to solve the problem by an exact algorithm, or an indication of how far one can be from the optimal solution when an approximate method is used.

This complexity measure has important applications. It can be useful for choosing among several methods of resolution the more adapted for the instance considered, and also in the evaluation stage of an algorithm. Indeed, when you test it, it is important to ensure that instances you are generating are well distributed in the "complexity space" of the problem. You should have both easy and hard instances.

The next of this paper is organized as follows: in section 2 we discuss the concept of flow dominance and review some previous works on the hardness of QAP instances. We generalize the concept of dominance in section 3. In section 4 we comment the notion of the ruggedness of a landscape and its link with local search, then we discuss about our two previously defined autocorrelation and ruggedness coefficients. In section 5, a precise characterization of the difficulty of a QAP instance for local search based heuristics is presented. The section 6 is devoted to experimental evaluations.

2 The flow dominance

The standard complexity measure for the QAP is the *flow dominance*, which was introduced by Vollmann and Buffa [22]. It measures to what extent the flow matrix F shows "dominant" flow patterns. A possible definition of the flow dominance is given by [16] [11]:

$$\begin{aligned}
fd(F) &= 100\,\frac{\sigma}{\mu}, \text{ with} \\
\mu &= \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} f_{ij}, \text{ the mean and} \\
\sigma &= \sqrt{\frac{1}{n^2-1}\sum_{i=1}^{n}\sum_{j=1}^{n}(f_{ij}-\mu)^2},
\end{aligned}$$

the unbiased estimator of the standard deviation. When the flow dominance is high, it means that there are few plants which exchange a lot of materials between themselves, and few with others. In the past, this parameter was used in a different way than now. When the flow dominance was high it meant that the instance could be easily resolved by visual based methods without employing sophisticated algorithms. To pick out the few plants which contributed to the majority of flows and to place them close to each other was enough to obtain a satisfactory solution.

Other derived parameters have been proposed, but serious doubts were raised about their interpretability, and they were proved to be largely unusable [11]. Yet, the flow dominance has some drawbacks too. First, different definitions of this parameter have been proposed since its introduction [11]. Some authors include the zeros on the diagonal matrix, others do not, and sometimes only the lower triangle under the diagonal is taken into account if the matrix is symmetric. Second, it takes into account only one matrix, instead of both matrices F and D. Third, it is independent of the chosen method applied to a problem instance.

In the past, the flow dominance was used in the study of exact methods of type branch and bound. In [15] Mautor and Roucairol have noticed that the higher is the flow dominance and the faster is a branch and bound algorithm. We think these results can be interpreted as follows. When the flow dominance is high it means that flows are very different to each other and so the algorithm can cut large portions of the search tree. For approximate algorithms the flow dominance was used, among others, in [5], [20] and [16], but the situation is more involved.

Bachelet, Preux and Talbi have observed in their parallel hybrid meta-heuristics combining tabu search and genetic algorithms [5], that tabu search is well fitted to instances with low flow dominance but its performance decreases in other cases. In contrary, genetic algorithms perform well on instances with a high flow dominance. Moreover, they write: “High flow dominance implies, for the landscape associated to the instance, a collection of rugged plateaus at different levels of fitness; low flow dominance corresponds to a rugged landscape without any dominating peak.”

Taillard in his experimental study concerning tabu methods [20], has found that for randomly instances generated with a uniform distribution, which typically have a low flow dominance, different local optima were only slightly correlated, and that the objective function values of the solutions lied in relatively tight bounds. Consequently, good solutions can be obtained easily, but it is very difficult to reach the global optimum. He concludes that these instances are not interesting when testing heuristics or trying to improve them, and he proposes non uniform distributions in order to obtain instances closer to real-life ones.

In the paper of Merz and Freisleben about the combining of genetic algorithms with local search [16] one can read “A high flow dominance indicates that few entries in the flow matrix have a high influence on the total cost. This may have a large impact on the efficiency of the search for both heuristics and exact methods. If almost all entries are equally sized, the flow dominance is low. Local changes in a solution can result in a extremely different fitness, and each assignment of a facility to a location has a large impact on the other assignments, making the instance hard to solve.”.

The two conclusions seem conflicting. Let us mention however that these last two studies were not aimed at studying the effect of the flow dominance. We think it is necessary to distinguish two aspects: the amplitude of variation of the cost function and the variation of the cost function when one moves to a neighboring solution. This will be detailed in the following two sections.

3 The dominance

The amplitude of variation of the cost function is the ratio C_{worst}/C_{opt}, with C_{worst} the cost of the worst solution, and C_{opt} the cost of the optimal one. As it is highly probable that computing such a quantity is NP-hard, we propose an alternative.

There is no need to distinguish between matrices F and D which play a symmetric role in the definition of the QAP, so we consider the *distance dominance*, noted $dd(D)$, defined like the flow dominance but for the matrix D. We have recently known that Talbi et al. use also this notion [6].

We define the *dominance* of an instance as the vector $(\min\{fd(F), dd(D)\}, \max\{fd(F), dd(D)\})$. In order to make comparisons, we introduce the partial order $<$ defined by $(u, v) < (u', v')$ if and only if $u < u'$ and $v < v'$.

Our interpretation is the following one. The dominance indicates the amplitude of variation of the cost function. When it is low, it means that all flows and distances are close to each other, and therefore one should not expect important difference between the cost of the optimal solution C_{opt} and the cost of the worst one C_{worst}. In contrary, when it is high, it means the presence of dominant patterns in matrice F and (or) D, and so it means that the difference between the optimal and the worst solution should be important.

Of course, cases when dominances cannot be compared are problematic. An alternative definition would be to set for the dominance the sum $fd(F) + dd(D)$. But in that case, the interpretation becomes difficult and rather approximative. It is certainly no true that an instance with for example $fd(F) = dd(D) = 50$ is equivalent to an instance with $fd(F) = 80$ and $dd(D) = 20$, even if their common dominance is 100.

Block in [7] proposes a normalization between 0 and 100 of the flow dominance, and call it the *complexity rating factor* C_f, by setting $C_f = 100\,(f_{UB} - fd(F))\,/\,(f_{UB} - f_{LB})$, with f_{UB} (respectively f_{LB}) the upper (respectively lower) bound for the flow dominance. Generalizing this idea, it is possible to define a normalized dominance by $(\min\{C_f, C_d\}, \max\{C_f, C_d\})$. Of course the upper and lower bounds, f_{UB} and f_{LB} remain to be calculated. Notice that we have $f_{UB} = d_{UB}$ and $f_{LB} = d_{LB}$ as we impose the same constraints on the matrices F and D (they are symmetric with a null diagonal). We think, as previous authors (see [11]), that the upper bound is obtained when all elements of the symmetric flow matrix are zero except two, i.e. for example $f_{12} = f_{21} = \alpha$ and $f_{ij} = 0$ otherwise, and that the lower bound is obtained when all the elements of the flow matrix are equal, except those on the diagonal which are null, i.e. $f_{ij} = \alpha$ for $i \neq j$ and $f_{ii} = 0$ otherwise. One obtains $f_{UB} = (100/\sqrt{2})\,n\,\sqrt{(n^2-2)/(n^2-1)} \simeq 70\,n$, and $f_{LB} = 100\,n/((n-1)\sqrt{n+1}) \simeq 100/\sqrt{n}$, independently of the value of α.

4 The ruggedness coefficient

We have seen it was necessary to distinguish between the global amplitude of variation of the cost function, and the shape of these variations when one moves to a neighboring solution. To take into account this second point, we propose to use the concept of ruggedness of a landscape. The union of the cost function and the neighborhood structure forms what is called a landscape. Intuitively, a rugged landscape means a lot of narrow local optima, and so it is a difficult task for a local search algorithm to explore it.

This link between the ruggedness of a landscape and its fitness for local search was suggested by

Weinberger [23], who the first proposed a mathematical definition of the notion of ruggedness by defining autocorrelation functions, and by Stadler who considerably developed the theory of landscapes [19]. In [2] and [3] we have introduced a straightforwardly derived general parameter, called the *autocorrelation coefficient* ξ. For the QAP we normalized it to range between 0 and 100, and named it the *ruggedness coefficient* ζ [4].

In [4] we have proved that the autocorrelation coefficient ξ was sharply minored by $n/4$, and we conjectured that it was majored by $n/2$. In fact, using numerical computations and some calculations we think it is possible to enhance this bound, and we propose definitely $(n-1)/2$ as a sharp upper bound. Hence, we define our ruggedness coefficient by setting $\zeta = 100 - \frac{400}{n-2}(\xi - \frac{n}{4})$.

By definition

$$\begin{aligned}\xi &= \frac{\langle (C(\pi) - C(\tilde{\pi}))^2 \rangle}{\langle (C(\pi) - C(\pi'))^2 \rangle} \\ &= \frac{2\,(\langle C^2 \rangle - \langle C \rangle^2)}{\langle (C(\pi) - C(\pi'))^2 \rangle},\end{aligned}$$

with $\langle (C(\pi) - C(\pi'))^2 \rangle$ the average value of $(C(\pi) - C(\pi'))^2$ over all neighboring permutations π and π', $\langle (C(\pi) - C(\tilde{\pi}))^2 \rangle$ the average value of the same quantity but this time over any two permutations π and $\tilde{\pi}$, $\langle C \rangle$ the average cost over all permutations and $\langle C^2 \rangle$ the average squared cost.

A value close to 0 for the ruggedness coefficient ζ means that ξ has a high value (close to its upper bound), and it means that the average squared difference cost between two neighboring permutations is small relative to the average squared difference cost between any two permutations. If so, the landscape of the instance is considered rather flat and therefore well suited for local search based heuristics. In contrary, a value close to 100 for the ruggedness coefficient means a very steep landscape with a lot of local optima, and therefore a landscape which is bad suited for local search.

Notice that taking only into account the quantity $\langle (C(\pi) - C(\pi'))^2 \rangle$ would be improper for measuring the ruggedness of the landscape. What is important is not the amplitude of variations between neighboring solutions, but the shape of these variations. It is a desirable property, since optimizing C or αC with $\alpha \in \mathbb{R}$ makes no difference for meta-heuristics, and indeed the ruggedness coefficient is invariant under such a multiplicative factor.

In [4] we have proved the following proposition.

Proposition 1 *For the Quadratic Assignment Problem, let* $F_k = \sum_{i,j} f_{ij}^k$, f_i *for the sum* $\sum_j f_{ij}$, D_k *for the sum* $\sum_{i,j} d_{ij}^k$, *and* d_i *for the sum* $\sum_j d_{ij}$. *Then, the variance of the cost function and the average squared cost difference between two neighboring solutions, for the 2-exchange neighborhood, are given by:*

$$\begin{aligned} Var(C) = \; & \{F_2\, D_2\, n^4 - 2\,(F_2\,(2\,D_2 + \sum_i d_i^2) + \sum_i f_i^2\,(D_2 - \sum_i d_i^2))\, n^3 + \\ & (F_1^2\,(D_2 - \; 2\sum_i d_i^2) + F_2\,(D_1^2 \; + \; 5\,D_2 + 4\,\sum_i d_i^2) - 2\sum_i f_i^2(D_1^2 - 2\,D_2))\, n^2 + \\ & (F_1^2\,(2D_1^2 - D_2 + 2\sum_i d_i^2) - F_2\,(D_1^2 + 2\,(D_2 + \sum_i d_i^2)) + \\ & 2\sum_i f_i^2\,(D_1^2 - D_2 - \sum_i d_i^2))\; n - 3\, F_1^2\, D_1^2\} \;/\; (2n^2\,(n-1)^2\,(n-2)\,(n-3)), \end{aligned}$$

and

$$\begin{aligned}\langle (C(\pi) - C(\pi'))^2 \rangle \;=\; & 4\,(F_2\, D_2\, n^3 - (2\, F_2\, (2\, D_2 + \sum_i d_i^2) + \sum_i f_i^2 (2\, D_2 - \sum_i d_i^2))\, n^2 + \\ & (F_1^2\, (D_2 - \sum_i d_i^2) + F_2\, (D_1^2 + 5\, D_2 + 4 \sum_i d_i^2) - \\ & \sum_i f_i^2 (D_1^2 - 4\, D_2 - 3 \sum_i d_i^2))\, n + \\ & F_1^2\, (D_1^2 - D_2 - \sum_i d_i^2) - (D_1^2 + 2\, (D_2 + \sum_i d_i^2))\, (F_2 + \sum_i f_i^2)) \\ & /\; (n^2\, (n-1)^2\, (n-2)\, (n-3)).\end{aligned}$$

Since we have $\xi = \frac{2\,Var(C)}{\langle (C(\pi) - C(\pi'))^2 \rangle}$, it allows us to exactly calculate the ruggedness coefficient in polynomial time.

5 A new complexity measure

We think that our ruggedness coefficient in conjunction with the dominance parameter provide important features of the search space explored during a local search algorithm, and using them together give a rather precise idea of the complexity of an instance for these heuristics.

The ruggedness coefficient is *a priori* independent of the dominance. However we have observed that in practice it was more and more difficult, because of their rarity, to generate instances with a high dominance when the autocorrelation coefficient decreases.

The way the dominance and the ruggedness coefficient can be combined to obtain a complexity measure for the QAP is resumed in the figure 1.

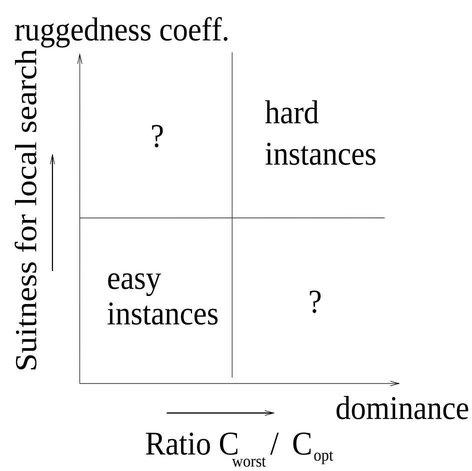

Figure 1: *classification of the QAP instances*

When both the dominance and the ruggedness coefficient have a low value, instances can be considered as very easy ones for local search. Indeed, it means that the cost of any solution cannot be very far away from the optimum cost, and moreover local search is well suited for these instances. In contrary, when both the dominance and the ruggedness coefficient have a high value, instances can be considered as very hard ones for local search. In other cases when the dominance is high and the ruggedness coefficient is low or the dominance is low and the ruggedness coefficient is high, the situation is no more so simple and one cannot conclude so easily. The computational results presented in the next section confirm the validity of such a classification.

Notice however, that because of the structure of the matrices which have a maximum dominance, theses conclusions applies up to a certain level. Indeed, according the section 3 if the two matrices F and D are such that all their elements is null except two, then the dominance is maximal, and nevertheless the instance is trivial.

We have also noticed that contrary to the dominance, it appears that visual inspection is not able to determine if the instance has a low or high ruggedness coefficient.

6 Computational results

We have chosen to generate QAP instances with known optimal solution. We have used a slightly modified version of the test problems generator GEN2 proposed by Li and Pardalos in [13] to ensure that both matrices F and D are symmetric.

Algorithm 1: *The symmetric test problems generator*

- Construct matrix F by setting $f_{ij} = 100$ if $i \neq j$, and $f_{ij} = 0$ otherwise. For $i < j$, generate randomly d_{ij} with an appropriate distribution (to be explained later: procedure 1), and set $d_{ji} = d_{ij}$, $d_{ii} = 0$.
- Sort d_{ij}, for $i < j$, ascendingly and store the rank of d_{ij} in r_{ij} in increasing order.
- Randomly generate integers x_i, $1 \leq i \leq n(n-1)/2$, with a uniform distribution in [0,100], and sort them ascendingly.
- For each f_{ij}, $i < j$, set $f_{ij} = f_{ij} - x_{r_{ij}}$ and $f_{ji} = f_{ij}$.

The identity permutation is the global optimum for this generated symmetric QAP.

For the construction of the matrix D, we apply the following procedure.

Procedure 1: *The generation of locations for the matrix D*
Let p be a fixed real between 0 and 1, then for each $i < j$ do:

- Choose a real r randomly between 0 and 1.
- If $r < p$,

  then $d_{ij} = d_{ji}$ is affected a random integer number between 0 and 100,

  else set $d_{ij} = d_{ji} = 0$.

The advantage of this method is that we can generate instances with high distance dominance, but the values of the ruggedness coefficient is almost always up than 80. We have found experimentally that for instances of size 20, by setting $p = 0.75$ (respectively $p = 0.5$ and $p = 0.3$)

ζ	l=5		l=10	
	% rel. error	nbr steps	% rel. error	nbr steps
$80 \leq \zeta < 82$	0.8	17072	0.1	29853
$82 \leq \zeta < 84$	0.7	16836	0.1	29910
$84 \leq \zeta < 86$	1.1	17129	0.1	29946
$86 \leq \zeta < 88$	0.9	17387	0.1	30210
$88 \leq \zeta < 90$	1.7	18306	0.0	31398
$90 \leq \zeta < 92$	2.1	18089	0.2	31567
$92 \leq \zeta < 94$	2.2	18610	0.3	32270
$94 \leq \zeta < 96$	3.9	19654	0.8	33907
$96 \leq \zeta < 98$	6.9	21134	1.7	36121
$98 \leq \zeta < 100$	11.1	22392	3.7	38369

Table 1: Performance of simulated annealing on various instances of size 20, with dominance $(67 \pm 3, 94 \pm 3)$.

we were able to generate instances with a dominance lying in $(67 \pm 3, 94 \pm 3)$ (respectively $(67 \pm 3, 135 \pm 3)$ and $(58 \pm 3, 193 \pm 3)$).

For our local search based heuristic, we have chosen the simulated annealing implementation of Johnson et al. [12]. Its robustness allows us to avoid the problem of tuning numerous parameters, otherwise some instances would have been penalized and others favorised.

At each step two plants i and j are chosen at random, and the change δ_{ij} in the cost function, of swapping them is computed (in linear time). The swap is accepted if, either $\delta_{ij} \leq 0$, or $X \leq e^{-\delta_{ij}/T}$ with X a random real number drawn from the uniform [0,1] distribution, and T is a parameter, called the temperature, which decreases every a fixed number (called the temperature length L) of steps in a geometric way, i.e. $T \leftarrow r\,T$, with r the geometric cooling ratio.

The initial temperature is experimentally fixed by a "trial run" approach in such a way that the fraction of accepted moves is between 38% and 42% during the first L steps. The temperature length L is set to be $n(n-1)l/2$, that is $l \times$ neighborhood size, with $l = 5$ and $l = 10$, and the geometric cooling ratio is 0.95. When at the end of a temperature the percentage of accepted moves is less than 2%, it means that the search is going to stop soon, because none moves will be accepted. If a such observation occurs five times, then we consider the search process as being "frozen", and the simulated annealing stops. There is an exception if a solution better than the previous best one is found, in that case we wait again for five new low-acceptance temperature completions, to stop the algorithm.

The final result is the best solution found during the entire search. For each instance, we have performed 10 trials, starting from random permutations, and for each category of problems we have generated 100 instances. The results are reported in tables 1,2,3.

We were unable to generate instances with a ruggedness coefficient below 80 with this method and the procedure 1. In [4] using another procedure we have obtained instances with a complete range of value for their ruggedness coefficient, but with a lower dominance than here.

It is worth to go back to the classification on the figure 1. For example, especially hard instances can be found in the table 3 (high dominance), with a ruggedness coefficient over 90. Notice, that both relative error and number of steps are clearly increasing with regard to the ruggedness

ζ	l=5		l=10	
	% rel. error	nbr steps	% rel. error	nbr steps
$80 \leq \zeta < 82$	4.3	18843	0.4	33430
$82 \leq \zeta < 84$	5.9	19478	1.0	33647
$84 \leq \zeta < 86$	7.3	19572	1.0	33581
$86 \leq \zeta < 88$	9.6	20218	1.8	34759
$88 \leq \zeta < 90$	9.1	20224	1.7	34831
$90 \leq \zeta < 92$	12.2	20916	2.5	35591
$92 \leq \zeta < 94$	16.2	21510	5.0	37116
$94 \leq \zeta < 96$	19.6	22430	6.3	38846
$96 \leq \zeta < 98$	28.5	23601	10.6	40979
$98 \leq \zeta < 100$	40.3	24968	18.9	44184

Table 2: Performance of simulated annealing on various instances of size 20, with dominance $(67 \pm 3, 135 \pm 3)$.

ζ	l=5		l=10	
	% rel. error	nbr steps	% rel. error	nbr steps
$80 \leq \zeta < 82$	22.1	21566	6.6	37588
$82 \leq \zeta < 84$	26.3	21693	6.7	37774
$84 \leq \zeta < 86$	26.3	21729	10.0	37973
$86 \leq \zeta < 88$	28.0	21903	12.8	38682
$88 \leq \zeta < 90$	32.1	22355	12.3	39355
$90 \leq \zeta < 92$	37.3	22962	18.2	40633
$92 \leq \zeta < 94$	49.4	24026	21.6	41378
$94 \leq \zeta < 96$	57.3	24323	27.1	43001
$96 \leq \zeta < 98$	70.0	25312	40.3	45245
$98 \leq \zeta < 100$	89.3	26522	58.1	48813

Table 3: Performance of simulated annealing on various instances of size 20, with dominance $(58 \pm 3, 193 \pm 3)$).

coefficient and this for both cases with $l = 5$ and $l = 10$. In contrary especially easy instances can be found in the table 1 (low dominance), with a ruggedness below 90. Notice also that it is not contradictory that simulated annealing when applied to a difficult instance in the sense of a high ratio C_{worst}/C_{opt}) even very well suited for local search algorithms (first lines of table 3), gives less good results than when it is applied on an easy instance in the sense of a low ratio C_{worst}/C_{opt} even bad suited for local search algorithms (last lines of table 1).

We assume we can order the dominances in the following way: $(67, 94) < (67, 135) < (58, 193)$, authorizing a certain degree of freedom in comparison with the formal definition of section 3. Then, it appears that the dominance is the determining parameter when measuring the difficulty of an instance. For example all instances in table 3 which all have a dominance around $(58, 193)$ are more difficult than all instances in table 1 which all have a dominance around $(67, 94)$, whatever the ruggedness coefficient. The difficulty of instances in table 2, which all have a dominance around $(67, 135)$, is logically bounded by those in tables 1 and 3.

But, the importance of the ruggedness coefficient is clearly demonstrated. When the dominance is fixed, the performance of simulated annealing is directly function of the ruggedness coefficient. The relative error monotonically increases in a very important way, when the landscape becomes more and more rugged. Notice that in table 3 when the temperature length is set to be $l \times$ instance size, with $l = 10$, the relative error is multiplied by a factor almost nine when the ruggedness coefficient passes from 80 to 100. This degradation of results cannot be imputed on a less number of iterations, indeed notice that this number increases along with the ruggedness coefficient, and so these results are effectively an evidence of the well (respectively bad) suitness of a flat (respectively rugged) landscape for simulated annealing.

The link between the performance of simulated annealing and the ruggedness coefficient is almost always true. There are few exceptions, but they should not make forget the remarkable concordance of results. It is intuitively clear that the more rugged a landscape is and the more difficult will be simulated annealing on it, but we do not have a theorem which states a precise law concerning the performance of simulated annealing with the ruggedness coefficient. These results suggest that we can go perhaps further than the classification of figure 1.

7 Conclusion

The ruggedness coefficient together with the dominance seem to give a powerful complexity measure for characterizing the difficulty of QAP instances. However, one should notice that this classification becomes less and less true as elaborate heuristics are employed such as hybrid methods, or when they are moving away from basic local search, for example genetic algorithms or tabu search with powerful diversification mechanisms. In that case, they are more adapted to explore rugged landscapes than basic simulated annealing.

Finally, the framework of our research could be resumed by the following quotation from the conclusion of [16] where future research lines are mentioned: "[...] a detailed analysis of the search space will certainly help to understand when the algorithm performs well and when not. Gaining more insight into the problem structure may lead to an algorithm that performs well even for instances that are considered to be hard to solve." We have used two theoretical parameters: the dominance and the ruggedness coefficient, to obtain some features of the landscape of QAP instances, but analyzing landscapes in a more empirical way is also possible [9] [14] [18]. Also, others characteristics of landscapes could be considered, for example: are all local optima uniformly dispersed, or concentrated in a small region? Gathering knowledge about a landscape could help to refine existing meta-heuristics well adapted for a certain types of instance or problems, or give ideas for developing new methods, enhancing the number of tractable instances.

References

[1] E. Angel and V. Zissimopoulos. On the quality of local search for the quadratic assignment problem. to appear in Discrete Applied Mathematics.

[2] E. Angel and V. Zissimopoulos. Towards a classification of combinatorial optimization problems relatively to their difficulty for generalized local search algorithms. Technical Report 1112, LRI, Université de Paris-Sud, 1997.

[3] E. Angel and V. Zissimopoulos. Autocorrelation coefficient for the graph bipartitioning problem. *Theoretical Computer Science*, 191:229–243, 1998.

[4] E. Angel and V. Zissimopoulos. On the landscape ruggedness of the quadratic assignment problem. Technical Report 98-02, LIPN, Institut Galilée, Université de Paris-Nord, 1998.

[5] V. Bachelet, P. Preux, and E-G. Talbi. Parallel hybrid meta-heuristics: Application to the Quadratic Assignment Problem. Technical Report LIL-96-2, LIFL, Université des Sciences et Technologies de Lille, 59655 Villeneuve d'Ascq, 1996. Proc. Parallel Optimization Colloquium, Versailles.

[6] V. Bachelet, P. Preux, and E-G. Talbi. The landscape of the Quadratic Assignment Problem and local search methods. In *Tenth Meeting of the European Chapter on Combinatorial Optimization, Teneriffe, Canary Islands, May*, 1997.

[7] T.E. Block. On the complexity of facilities layout problems. *Management Science*, 25:280, 1979.

[8] D.T. Connolly. An improved annealing scheme for the qap. *European Journal of Operational Research*, 46:93–100, 1990.

[9] C. Fonlupt, D. Robilliard, P. Preux, and E-G. Talbi. Fitness landscapes and performance of meta-heuristics. In *2nd International Conference on Metaheuristics, Sophia-Antipolis, France*, July 21–24 1997.

[10] M.R. Garey and D.S. Johnson. *Computers and intractability - A guide to the theory of NP-completeness.* W.H. Freeman and Company, 1979.

[11] W. Herroelen and A. Van Gils. On the use of flow dominance in complexity measures for facility layout problems. *International Journal of Production Research*, 23(1):97–108, 1985.

[12] D.S. Johnson, C.R. Aragon, L.A. McGeoch, and C. Schevon. Optimization by simulated annealing: an experimental evaluation; part I, graph partitioning. *Operations Research*, 37(6):865–892, 1989.

[13] Y. Li and P.M. Pardalos. Generating quadratic assignment test problems with known optimal permutations. *Computational Optimization and Applications*, 1:163–184, 1992.

[14] O. Martin. Propriétés des solutions engendrées par des algorithmes heuristiques: la limite de grande taille. In *Premier Congrès de la Société Française de Recherche Opérationnelle et Aide à la Décision, 14-16 Janvier, Ministère de la Recherche, Carré des Sciences, Paris*, 1998.

[15] T. Mautor and C. Roucairol. Difficulties of exact methods for solving the quadratic assignment problem. In *DIMACS (Series in Discrete Mathematics and Theoretical Computer Science) Workshop*, volume 16, pages 263–274, May 20–21 1993.

[16] P. Merz and B. Freisleben. A genetic local search approach to the quadratic assignment problem. In *Proceedings of the Seventh International Conference on Genetic Algorithms (ICGA'97), East Lansing, USA*, 1997.

[17] S. Sahni and T. Gonzalez. P-complete approximation problems. *Journal of the ACM*, 23:555–565, 1976.

[18] G.R. Schreiber and O. Martin. Cut size statistics of graph bisection heuristics. To appear in SIAM Journal on Optimization.

[19] Peter F. Stadler. Landscapes and their correlations functions. Technical Report 95-07-067, Santa Fe Institute, Santa Fe, NM, 1995.

[20] E.D. Taillard. Comparison of iterative searches for the quadratic assignment problem. *Location Science*, 3:87–105, 1995.

[21] M. Dorigo V. Maniezzo and A. Colorni. Algodesk: An experimental comparison of eight evolutionary heuristics for the quadratic assignment problem. *European Journal of Operational Research*, 81:188–204, 1995.

[22] T.E Vollmann and E.S. Buffa. The facilities layout problem in perspective. *Management Science*, 12(10):450–468, 1966.

[23] E.D. Weinberger. Correlated and uncorrelated fitness landscapes and how to tell the difference. *Biological Cybernetics*, 63:325–336, 1990.