Journal of Heuristics, 8: 399–414, 2002

On the Hardness of the Quadratic Assignment Problem with Metaheuristics

ERIC ANGEL
Université d'Évry-Val d'Essonne, LaMI, CNRS-UMR 8042, 91025 Evry, France
email: angel@lami.univ-evry.fr

VASSILIS ZISSIMOPOULOS
Université Paris Nord, LIPN, CNRS-URA 1507, 93430 Villetaneuse, France
email: 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 metaheuristics 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 and taboo search are presented.

Key Words: 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 (Garey and Johnson, 1979), and non approximable (Sahni and Gonzalez, 1976). 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. In the sequel we shall assume 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 Dorigo, Maniezzo, and Colorni (1995): "[...] 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 Connolly (1990): "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 Angel and Zissimopoulos (1998) the authors have presented a theoretical performance guarantee result for a basic local search procedure, nonetheless the quality of solutions obtained with metaheuristics 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 one tests it, it is important to ensure that instances that are generated are well distributed in the "complexity space" of the problem. One 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 (1966). It measures to what extent the flow matrix F shows "dominant" flow patterns. A possible definition of the flow dominance is given by Merz and Freisleben (1997) and Herroelen and Van Gils (1985):

$$fd(F) = 100\frac{\sigma}{\mu}, \quad \text{with}$$

$$\mu = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} f_{ij}, \quad \text{the mean and}$$

$$\sigma = \sqrt{\frac{1}{n^2-1}\sum_{i=1}^{n}\sum_{j=1}^{n}(f_{ij}-\mu)^2},$$

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 (Herroelen and Van Gils, 1985). Yet, the flow dominance has some drawbacks too. First, different definitions of this parameter have been proposed since its introduction (Herroelen and Van Gils, 1985). 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 Mautor and Roucairol (1993) 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 Bachelet, Preux, and Talbi (1996), Taillard (1995) and Merz and Freisleben (1997), but the situation is more involved.

Bachelet, Preux, and Talbi have observed in their parallel hybrid metaheuristics combining tabu search and genetic algorithms (Bachelet, Preux, and Talbi, 1996), 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 (Taillard, 1995), 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 combining genetic algorithms with local search (Merz and Freisleben, 1997) 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 from a solution to a neighboring one. 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. Burkard and Fincke have shown that for infinitely large instances generated in a randomly uniform way this ratio tends to 1 (Burkard and Fincke, 1985). We shall comment this result later in Section 5. 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 (Bachelet, Preux, and Talbi, 1997).

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 (1979) 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 Herroelen and Van Gils, 1985), 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 (1990), 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 (Stadler, 1995; Reidys and Stadler, to appear). In Angel and Zissimopoulos (1998, 2000) 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 (Angel and Zissimopoulos, 2001).

In Angel and Zissimopoulos (2001) 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 metaheuristics, and indeed the ruggedness coefficient is invariant under such a multiplicative factor.

In Angel and Zissimopoulos (2001) 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*:

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

and

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

Since we have $\xi = \frac{2\,Var(C)}{\langle (C(\pi)-C(\pi'))^2 \rangle}$, Proposition 1 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 provides 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.

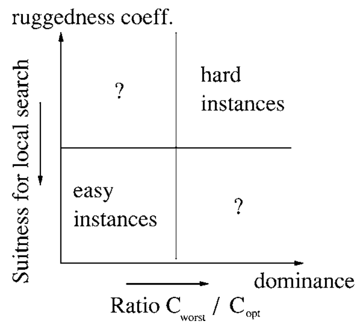

Figure 1. Classification of the QAP instances for metaheuristics.

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.

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 are null except two, then the dominance is maximal, and nevertheless the instance is trivial.

Recall also that the result of Burkard and Fincke (1985) mentioned in Section 3, i.e. $\lim_{n\to\infty} C_{worst}/C_{opt} = 1$, implies that large instances randomly generated in an uniform way lie on the left in figure 1 and are rather easy to solve according our classification.

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

The aim of this section is to provide some experimental evidences that the classification proposed is pertinent. It is by no means a performance comparison of various existing metaheuristics for the quadratic assignment problem.

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 (1992) 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 $p = 0.3$) 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 local search based metaheuristics, we have considered the simulated annealing implementations of Johnson et al. (1989) and Connolly (1990) (their robustness allows us to avoid the problem of tuning numerous parameters, otherwise some instances would have been penalized and others favorised) and the robust taboo search of Taillard (1991).

We begin by a description of the simulated annealing algorithm of Johnson et al. At each iteration 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. $T \leftarrow rT$, 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 iterations. 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 (not necessarily consecutive), 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 category of problems we have generated 100 instances, and for each instance we have performed 10 trials, starting from random permutations. The results (the average relative error and the average numbers of steps over all runs and instances for each category) are reported in Tables 1–3.

We were unable to generate instances with a ruggedness coefficient below 80 with this method and the procedure 1. In Angel and Zissimopoulos (2001) 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 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

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

	l = 5		l = 10	
ζ	% Rel. error	No. of iterations	% Rel. error	No. of iterations
$80 \leq \zeta < 82$	0.4	14312	0.0	25978
$82 \leq \zeta < 84$	0.3	14605	0.1	26240
$84 \leq \zeta < 86$	0.7	14622	0.0	25800
$86 \leq \zeta < 88$	1.2	15007	0.0	26617
$88 \leq \zeta < 90$	1.4	15467	0.2	27067
$90 \leq \zeta < 92$	2.0	15735	0.3	27863
$92 \leq \zeta < 94$	3.3	16525	0.2	28511
$94 \leq \zeta < 96$	3.6	17022	0.8	29947
$96 \leq \zeta < 98$	6.7	18521	1.4	31673
$98 \leq \zeta < 100$	10.1	19661	3.0	34707

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

	l = 5		l = 10	
ζ	% Rel. error	No. of iterations	% Rel. error	No. of iterations
$80 \leq \zeta < 82$	4.0	16065	0.2	28405
$82 \leq \zeta < 84$	5.3	16351	0.7	28627
$84 \leq \zeta < 86$	6.9	16820	0.9	29558
$86 \leq \zeta < 88$	9.0	17141	1.5	29839
$88 \leq \zeta < 90$	8.2	17006	1.2	30381
$90 \leq \zeta < 92$	12.2	17848	2.0	30920
$92 \leq \zeta < 94$	14.7	18410	3.7	32260
$94 \leq \zeta < 96$	21.5	19235	7.3	33829
$96 \leq \zeta < 98$	30.8	20445	11.7	36291
$98 \leq \zeta < 100$	43.8	22135	21.0	39683

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

	l = 5		l = 10	
ζ	% Rel. error	No. of iterations	% Rel. error	No. of iterations
$80 \leq \zeta < 82$	25.4	18532	6.3	33008
$82 \leq \zeta < 84$	27.2	18545	9.3	33470
$84 \leq \zeta < 86$	37.4	19160	13.7	33761
$86 \leq \zeta < 88$	36.3	18938	14.3	34015
$88 \leq \zeta < 90$	52.7	20269	13.0	34340
$90 \leq \zeta < 92$	54.1	20174	21.3	35406
$92 \leq \zeta < 94$	65.0	20808	27.1	36594
$94 \leq \zeta < 96$	77.6	21669	33.4	38438
$96 \leq \zeta < 98$	105.6	22732	55.1	41306
$98 \leq \zeta < 100$	125.7	23634	77.3	43667

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.

In the simulated annealing implementation of Connolly (1990), $M/100$ random swaps are performed in a preprocessing phase, with M the total number of iterations of the algorithm. During this phase $\delta_{\max}$(respectively $\delta_{\min}$) the maximum (respectively minimum) positive cost changes are computed. The temperature length is 1, and the temperature T_{i+1} at the $i + 1$-th iteration is $T_{i+1} = T_i/(1 + \beta T_i)$, where $\beta \ll T_0$ and $\beta = (T_0 - T_f)/MT_0T_f$. Such a value for β ensures that starting at the initial temperature T_0 one reaches the final temperature T_f after M iterations. Temperatures T_0 and T_f are fixed by $T_0 = \delta_{\min} + \frac{1}{10}(\delta_{\max} - \delta_{\min})$ and $T_f = \delta_{\min}$. Now if MXFAIL consecutive uphill steps are rejected the next uphill is accepted and the temperature is fixed to the temperature at which the current best solution was found. The optimal value of the temperature is found, then the cooling is stopped by setting $\beta = 0$ and the algorithm continues until M iterations are performed. Connolly has observed that when the temperature was fixed at an "optimal value" the simulated annealing performed remarkably well, and the aim of this scheme is to find such an optimal temperature. The default value of MXFAIL is the size of the neighborhood, i.e. $n(n - 1)/2$.

We have used the source code (in C++) available at the Taillard Web page which implements the simulated annealing of Connolly with the slight modification that at the beginning 1000 random swaps are used instead of the proposed $M/100$. The results with $M = 25000$ and $M = 50000$ (these numbers are roughly the average number of iterations performed in the Johnson simulated annealing in Tables 1–3) are presented in Table 4 . One observes few anomalies in the progression of difficulty when the ruggedness coefficient increases (especially for instances with a high dominance), but we can draw the same conclusions than for the Johnson implementation of simulated annealing.

We briefly describe now the features of the robust taboo search proposed by Taillard (1991) and whose source code is available (in Pascal) at the Taillard Web page. The taboo list stores for each plant and location the latest iteration at which the plan occupied that location. The length s of the taboo list randomly changes between $s_{\min}$ and $s_{\max}$, with $s_{\min} = \lfloor 0.9\, n \rfloor$ and $s_{\max} = \lceil 1.1\, n \rceil$. This change occurs every $2s_{\max}$ iterations. The classical aspiration criterion is used. It allows a taboo move to be performed if the move leads to a

Table 4. Performance of simulated annealing (Connolly) on various instances of size 20 with (a) dominance in $(67 \pm 3, 94 \pm 3)$, (b) dominance in $(67 \pm 3, 135 \pm 3)$ and (c) dominance in $(58 \pm 3, 193 \pm 3)$.

	25000 iterations			50000 iterations		
ζ	*a*	*b*	*c*	*a*	*b*	*c*
$80 \leq \zeta < 82$	8.0	25.3	133.4	3.4	14.0	108.6
$82 \leq \zeta < 84$	7.6	30.3	139.1	3.9	12.4	103.7
$84 \leq \zeta < 86$	8.2	30.4	133.7	3.7	17.1	99.7
$86 \leq \zeta < 88$	10.0	34.4	145.3	4.3	16.7	113.5
$88 \leq \zeta < 90$	9.9	35.6	146.9	4.1	17.2	109.6
$90 \leq \zeta < 92$	12.4	34.9	153.2	5.4	18.1	106.7
$92 \leq \zeta < 94$	14.0	42.3	144.2	6.6	22.4	105.3
$94 \leq \zeta < 96$	15.8	48.1	142.8	7.3	25.7	102.6
$96 \leq \zeta < 98$	17.8	55.7	165.2	9.8	31.4	120.9
$98 \leq \zeta < 100$	22.4	63.5	166.2	11.6	37.1	117.1

solution better than the best one found so far. A long term memory is also used to diversify the search process. A move will be automatically performed if it places both plants at locations they have not occupied during the last t iterations, whatever the cost of the new solution is. In the code source the value of t is $3n^2$.

We have considered runs consisting of 150 and 300 steps. Recall that in the taboo search the entire neighborhood is examined at each step, which gives respectively 28500 and 57000 swaps evaluations. The results are reported in Table 5.

Table 5. Performance of robust taboo search on various instances of size 20 with (a) dominance in $(67 \pm 3, 94 \pm 3)$, (b) dominance in $(67 \pm 3, 135 \pm 3)$ and (c) dominance in $(58 \pm 3, 193 \pm 3)$.

	150 steps (28500 iterations)			300 steps (57000 iterations)		
ζ	*a*	*b*	*c*	*a*	*b*	*c*
$80 \leq \zeta < 82$	4.5	19.6	108.0	0.9	7.3	67.2
$82 \leq \zeta < 84$	4.9	23.7	94.9	0.7	6.4	60.5
$84 \leq \zeta < 86$	6.0	22.5	100.8	1.2	8.9	63.3
$86 \leq \zeta < 88$	6.1	28.1	105.2	1.0	11.0	61.6
$88 \leq \zeta < 90$	8.4	29.2	105.7	1.8	10.2	66.8
$90 \leq \zeta < 92$	9.0	28.3	113.5	2.6	9.0	68.1
$92 \leq \zeta < 94$	10.9	37.4	122.2	3.3	15.7	75.1
$94 \leq \zeta < 96$	14.9	38.7	118.6	5.3	15.5	70.1
$96 \leq \zeta < 98$	16.9	47.9	132.1	7.3	23.8	85.7
$98 \leq \zeta < 100$	19.0	55.0	143.7	8.8	30.6	95.2

Table 6. Symmetric instances of size 20 of the QAPLIB.

			Johnson et al.		Connolly		Taillard	
Name	Dominance	ζ	% Error	No. of iterations	25000	50000	150	300
had20	(46.0, 64.3)	81.6	0.2	112290	0.2	0.2	0.4	0.4
rou20	(64.4, 65.7)	92.1	1.0	56050	1.3	1.1	0.9	0.7
tai20a	(64.9, 67.0)	95.3	1.9	55670	2.4	1.9	1.7	1.4
nug20	(54.2, 103.8)	82.2	0.5	70300	0.6	0.4	0.9	0.3
scr20	(54.2, 254.7)	88.6	1.2	87400	1.9	1.4	2.0	1.6
chr20c	(65.7, 346.4)	94.2	17.3	195890	60.3	44.3	69.0	57.3
chr20a	(59.5, 346.4)	97.7	14.2	96520	22.0	24.1	18.4	15.9
chr20b	(59.5, 346.4)	99.2	13.8	81130	17.9	15.1	14.0	12.8

As for the simulated annealing, the dominance is still the determining parameter to measure the difficulty of an instance. But when dominance is fixed, one observes that the ruggedness coefficient has a very important impact. The effect is less regular than in the simulated annealing case, but still widely observable.

We have also considered the symmetric instances of the QAPLIB (Burkard, Karisch, and Rendl, 1997), with a size 20 because we are able to define an order on them (see below). Each metaheuristic was executed 100 times on each instance. Results are reported in Table 6. We first give after the name of an instance, its dominance and the value of its ruggedness coefficient, then the relative error and the number of iterations executed with the simulated annealing of Johnson et al., then the relative error with the simulated annealing of Connolly when respectively 25000 and 50000 iterations are performed, and finally the relative error with the taboo search of Taillard when the number of steps is respectively 150 and 300 (which corresponds to respectively 28500 and 57000 swaps evaluations). The aim of the experiments was not to compare these metaheuristics but rather to determine if there is an observable trend supporting our complexity measure.

It is interesting to notice that each instance is always easy or difficult to solve in a simultaneous way for all these metaheuristics. Given two instances $\mathcal{I}_1$ and $\mathcal{I}_2$, we note $\mathcal{I}_1 < \mathcal{I}_2$ to mean that instance $\mathcal{I}_2$ has both a greater dominance and a greater ruggedness coefficient. The instances are sorted roughly in increasing difficulty according the partial order "$<$". Notice it is not possible to obtain a total order on the eight instaces, but we have $had20 < rou20 < tai20a$, $nug20 < scr20 < chr20c$ and $chr20a < chr20b$. Experimental results always confirm this classification, excepted for $chr20a$ and $chr20b$. Notice also that for all the other couples of comparable instances, such as $rou20$ with $chr20c$, experimental results are satisfactory. Moreover the instance $had20$ is comparable with all the other instances and it is the smallest one according the partial order "$<$". One should then expect that it is the easiest instance, and this is the case. Notice also that the most difficult instances for all methods are the three last in the table and effectively they have both a high dominance and ruggedness coefficient.

However, because of the limited number of symmetric instances, these results should be considered with care. We think that our classification scheme should be applied to clusters

Table 7. Nonsymmetric instances of size 20 of the QAPLIB.

			Johnson et al.		Connolly		Taillard	
Name	Dominance	ζ	% Error	No. of iterations	25000	50000	150	300
lipa20a	(39.8, 45.6)	97.4	0.7	43700	0.7	0.3	1.8	0.8
lipa20b	(45.6, 68.9)	96.1	0.0	30970	2.7	1.2	3.8	1.3
tai20b	(128.3, 333.2)	26.9	10.5	102600	10.9	17.8	17.7	17.6

of instances rather than isolated instances in order to compare the average performance of a local search metaheuristic. Indeed, our classification certainly cannot capture all the complexity inherent to the process of running a metaheuristic on an instance.

Results on nonsymmetric instances are reported in Table 7, however they raise several difficulties. A first one is that the expressions given in Section 4 to compute the autocorrelation coefficient ξ are no longer applicable on nonsymmetric instances. Thus the value we give for the ruggedness coefficient is only an approximate one, obtained by performing a random walk of length 10^6 on the landscape and collecting statistics. A second difficulty is that the bounds for the autocorrelation coefficient ξ are no longer valid, and therefore applying the same normalizing procedure as in the symmetric case does not lead to a ruggedness coefficient ζ lying between 0 and 100. So we cannot interpret theses results in the light of the ruggedness coefficient, nor include these instances with the symmetric ones. Notice however that the most easy (resp. hard) instance, lipa20a (resp. tai20b), has the lowest (resp. highest) dominance, suggesting (as it was noted before) that the dominance is the determining parameter, and the ruggedness coefficient acts as a modulation of it.

We have considered only instances of fixed size, because comparing instances of different sizes seems much more delicate. Our classification does not consider the computational effort devoted to each instance. In the simulated annealing of Connolly and tabu search of Taillard the number of iterations is fixed at the begining of the algorithm. Therefore to be fair, experiments should be performed with equal computational effort for each instance. Assuming that the computational effort is proportional to the number of iterations, should this number of iterations be proportional to the size n of the instance, or to the size $n!$ of the landscape? Therefore we think that our measure of complexity is a relative one, only exploitable if one considers a set of instances which have the same size.

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 could become 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 ants based systems.

Finally, the framework of our research could be resumed by the following quotation from the conclusion of Merz and Freisleben (1997) 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 (Fonlupt et al., 1997; Martin, 1998; Schreiber and Martin, 1999). 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 metaheuristics well adapted for a certain types of instance or problems, or give ideas for developing new methods, enhancing the number of tractable instances.

Acknowledgment

We thanks anonymous referees for their suggestions and useful remarks which contributed to improve the paper.

References

Angel, E. and V. Zissimopoulos. (1998). "Autocorrelation Coefficient for the Graph Bipartitioning Problem." *Theoretical Computer Science* 191, 229–243.

Angel, E. and V. Zissimopoulos. (1998). "On the Quality of Local Search for the Quadratic Assignment Problem." *Discrete Applied Mathematics* 82, 15–25.

Angel, E. and V. Zissimopoulos. (2000). "On the Classification of NP-Complete Problems in Terms of their Correlation Coefficient." *Discrete Applied Mathematics* 99, 261–277.

Angel, E. and V. Zissimopoulos. (2001). "On the Landscape Ruggedness of the Quadratic Assignment Problem." *Theoretical Computer Science* 263(1/2), 159–172.

Bachelet, V., P. Preux, and E-G. Talbi. (1996). "Parallel Hybrid Meta-Heuristics: Application to the Quadratic Assignment Problem." Technical Report LIL-96-2, LIFL, Université des Sciences et Technologies de Lille.

Bachelet, V., P. Preux, and E-G. Talbi. (1997). "The Landscape of the Quadratic Assignment Problem and Local Search Methods." In *Tenth Meeting of the European Chapter on Combinatorial Optimization.* Teneriffe, Canary Islands.

Block, T.E. (1979). "On the Complexity of Facilities Layout Problems." *Management Science* 25, 280.

Burkard, R.E. and U. Fincke. (1985). "Probabilistic Asymptotic Properties of Some Combinatorial Optimization Problems." *Discrete Applied Mathematics* 12, 21–29.

Burkard, R.E., S.E. Karisch, and F. Rendl. (1997). "QAPLIB—A Quadratic Assignment Problem Library." *Journal of Global Optimization* 10, 391–403. See also at http://www.imm.dtu.dk/~sk/qaplib/.

Connolly, D.T. (1990). "An Improved Annealing Scheme for the QAP." *European Journal of Operational Research* 46, 93–100.

Dorigo, M., V. Maniezzo, and A. Colorni. (1995). Algodesk: An Experimental Comparison of Eight Evolutionary Heuristics for the Quadratic Assignment Problem." *European Journal of Operational Research* 81, 188–204.

Fonlupt, C., D. Robilliard, P. Preux, and E-G. Talbi. (1997). "Fitness Landscapes and Performance of Meta-Heuristics." In *2nd International Conference on Metaheuristics*. Sophia-Antipolis, France.

Garey, M.R. and D.S. Johnson. (1979). *Computers and Intractability—A Guide to the Theory of NP-Completeness*. W.H. Freeman and Company.

Herroelen, W. and A. Van Gils. (1985). "On the Use of Flow Dominance in Complexity Measures for Facility Layout Problems." *International Journal of Production Research* 23(1), 97–108.

Johnson, D.S., C.R. Aragon, L.A. McGeoch, and C. Schevon. (1989). "Optimization by Simulated Annealing: An Experimental Evaluation; part I, Graph Partitioning." *Operations Research* 37(6), 865–892.

Li, Y. and P.M. Pardalos. (1992). "Generating Quadratic Assignment Test Problems with Known Optimal Permutations." *Computational Optimization and Applications* 1, 163–184.

Martin O. (1998). "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*. Paris, Janvier.

Mautor, T. and C. Roucairol. (1993). "Difficulties of Exact Methods for Solving the Quadratic Assignment Problem." In *DIMACS (Series in Discrete Mathematics and Theoretical Computer Science) Workshop*, Vol. 16, pp. 263–274.

Merz, P. and B. Freisleben. (1997). "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.

Reidys, C.M. and P.F. Stadler. "Combinatorial Landscapes." *SIAM Review*, to appear.

Sahni, S. and T. Gonzalez. (1976). "P-Complete Approximation Problems." *Journal of the ACM* 23, 555–565.

Schreiber, G.R. and O. Martin. (1999). "Cut Size Statistics of Graph Bisection Heuristics." *SIAM Journal on Optimization* 20(1), 231–251.

Stadler, P.F. (1995). "Landscapes and their Correlations Functions." Technical Report 95-07-067, Santa Fe Institute, Santa Fe, NM.

Taillard, E.D. (1991). "Robust Taboo Search for the QAP." *Parallel Computing* 17, 443–455.

Taillard, E.D. (1995). "Comparison of Iterative Searches for the Quadratic Assignment Problem." *Location Science* 3, 87–105.

Taillard, E.D. www.idsia.ch/~eric.

Vollmann, T.E. and E.S. Buffa. (1966). "The Facilities Layout Problem in Perspective." *Management Science* 12(10), 450–468.

Weinberger, E.D. (1990). "Correlated and Uncorrelated Fitness Landscapes and How to Tell the Difference." *Biological Cybernetics* 63, 325–336.