A Scatter Search Based Approach for the Quadratic Assignment Problem

Van-Dat CUNG, Thierry MAUTOR, Philippe MICHELON , Andréa TAVARES

N ° 96/037

December 30^{th}, 1996

A Scatter Search Based Approach for the Quadratic Assignment Problem

Van-Dat Cung, Thierry Mautor, Philippe Michelon[*], [‡] Andréa Tavares[†] [‡]

Thème/Theme 1 — Parallélisme/Parallelism
Équipe/Team PNN[§]

Rapports de Recherche n° 96/037 — December 30th, 1996 — 10 pages

Abstract: In this report, we are mainly interested in Scatter Search which is an evolutionary heuristic, proposed two decades ago, that uses linear combination of a population subset to create new solutions. A special operator is used to ensure their feasibility and to improve their quality.

We propose hereafter a Scatter Search approach to the Quadratic Assignment Problem (QAP) problem. The basic method is extended with intensification and diversification stages and we present a procedure to generate good scattered initial solutions.

Key-words: Scatter Search, Tabu Search, Evolutionary Algorithms, Heuristic Methods, Quadratic Assignment Problem (QAP), Combinatorial Optimization.

(Résumé : tsvp)

[§] Email : Firstname.Name@prism.uvsq.fr or pnn@prism.uvsq.fr

[*] Département d'Informatique et Recherche Opérationelle, Université de Montréal, BP 6128, Succ. Centre Ville, Montréal H3C 3J7, Qc, Canada. Email: michelon@iro.umontreal.ca

[†] Departamento de Informática, Pontifícia Universidade Católica, Rio de Janeiro, Brasil. Email: andrea@inf.puc-rio.br

[‡] This work has been achieved while the author was visiting the PR*i*SM Laboratory.

Laboratoire PR*i*SM - CNRS URA 1525
Université de Versailles - Saint Quentin en Yvelines
78035 Versailles Cedex, FRANCE.
Téléphone/Phone : (33) 0(1) 39 25 40 56 – Télécopie/Fax : (33) 0(1) 39 25 40 57

Recherche Dispersée pour le Problème de l'Affectation Quadratique

Résumé : Dans ce rapport, nous nous intéressons à la "Recherche Dispersée", une méthode heuristique qualifiée d'"évolutionniste" et proposée il y a deux décennies par Fred GLOVER. La Recherche Dispersée est fondée principalement sur l'utilisation de la combinaison linéaire d'un sous-ensemble d'une population pour engendrer de nouvelles solutions. Un opérateur spécial est également intégré dans la méthode afin d'assurer la réalisabilité et la qualité des solutions.

Nous proposons ci-après une approche de la Recherche Dispersée pour le problème de l'Affectation Quadratique. La méthode de base est enrichie des étapes d'intensification et de diversification. De même, une procédure permettant d'engendrer des solutions initiales bien dispersées est présentée.

Mots-clé : Recherche Dispersée, Recherche Tabou, Algorithmes Évolutionnistes, Méthodes Heuristiques, Affectation Quadratique, Optimisation Combinatoire.

Contents

1 Introduction 1

2 Combining solutions 2

3 The operator 4

4 Adding diversification and intensification 5

5 Generating the initial collection of points 6

6 Numerical results 7

1 Introduction

Introduced in 1977 by Fred Glover [4] (see also [5, 2]), Scatter Search has been largely underlooked for about twenty years. Nevertheless, the method has the same kind of features as most of modern and efficient heuristics. It furthermore easily allows to introduce some advanced techniques like adaptative memory programming, strategic oscillation etc (see [5] for an interesting discussion on these topics). Therefore, Scatter Search appears to be a very appealing framework for the design of a heuristic method.

Basically, the Scatter Search method starts with a collection of feasible solutions. At each step, some of the best solutions are extracted from the collection to be combined. A "trial " point is then created as a linear combination of the extracted points and an operator is applied to the trial point. This operator has two purposes. In many cases, a linear combination of integer points will not result in an integer point. The first purpose of the operator is thus to produce an integer (feasible) solution from the trial point. The second purpose is to improve the quality of the created solution. As a result of the operator, a new feasible solution is obtained which might be included or not (according to some criteria) in the collection.

There are some great differences between Scatter Search and Genetic Algorithms although, in both of them, a set of feasible solutions evolves. First of all, there is no metaphor with the nature's behavior in Scatter Search. Its rational is rather of geometric or analytic type : by taking a linear combination of good solutions, one might expect to obtain a new good solution. Second, this is the very first method which allows to combine more than two solutions. Therefore, at each step, Scatter Search gets more information that a Genetic Algorithms does. On the other hand, this information might be more difficult to analyze or to exploit. For instance, by combining a large number of solutions, one might combine different subregions of the feasible set and get a trial point with no particular signification. Note however that in this case the operator could lead to a subregion not explored yet. Overall, as indicated by its name, the method induces a real willingness for maintaining the collection points as scattered as possible, hence to have a good diversification (the operator being in charge of intensification). However, as emphasized in [5], intensification (resp. diversification) can also be achieved by identifying clusters in the collection and by combining points from the same cluster (resp. from different clusters).

The aim of this paper is to illustrate how Scatter Search could be used for solving the Quadratic Assignment Problem (QAP).

For the sake of simplicity, we will refer to the QAP as the problem of assigning n firms to n sites and a feasible solution will be represented by a $n \times n$ matrix X such that

$$\begin{aligned} \sum_{i=1}^{n} X_{ik} &= 1, \quad k \in \{1, ..., n\} \\ \sum_{k=1}^{n} X_{ik} &= 1, \quad i \in \{1, ..., n\} \\ X_{ik} &= 0 \text{ or } 1 \end{aligned}$$

and $X_{ik} = 1$ means that firm k is assigned to site i.

Obviously, the QAP which is still one of the most important and challenging 0-1 quadratic problems, has been widely studied. Heuristic methods for the QAP include Tabu Search [1, 8, 9, 7], Genetic Hybrid Algorithms [3], Simulated Annealing [10] among others.

The remaining of the paper is organized according to the main steps of the algorithm (Fig. 1).

```
SCATTER SEARCH
Generate initial population (section 5)
WHILE not StopCondition DO
    • Select solutions to be combined (section 2)
    • Generate new solution[1] (section 2)
    • Apply operator to improve new solution[1] (section 3)
    • Insert new solution in population (section 3)
```

[1] According to the iteration, one might use different procedures to perform intensification or diversification (section 4).

Figure 1: High level description of Scatter Search procedure.

2 Combining solutions

As indicated above, at each step, Scatter Search extracts some of the best solutions. For this purpose, a subset (let's say the Elite subset) of the collection of points, containing the best available solutions, is maintained. The cardinality of the Elite subset E is a parameter of the method. Another parameter R concerns the maximum number of solutions one wishes to combine. In practice, E might be quite large while R should be relatively small ($R \simeq 5, |E| \simeq n$). Given these parameters, a number r in $\{2, 3, ..., R\}$ is randomly chosen, according to a uniform distribution. Then, also following a uniform distribution, r points (let's say $X^1, ..., X^r$) are drawn in E.

For the selection process, we also carried out some numerical tests with a roulette wheel selection over the whole collection of points but the results appeared to be worst than with the above elitist strategy. This can be explained by the fact that the main advantage of a roulette wheel strategy is that it prevents from selecting almost always the same points, and hence, increases the diversification in the population. It happens that, in a Scatter Search approach, the diversification is ensured by many other rules (described below). Therefore, the roulette wheel selection loses its main interest.

Once $X^1, X^2, ..., X^r$ are selected, we have to combine them to obtain a trial point $T = \sum_{l=1}^{r} \lambda_l X^l$, where $\lambda_l \in \mathbb{R}$, $l = 1, ..., r$. T is thus a linear combination of the selected points and is generally not feasible. We then have to deduce from T a feasible solution. A natural way to reach this goal is to find out the feasible solution X which is the closest to T, that is, we have to solve the problem.

$$\begin{aligned} \min \quad & \|X - T\|^2 & (1) \\ s.t. \quad & \sum_{i=1}^{n} X_{ik} = 1, \quad k = 1, ..., n & (2) \\ & \sum_{k=1}^{n} X_{ik} = 1, \quad i = 1, ..., n & (3) \\ & X_{ik} \in \{0,1\} \quad i = 1, ..., n\,, k = 1, ..., n & (4) \end{aligned}$$

where $\|X - T\|$ is the euclidean distance between X and T.

Actually, as shown below, such a problem reduces to a linear assignment problem :

$$\begin{aligned} \|X - T\|^2 &= \sum_{i=1}^{n}\sum_{k=1}^{n}(X_{ik} - T_{ik})^2 \\ &= -2\sum_{i=1}^{n}\sum_{k=1}^{n} T_{ik}X_{ik} + \sum_{i=1}^{n}\sum_{k=1}^{n} X_{ik}^2 \\ &\quad + \sum_{i=1}^{n}\sum_{k=1}^{n} T_{ik}^2. \end{aligned}$$

The last two terms are constants since for any feasible solution we have $\sum_{i=1}^{n}\sum_{k=1}^{n} X_{ik}^2 = n$, and $\sum_{i=1}^{n}\sum_{k=1}^{n} T_{ik}^2$ does not depend on X.

It follows that X is simply an optimal solution of:

$$\max \sum_{i=1}^{n}\sum_{k=1}^{n} T_{ik}X_{ik}$$

$$s.t. \quad (2), (3), (4).$$

Although this linear assignment problem can be solved in $O(n^3)$, we prefer to solve it by the “not-so-greedy” heuristic described in Fig. 2.

The heuristic constructs a feasible solution X from T in $O(n^2)$. Furthermore, it will be explained in section 4 how this method easily allows introducing some diversity in the whole process. Another interesting feature relies on its random nature : if we apply it several times to the same T, it will probably construct different feasible solutions X. Hence, even if the same $X^1, X^2, ..., X^r$ are drawn at two different iterations of the process, we might expect to construct two different new solutions.

```
/* Initialization */
AvailableSites = $\{1, ..., n\}$
AvailableFirms = $\{1, ..., n\}$
$X_{ik} = 0, \quad i = 1, ..., n \quad k = 1, ..., n$
/* Main Steps */
FOR $t = 1, n$ DO
    • Choose at random (according to a uniform distribution) a site $i \in$ AvailableSites
    • Choose at random (according to a uniform distribution) a firm $k \in$ Argmax$\{T_{ik} | k \in$ AvailableFirms$\}$
    • $X_{ik} = 1$
    • AvailableSites = AvailableSites-$\{i\}$
    • AvailableFirms = AvailableFirms-$\{k\}$
```

Figure 2: Heuristic for combining.

We must point out that we tried several other heuristic procedures for obtaining a feasible solution X without any significative change in the numerical results.

In practice, we just fix $\lambda_l = 1, \quad l = 1, ..., r$.

3 The operator

Since we obtained an integer feasible solution, the aim of the operator is now only to improve the cost of this solution. It was then decided to use a very basic Tabu Search method for a fixed number of iterations.

In this method, a move consists in exchanging the firms located on sites i and j, and the neighborhood of a given point is defined as the set of solutions which can be reached in one move from the given solution. Once a move, which locates firm k_1 on i_2 and firm k_2 on i_1 has been done, the opposite move (which consists of leading back k_1 on i_1 and k_2 on i_2) will be considered as tabu (prohibited) for a given number of T iterations, except if it improves the overall best solution. Note that this is probably the most unrestrictive tabu list (for instance it is still allowed to exchange firms on i_1 and i_2 if firm on i_1 is no more k_2 or if firm on i_2 is no more k_1). We designed such an unrestrictive Tabu Search since, in our opinion, the operator plays a role of intensification in the whole process.

In practice, we observe that T had almost no practical effect for small size problems (let's say $n \leq 90$) while $T \simeq 200$ seems to work well for large size problems.

As a result of the operator, a new solution Y is obtained and compared with the worst element of the population (collection of points). The new solution is included in the collection if it is better than the worst element. In a earlier version of the algorithm, the new point Y was compared with its closest point in the collection. The results were slightly worst. One more time, this can be explained by the fact that the main advantage of such a criteria is to enhance the diversity of the collection of points (in particular, it ensures of not generating a point which is yet in the population). However, as mentioned earlier for the roulette wheel selection strategy, the diversification is ensured from one hand by the heuristic procedure for combining the points and, from the other hand, by some additional rules which are stated in the next section. Therefore, this strategy loses its main advantage and has the drawback of suppressing points which might be good.

4 Adding diversification and intensification

The first way to keep the collection of points as scattered as possible is to prevent that some points, maybe slightly better than the others, are too often used for combination. This can easily be achieved by assigning a tabu status, for a given number of iterations (let's say n), to a point which has been used for combination. Such an idea was used in [2].

Another way of maintaining a certain level of diversity in the population is to consider, from time to time, some assignments of firms to sites which have not been used frequently before. For this purpose, a frequency matrix $(n \times n)$ F is updated at each iteration of the Scatter Search process. F_{ik} reports on the number of times where firm k was assigned to site i in the starting points of the Tabu Search (operator). Hence, each time a point X is computed by the heuristic of section 2, F is updated by $F = F + X$. Whenever we wish to increase the diversity of the collection of points, the procedure of Fig. 3 is used to create a partial solution with $\lceil cn \rceil$ (c is a parameter in [0,1], in practice $c \simeq 0.05$) assignments of firms to sites which were not frequently used.

```
/* Initialization */
AvailableSites = $\{1, ..., n\}$
AvailableFirms = $\{1, ..., n\}$
$X_{ik} = 0, \quad i = 1, ..., n \quad k = 1, ..., n$
/* Main Steps */
FOR $t = 1, \lceil cn \rceil$ DO
    • Choose at random (according to a uniform distribution) a site $i \in$ AvailableSites
    • Choose at random (according to a uniform distribution) a firm $k \in$ Argmin$\{F_{ik} | k \in$ AvailableFirms$\}$
    • $X_{ik} = 1$
    • AvailableSites = AvailableSites-$\{i\}$
    • AvailableFirms = AvailableFirms-$\{k\}$
```

Figure 3: Procedure for adding diversification.

The remaining $n - \lceil cn \rceil$ assignments of firms to sites are then computed by the heuristic of section 2 (based on the combination of points). As usual, the generated point will be the starting point of the Tabu Search.

It is also interesting to add some intensification in the whole process. This is done by increasing the number of iterations in the Tabu Search procedure run. Hence, the Tabu Search procedure runs $I1$ iterations (in practice $I1 \simeq 80$ in an ordinary diversification Scatter Search iteration and $I2$ iterations ($I2 \simeq 1200$) in an intensification Scatter Search iteration.

In the whole process, we alternate between the three types of iteration, that is : we first carry out X ordinary iterations, then Z intensification iterations and then Y diversification iterations and this cycle is repeated until the maximum number of iterations of Scatter Search procedure is reached. Good values for X, Z and Y seem to be 7, 1, 1.

5 Generating the initial collection of points

An aspect of evolutionary algorithms which might be sometimes neglected is the generation of the initial population : most of the methods generate it randomly. However, our opinion is that the initial collection of points should meet the following criteria :

1. Since all along the algorithm, points which are issued of the initial population are combined, all the possibilities should be contained in the initial population. In other words, using the Genetic Algorithms terminology, all the possible genes should be represented in the population. For the QAP, this means that for any firm k and any site i there should be a point in the initial collection for which the firm k is located on the site i.

2. Since the subregion containing the optimal solution is generally unknown, the initial collection of points should cover as good as possible the feasible set, in order to be able to reach any of its subregions. Therefore, the initial population should be as scattered as possible. This can be achieved by generating a population which maximizes the euclidean minimum distance between any couple of points. For the QAP, we therefore should try to generate points such that the distance between any two points is $\sqrt{2n}$ (i.e. the maximum distance between two feasible points).

In order to satisfy these criteria, we first generate at random a feasible solution X^1. Let now π be a circular permutation of $\{1, ..., n\}$. A second solution X^2 is constructed by assigning $\pi(k_i^1)$ to site i, where k_i^1 is the firm located on the site i in X^1, and a third solution X^3 is then constructed by assigning $\pi(k_i^2)$ to site i, where k_i^2 is the firm located on the site i in X^2, etc. The process is repeated $n-1$ times and it can be easily shown that these n solutions satisfy the criteria. If one wishes, as we do, to generate more than n solutions, a new random solution is created and the above procedure might be used again. In practice, we generate an initial collection of $2n$ points. The Tabu Search operator (section 3) is then applied to every solution before its insertion in the population.

Although this procedure is valid for any circular permutation π, we compute a permutation π as indicated in Fig. 4. Its computation is based on the observation that in the QAP the distance matrix plays a central role, inducing the existence of symmetric solutions (see [6] for a discussion on this topic). Therefore, in order to obtain a good cover of the feasible set, the permutation π is computed in such a way that it prohibits whenever it is possible for a couple of firms to be located at the same distance in two different solutions.

```
TabuList=$\emptyset$
$i = 1$
Set site 1 as used
REPEAT $n - 1$ TIMES
    • Search first $\pi - i$ such that $\pi - i$ is not used and $d(i, \pi - i)$ is not tabu
    • IF $\nexists(\pi - i)$ THEN
        – TabuList=$\emptyset$ /* reinitialization */
        – Search first $\pi - i$ such that $\pi - i$ is not used
    • $\pi(i) = \pi - i$
    • Set $\pi - i$ as used
    • Insert $d(i, \pi - i)$ in TabuList
    • $i = \pi - i$
$\pi(1) = i$
```

Figure 4: Computation of π.

6 Numerical results

We first would like to emphasize that, due to a lack of time, we did not fully exploit the Scatter Search opportunities. Especially, our intensification iterations consist in using the same Tabu Search procedure as in an ordinary iteration but for a higher number of steps. Our feeling is that the intensification proposed in [5] could improve the results.

The above Scatter Search procedure has been implemented in C++ and numerical tests were performed according to the QAP Problems Performance Measure Protocol of the second international contest on evolutionary algorithms. The results are reported in table 1. It appears that the algorithm finds quite quickly the best known solution for most of the test problems. It also finds some solutions which are better than the solution reported in QAPLIB as the best known solution for the problem tai256c. However, for the problems tai100a, tai150b and tho150, the method does not find the best known values.

Even in these cases, the best values found are within 0.09% of the best known value for tho150, within 0.52% for tai100a, within 0.07% for tai150b. These gaps could probably be eliminated by a better tuning of the parameters.

Problems		1	50	100	500	1000	2500
Els19	Avrg	17228707	17212548	17212548	17212548	17212548	17212548
	Best	17212548	17212548	17212548	17212548	17212548	17212548
		Average Total Time: 45.6					
Bur26a	Avrg	5428736	5428113	5427615	5427269	5427016	5427016
	Best	5426670	5426670	5426670	5426670	5426670	5426670
		Average Total Time: 96.7					
Kra30a	Avrg	90535	90032	89364	89232	89232	89232
	Best	88900	88900	88900	88900	88900	88900
		Average Total Time: 127.4					
Lipa90b	Avrg	15081175	13502112	12994369	12490441	12490441	12490441
	Best	15053917	12490441	12490441	12490441	12490441	12490441
		Average Total Time: 1967.3					
Sko100a	Avrg	155055	153608	153292	152654	152242	152166
	Best	152834	152324	152324	152184	152002	152002
		Average Total Time: 2522.4					
Tai100a	Avrg	21589668	21436282	21418348	21344212	21338965	21310178
	Best	21308480	21308480	21308480	21285858	21285858	21235942
		Average Total Time: 2524.8					
Tho150	Avrg	8449562	8253571	8244509	8184747	8161579	8150452
	Best	8338586	8178680	8178680	8151454	8141208	8141208
		Average Total Time: 15326.2					
Tai150b	Avrg	509165448	505956700	504053100	501179165		
	Best	507745428	503161401	501165936	499710131		
		Average Total Time: 3629.5					
Tai256c	Avrg	45324250	44871017	44856403	44832872		
	Best	44824542	44824542	44823712	44823712		
		Average Total Time: 15762.1					

Table 1: Computational results (times in seconds are obtained on a DEC AlphaStation 500/400Mhz).

Acknowledgements

The authors would like to thank Diane Béland for the afternoon she spent with us, and F.-X. Robin, J. Bertin, J. Rousseau, S. Letniowski and Y. Simonet for having developed some preliminary versions of the algorithm.

Special thanks to DEC France for allowing us to test our program on a DEC AlphaStation 500/400Mhz.

References

[1] R. Battiti and G. Tecchiolli. The reactive tabu search. *ORSA Journal on Computing*, 6(2):126–140, 1994.

[2] P. M. C. Fleurent, F. Glover and Z. Valli. A scatter search approach for unconstrained continuous optimization. In *Proceedings of the 1996 IEEE international conference on evolutionary computation*, pages 643–647, 1996.

[3] C. Fleurent and J. Ferland. Genetic hybrids for the quadratic assignment problem. In *Quadratic Assignment and Related Problems*, volume 16 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 173–187, 1994.

[4] F. Glover. Heuristic for integer programming using surrogate constraints. *Decision Sciences*, 8:156–166, 1977.

[5] F. Glover. Tabu search and adaptative memory programming: Advances, applications and challenges. In Helgason and Kennington, editors, *Interfaces in Computer Sciences and Operations Research*. Kluwer academic publishers, 1996. To appear.

[6] T. Mautor and C. Roucairol. A new exact algorithm for the solution of quadratic assignment problems. *Discrete Applied Mathematics*, 55:281–293, 1994.

[7] J. Skorin-Kapov. Tabu search applied to the quadratic assignment problem. *ORSA Journal on Computing*, 2(1):33–45, 1990.

[8] E. Taillard. Robust tabu search for the quadratic assignment problem. *Parallel Computing*, 17:443–455, 1991.

[9] E. Taillard. Comparison of iterative searches for the quadratic assignment problem. *Location Science*, 1994. To appear.

[10] U. Thonemann and A. Bölte. An improved simulated annealing algorithm for the quadratic assignment problem. Working paper, School of Business, Department of Production and Operations Research, University of Paderborn, Germany, 1994.

Laboratoire PRiSM-CNRS URA 1525, Université de Versailles - Saint Quentin en Yvelines,
45 avenue des État-Unis, 78035 Versailles Cedex, FRANCE.