Multiobjective algorithms performance on photo album QAP instance

Supervisors : F. Teytaud, S. Verel.
Jérôme Buisine, *IT Student at ULCO Calais.*

Contents

1 Abstract **2**

2 Presentation of photo album QAP **3**
2.1 QAP explanation . . . 3
2.2 Single objective QAP definitions . . . 3
2.3 mQAP photo album formulation . . . 4
2.4 Objectives choices . . . 4
2.5 Test context presentation . . . 5

3 Multiobjective algorithms studied **6**
3.1 Random walk Multiobjective algorithm . . . 6
3.2 Pareto Local Search algorithm . . . 6
3.3 MOEA/D algorithm . . . 6
3.4 TP-LS algorithm . . . 7
3.5 Performance comparisons . . . 9
3.5.1 ND & D information . . . 10
3.5.2 Hyper volume information . . . 11

4 Conclusion **12**

1 Abstract

This paper presents the results of my research project I have done during my first year Master's Degree. Its aim is to learn more about different kind of Multiobjective algorithm, compare them and demonstrate which algorithm can be well adapted for a specific quadratic assignment problem instance.

State of art :
The quadratic assignment problem (QAP) was introduced by Koopmans and Beckmann in 1957 [1] in the context of locating "*indivisible economic activities*". The objective of the problem is to assign a set of facilities to a set of locations in such a way as to minimize the total assignment cost. The assignment cost for a pair of facilities is a function of the flow between the facilities and the distance between the locations of the facilities. It's one of the most challenging problem from single-objective combinatorial optimization.

In this paper I'll first present you the photo album QAP instance and mainly the multiobjective QAP (mQAP) treated. In a second time, we will take attention of different multiobjective algorithms implemented for this QAP instance and compare them. Then, conclude and define in which way it would be interesting to continue and investigate.

Note : All algorithms source code is in Scala multi paradigm language. Scala has been selected to get benefit of its functional paradigm for this mQAP. The platform used for test suites is a Cloud platform solution with 1 vCPU and 1.7 GB of RAM.

2 Presentation of photo album QAP

2.1 QAP explanation

As said earlier, "Photo album problem" is a QAP instance. In our case, a single-objective QAP can be defined with one matrix which represents the similarity score between photos and the second, which represents the Euclidean distance between each location on a page. We can have multiple page's location, such as example 6 pages which contains 2 per 3 photo locations. The assignment of 4 photos to 4 locations can be written as the permutation :

$$p = \{1, 2, 4, 3\} \tag{1}$$

This assignment means that photo 1 is assigned to location 1, photo 2 is assigned to location 2, photo 4 is assigned to location 3 and then, photo 3 is assigned to location 4. Imagine picture 1 is into page 1 and picture 40 is into page 6 means we consider picture 1 is far away from picture 40 in terms of similarity criteria. The objective of this Quadratic Assignment Problem is to assign each photo to a location in such a way as to minimize the total cost.

2.2 Single objective QAP definitions

A single objective QAP mathematical representation can be defined as follows.

Sets

- A solution is represented by :
$$N = \{1, 2, ..., n\}$$
- All feasible solutions is described by :
$$S_n = \phi : N \to N$$
- $V(N)$ is the set of neighbor's solutions of N.
- As we have to permute two photos of the solution N to obtain the neighbor set, we'll write the size of $V(N)$ with n the size of the solution N as follows.
$$Size(V(N)) = \frac{n!}{2!(n-2)!}$$
Binomial coefficient is used in our case to avoid repetitions of solutions.

Parameters

- $S = (s_{ij})$ is an $n \times n$ matrix where s_{ij} is the computed similarity distance between photos i and j.
- $D = (d_{ij})$ is an $n \times n$ matrix where d_{ij} is the euclidean distance between photos i and j.

Optimization Problem

- The single objective fitness to minimize for this QAP problem is formalized as follows :
$$\min_{\phi \in S_n} \sum_{i=1}^{n} \sum_{j=1}^{n} s_{ij}.d_{\phi(i)\phi(j)}$$

The assignment of photos to locations is represented by permutation ϕ, where $\phi(i)$ is the location to which photo i is assigned. Each individual product $s_{ij}.d_{\phi(i)\phi(j)}$ is the cost of assigning photo i to location $\phi(i)$ and photo j to location $\phi(j)$.

2.3 mQAP photo album formulation

Now we have defined the single objective photo album QAP instance, we have to introduce multiobjective QAP. mQAP is used when we want to minimize or maximize a problem based on multiple criteria. We always have the matrix distance, but our main objective has to follow multiple objective based on specific criteria. So, our mQAP can treat n similarity matrix. In this paper, we will take a look at the different criteria which can be used and select two of them for test suites.

Following [3], this mQAP treated on this paper can be formalized as follows :

$$\min f_1(\phi) = \sum_{i=1}^{n}\sum_{j=1}^{n} s_{ij}^1 . d_{\phi(i)\phi(j)}$$

$$\min f_2(\phi) = \sum_{i=1}^{n}\sum_{j=1}^{n} s_{ij}^2 . d_{\phi(i)\phi(j)}$$

where (f_1, f_2) is the pair of objective functions to be minimized, n the number of photos, $S_1 = (s_{ij}^1)$ and $S_2 = (s_{ij}^2)$ are both an $n \times n$ matrix of constant values either positive or zero. Zero value means that the similarity is flagrantly, so consequently it's the same or current photo.

2.4 Objectives choices

In this project we have multiple criteria to assume that photo i is simular to photo j or as contrary assume they are opposite. Our goal is to compute similarity matrix based on these criteria. Below you have a list of these different kind of criteria objectives.

1. Hash algorithms values :
   - a-hash
   - p-hash
   - d-hash

   For each hash, matrix values are computed as below.

$$\sum_{i=1,j=1}^{n} \sum_{k=1,m=1}^{n} |x_{i,m} - x_{j,k}|$$

2. Semantic tags :
   - Common tags values
   - Uncommon tags values
   - Number of uncommon tags

   For each tags, matrix values are computed as follows.

$$\sum_{i=1,j=1}^{n} \sum_{k=1,m=1}^{n} |x_{i,m} - x_{j,k}|$$

3. Colors :

   In this project we treat only two colors. These colors are the most dominant colors of the photo. Distance color formula based on RGB values is used to obtain photo value and compute matrix.

$$F(C_1, C_2) = \sqrt{((R_{C_1} - R_{C_1})^2) \times ((G_{C_1} - G_{C_1})^2) \times ((B_{C_1} - B_{C_1})^2)}$$

4. Grey AVG :

   Another criteria is the grey average present on the photo. Matrix of this criteria is computed as follows.

$$\sum_{i=1,j=1}^{n} \sum_{k=1,m=1}^{n} |x_{i,m} - x_{j,k}|$$

After defining all these criteria, we have to know that only non-correlated criteria are important to consider having better results. So, into this paper we will take attention of the *commonTags* and *grey average* criteria.

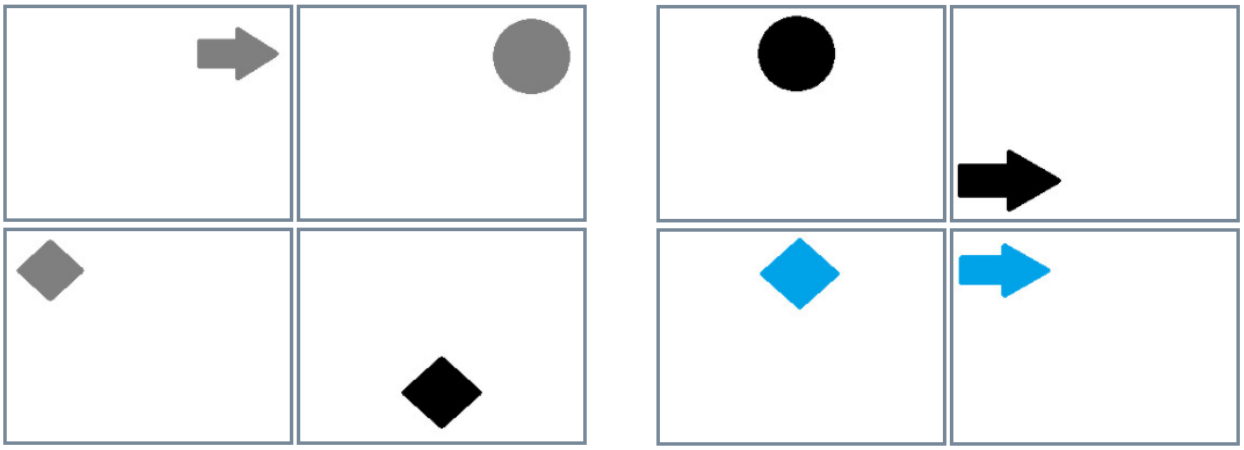
2.5 Test context presentation

For all test suites demonstrate later, a test context is defined. This context is established first of all to make comparison easier and to obtain great results quickly.

This test context is defined as follows.

- $N = \{1, 2, ..., 16\}$
- 4 pages which each contains a 2 per 2 matrix photos.
- "*Common tags*" and "*grey average*" the two similarity criteria to minimize.
- Following a major need of performance, research will be carried out on only ***20,000*** iterations for each algorithm.
- Photos with different grey average values and tags such as example forms, positions and colors.

A photo disposition in our test suites is represented as follows.



Each algorithm performance will be rated with features based on $V(N)$, the neighbor set of current solution iteration treated. We will regard mean and standard deviation for each thousand iterations. Thus, comparisons will show which algorithm is better for our mQAP.

3 Multiobjective algorithms studied

Before beginning our test suites, we take a look at the different exploited algorithms. Each algorithm has its benefits and also its disadvantages.

For all of these algorithms, we will use the term "dominated" or "non dominated". This term indicates the dominance relation of solutions. A solution $\phi \in S_n$ is dominated by a solution $\phi' \in S_n$, denoted as $\phi \prec \phi'$, if $f_k(\phi') <= f_k(\phi)$ for all $k \in [1, 2]$, with at least one strict inequality. If neither $\phi \not\prec \phi'$ nor $\phi' \not\prec \phi$ holds, then both solutions are *mutually non-dominated*.

3.1 Random walk Multiobjective algorithm

Random walk algorithm goal is to simply search solution randomly from the search space at each iteration and refresh the non dominated solution. Just below the implementation of this algorithm.

Algorithm 1: Random walk

Input: ***nbEval*** evaluation stopping criteria
Output: A

```
1 A := θ;
2 evaluation := 0;
3 repeat
4 |   s := select randomly a solution;
5 |   A := A + s;
6 |   A := getNonDominated(A);
7 |   evaluation := evaluation + 1;
8 until evaluation >= nbEval;
```

3.2 Pareto Local Search algorithm

Pareto Local Search algorithm is a kind of archiving algorithm. It's aim is to course all solutions of the search space to obtain all non dominated solutions. These solutions produce the Pareto Front and they are all Pareto optima solutions if all solutions have been seen. A preview implementation of this algorithm is available below.

Algorithm 2: Pareto Local Search

Input: A_0 an initial set of non dominated solutions, ***nbEval*** evaluation stopping criteria
Output: A

```
1  A := A_0;
2  explored := A_0;
3  evaluation := 0;
4  repeat
5  |   s := select randomly a solution ∉ A;
6  |   foreach s' ∈ V(s) do
7  |   |   if s' ∉ explored then
8  |   |   |   A := A + s';
9  |   |   |   A := getNonDominated(A);
10 |   |   |   evaluation := evaluation + 1;
11 |   |   end
12 |   |   explored := explored +s';
13 |   end
14 until evaluation >= nbEval;
```

3.3 MOEA/D algorithm

The Multiobjective Evolutionary Algorithm Based on Decomposition [4] is a scalarizing approach with population. Scalarizing approaches consist in transforming the original problem into a single-objective one

by means of an aggregation of the objective function values. There is two typical scalarizing functions used into this paper.

- Weighted-sum scalarizing function [2] defined below.

$$g_\lambda(x) = \lambda_1.f_1(x) + \lambda_2.f_2(x)$$

  where $x \in S_n$ is a candidate solution, and $\lambda = (\lambda_1, \lambda_2)$ is a weighting coefficient vector.

- Tchebycheff scalarizing function defined as follows.

$$g_\lambda(x) = \min\left\{\lambda_1 * |f_1(x) - r_1|, \lambda_2 * |f_2(x) - r_2|\right\}$$

  where r is a reference point in the objective space, as example $r(0, 0)$.

In this paper, we will compare these two approaches on the MOEA/D algorithm. The implementation of this algorithm is described below.

Algorithm 3: Multiobjective Evolutionary Algorithm Based on Decomposition

Input: ***N*** the number of sub problem, ***T*** the number of the weight vectors in the neighborhood of each weight vector, ***g*** the single objective scalarizing approach, ***nbEval*** evaluation stopping criteria

Output: EP

```
EP := θ;
λ := computeWeightVectors(N);
B := generating with B(i) = {i_1, ..., i_T} where λ_i1, ..., λ_iT are the closest weight vectors to λ_i;
P := initial population x_1, ..., x_N of each sub problem set randomly;
FV := matrix which contains objective values of each P solution where FV_i is the F-Value of x_i
 represented as FV_i = F(x_i);
z := reference point generating with min value of each objective found so far into FV;
evaluation := 0;
repeat
    for i := 0 to N do
        k, l := random indexes from B(i);
        s := new solution from {x_k, x_l} using genetic operators;
        s' := new solution produce from s using improvement heuristic;
        z := set min value of each objective found so far into FV to update reference point z;
        for j := 0 to T do
            if g(s') < g(P(j)) then
                P(j) := s';
                FV_j := F(s');
                EP := EP + P(j);
                EP := getNonDominated(EP);
            end
            evaluation := evaluation + 1;
        end
    end
until evaluation >= nbEval;
```

3.4 TP-LS algorithm

The Two-Phase Local Search [2] consists in a hybrid two-phase approach. In this paper we will apply MOEA/D and PLS algorithm in a sequential way. We also look at the different scalarizing approach used on MOEA/D and their benefits on the PLS algorithm. Below the implementation of this algorithm.

Algorithm 4: Two-phase Local Search

Input: ***N*** the number of sub problem, ***T*** the number of the weight vectors in the neighborhood of each weight vector, ***g*** the single objective scalarizing approach, ***nbEvalMOEAD*** MOEAD evaluation stopping criteria, ***nbEvalPLS*** PLS evaluation stopping criteria

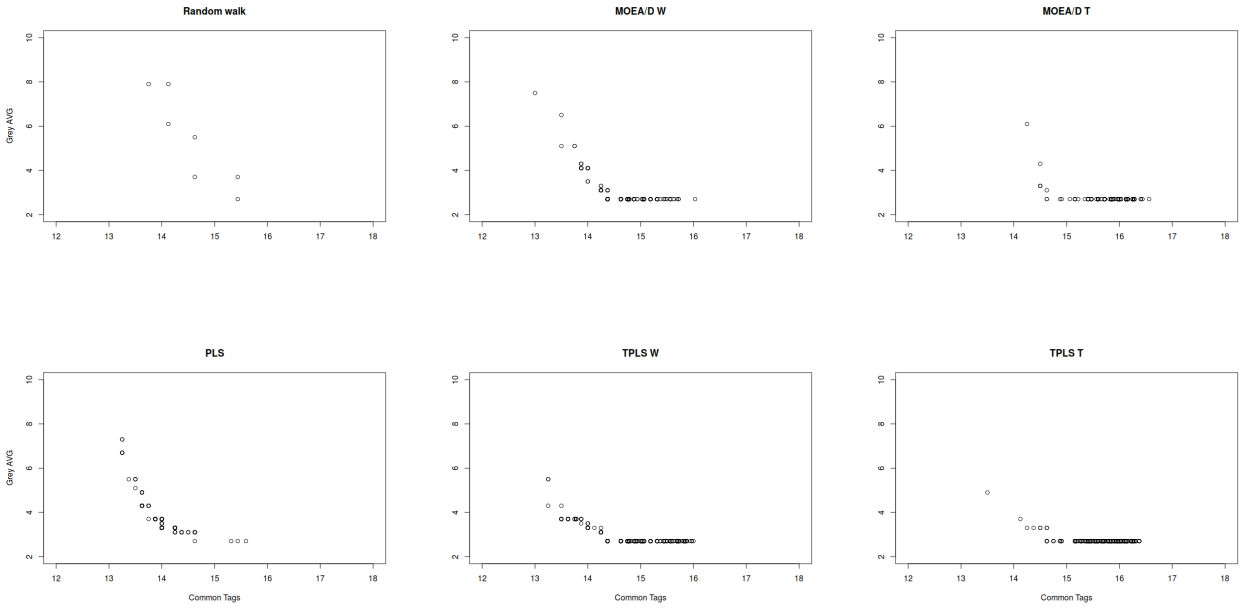
Output: A

```
1 A := MOEAD_Algo(nbEvalMOEAD, N, T, g);
2 A := PLS_Algo(nbEvalPLS, A);
```

3.5 Performance comparisons

As said before, our test suites is based on a maximum of 20.000 iterations. First of all we compare landscape of each algorithm. The two phase algorithms are composed of 10.000 iterations for MOEA/D and 10.000 iterations for PLS algorithm. Note that **W** means *weighed sum* and **T** *Tchebycheff* scalarizing single-objective approach.

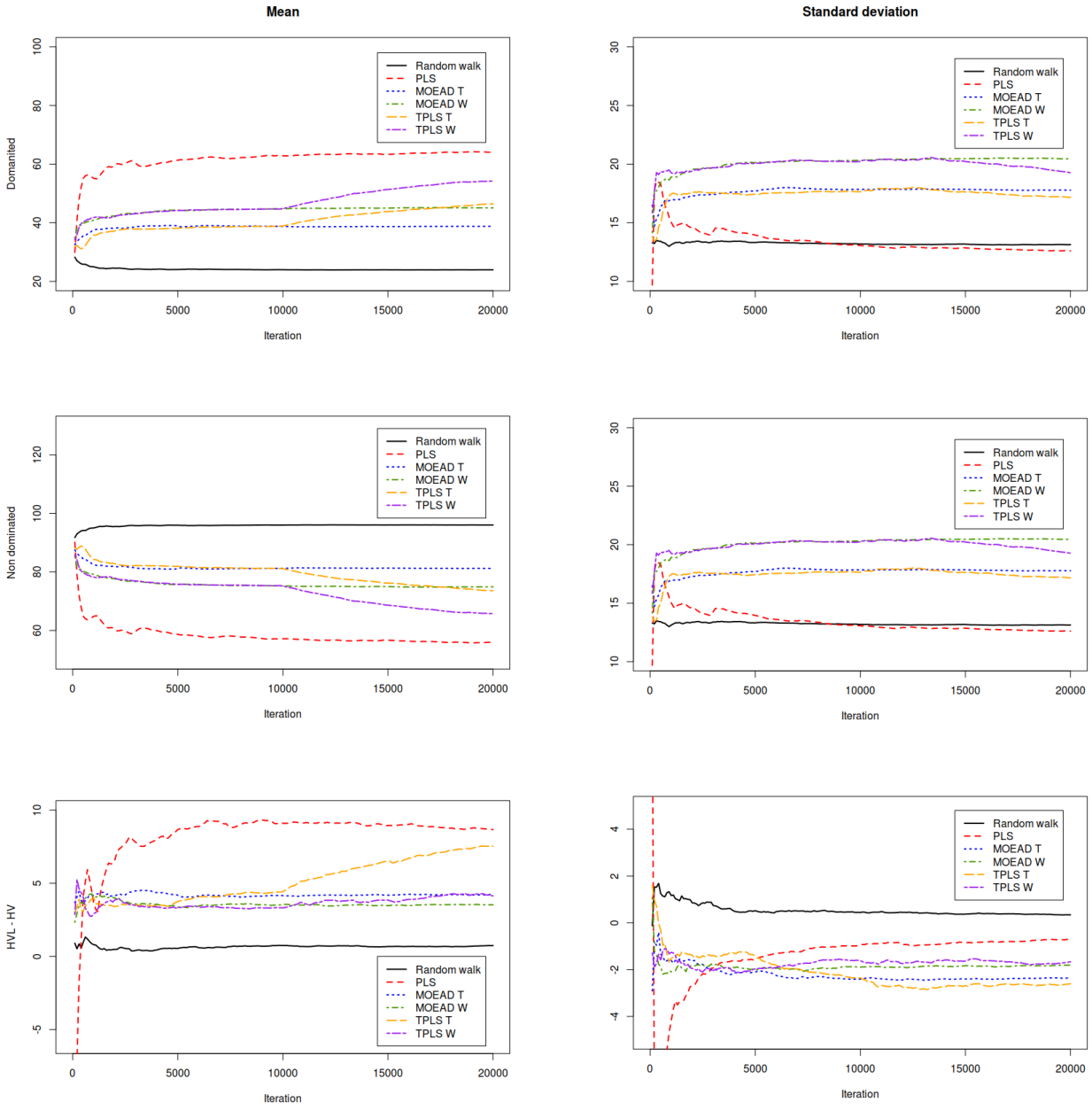
I have to say I didn't show photo album preview because the data set is too small to compare it visually. So, I had preferred to compare computations results of all algorithms used. Below the different graphs results.



Random walk is perhaps better in terms of computing resources but graph shows it is not a well perform algorithm for our QAP. It does not found a large set of non dominated solutions. The Pareto front is represented by the PLS archiving algorithm graph, but it needs more computing resources than all others algorithms.

MOEA/D algorithm give fast results after Random walk. They do not approach the results of PLS algorithm but it's a well compromise in terms of performance and time. TPLS with the weighted sum as single-objective function give nice solutions and deeper than PLS and better than TPLS with Tchebycheff single-objective implementation. In general, it seems that Tchebycheff single-objective approach take care of only one objective. We could also say that weighted sum approach appear to be more suitable for or QAP instance.

Landscapes of each algorithm gave some information about their performance. Now we will take look at the mean and standard deviation for each algorithm. These statistics are computed incrementally after each 1000 iterations, from 0 to 20.000. These statistics are based on the neighbor set of the solution such as number of dominated neighbors, number non dominated neighbors, Hyper volume difference between hyper volume of the current solution and hyper volume local (Hyper volume of its neighbors).



3.5.1 ND & D information

With these graphs we have some information about the behaviors of the algorithms. We can see that non-dominated and dominated mean graphs are opposite. The Random walk algorithm does not vary. As the search is random, it's so hard for this algorithm to reaches always better solutions without any peace of tractability information. PLS algorithm has high dominated neighbor solutions quickly. That's information means PLS algorithm reach nice solutions faster than others algorithms and non-dominated in the neighbor set is rare. MOEAD algorithms grow quickly, but they stay constant after a certain number of interations. TP-LS algorithm grow up when PLS is thrown. We can note that the weighted sum seems to be more performing than Tchebycheff approach.

Standard deviation graphs based on dominated and non dominated values tell us that the PLS and Random walk algorithm have low variation after a certain time. As contrary the others continue to vary during the computation. Perhaps PLS algorithm can't find new non dominated solutions readily as a lot of nice solutions have been already seen and random walk is so random and can find good solutions easily.

3.5.2 Hyper volume information

The mean of hyper volume difference on these algorithms let us say that random walk algorithm does not change during computation as he's so random. That's explain why random has a low number of non dominated solutions. The PLS and TPLS T algorithm have better hyper volume difference than others and it means these algorithm have a lot of non dominated solutions, certainly more than the other algorithms. The other algorithms seem to don't change. Although the TPLS W algorithm grow up slowly when its PLS phase begin. I think PLS and two phase algorithms have great results in terms of hyper volume because they always search for new local or gloabl optima easily like they are archive algorithm.

For the standard deviation we can also demonstrate that random walk does not find better solution quickly. Moreover, all others algorithms have negative standard deviation values which signify they find better solutions during computation. Except random walk algorithm, after a certain time they are stable. We can also say that the TPLS T algorithm decreases more than others and PLS increases lower. PLS algorithm perhaps encounters problems to obtain good solutions after a certain number of iterations.

4 Conclusion

In terms of algorithm performance I think the two-phase local search algorithms is a favorable compromise. These kind of algorithms decrease a lot the computation time and bring nice results as shown in this paper.

The first part of this two-phase algorithm is a key fact of final results. The choice of the scalarizing single objective approach is very important. The weighted sum approach corresponds better for this QAP instance. This approach seems to take care of the two criteria used in this paper. It could be also interesting to compare computation results with more than 2 criteria.

Scala language used for this project is greedy in terms of computation needs. To have multiple and better result values, C++ is perhaps a better language I have to use for the future.

Another point not treated in this paper is the possibility to add an enhancement algorithm with bonus and penalties on one or multiple objective criteria. We can apply this kind of algorithm when we proposed solution to client, let him modify photos disposition before validate and compare the impact on objectives. Then, set bonus and penalties in function of these latest modifications after comparison. Each criteria significance can be personalized for the client and its future album.

References

[1] T. C. Koopmans and M. Beckmann. Assignment problems and the location of economic activities. *Econometrica: journal of the Econometric Society*, pages 53–76, 1957.

[2] A. Liefooghe, B. Derbel, S. Vérel, H. E. Aguirre, and K. Tanaka. A fitness landscape analysis of pareto local search on bi-objective permutation flowshop scheduling problems. In H. Trautmann, G. Rudolph, K. Klamroth, O. Schütze, M. M. Wiecek, Y. Jin, and C. Grimme, editors, *Evolutionary Multi-Criterion Optimization - 9th International Conference, EMO 2017, Münster, Germany, March 19-22, 2017, Proceedings*, volume 10173 of *Lecture Notes in Computer Science*, pages 422–437. Springer, 2017.

[3] A. Liefooghe, S. Vérel, L. Paquete, and J. Hao. Experiments on local search for bi-objective unconstrained binary quadratic programming. In A. Gaspar-Cunha, C. H. Antunes, and C. A. C. Coello, editors, *Evolutionary Multi-Criterion Optimization - 8th International Conference, EMO 2015, Guimarães, Portugal, March 29 -April 1, 2015. Proceedings, Part I*, volume 9018 of *Lecture Notes in Computer Science*, pages 171–186. Springer, 2015.

[4] Q. Zhang and H. Li. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Trans. Evolutionary Computation*, 11(6):712–731, 2007.