Photo album multiobjective QAP

Master's Degree first year project

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

ULCO Calais, June 2017

Table of Contents

1. Problem presentation
   - QAP definitions
   - Photo album mQAP
2. Algorithms studied
3. Algorithms comparisons
4. Web platform
5. Conclusion

1.1. QAP definitions

Origin

QAP was introduced by Koopmans and Beckmann in 1957

1.1. QAP definitions

Origin

QAP was introduced by Koopmans and Beckmann in 1957

Explanation

- NP-Hard problem

1.1. QAP definitions

Origin

QAP was introduced by Koopmans and Beckmann in 1957

Explanation

- NP-Hard problem
- Assign a set of facilities to a set of locations

1.1. QAP definitions

Origin

QAP was introduced by Koopmans and Beckmann in 1957

Explanation

- NP-Hard problem
- Assign a set of facilities to a set of locations
- Minimize the total assignment cost

1.1. QAP definitions

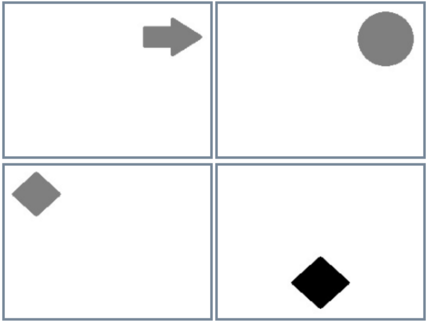


Figure: Page 1

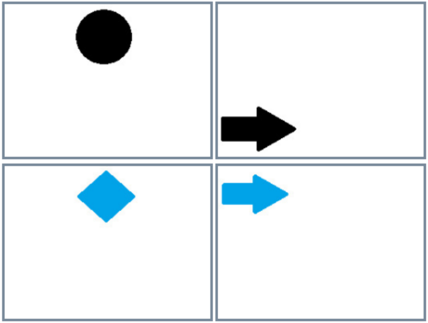


Figure: Page 2

Example

$$p = \{7, 3, 1, 2, 8, 5, 6, 4\}$$

Solution definitions

- $N = \{1, 2, ..., n\}$, the solution representation

1.1. QAP definitions

Solution definitions

- $N = \{1, 2, ..., n\}$, the solution representation
- $S_n = \phi : N \rightarrow N$, the set of all permutations

1.1. QAP definitions

Solution definitions

- $N = \{1, 2, ..., n\}$, the solution representation
- $S_n = \phi : N \rightarrow N$, the set of all permutations
- $V(N)$, set of neighbor solutions

1.1. QAP definitions

Solution definitions

- $N = \{1, 2, ..., n\}$, the solution representation
- $S_n = \phi : N \rightarrow N$, the set of all permutations
- $V(N)$, set of neighbor solutions

Matrix of QAP

- $S = (s_{ij})$ 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.

1.1. QAP definitions

Solution definitions

- $N = \{1, 2, ..., n\}$, the solution representation
- $S_n = \phi : N \rightarrow N$, the set of all permutations
- $V(N)$, set of neighbor solutions

Matrix of QAP

- $S = (s_{ij})$ 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.

Single objective function to minimize

$$\min_{\phi \in S_n} \sum_{i=1}^{n} \sum_{j=1}^{n} s_{ij} . d_{\phi(i)\phi(j)}$$

Multiobjective function to minimize with $k \in [1, 2]$

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

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

1.2. Photo album mQAP

Multiobjective function to minimize with $k \in [1, 2]$

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

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

Domination definition

$$\phi \prec \phi', \text{ if } f_k(\phi') <= f_k(\phi) \text{ for all } k \in [1, 2]$$

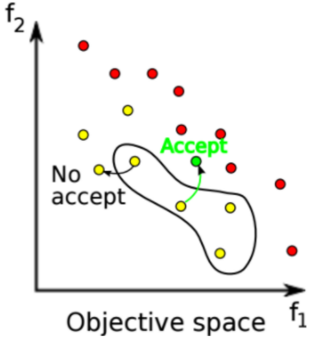
2.1. Random walk

Algorithm 1: Random walk

Input: **nbEval** evaluation stopping criteria

Output: A

```
A := θ;
evaluation := 0;
repeat
    s := select randomly a solution;
    A := A + s;
    A := getNonDominated(A);
    evaluation := evaluation + 1;
until evaluation >= nbEval;
```



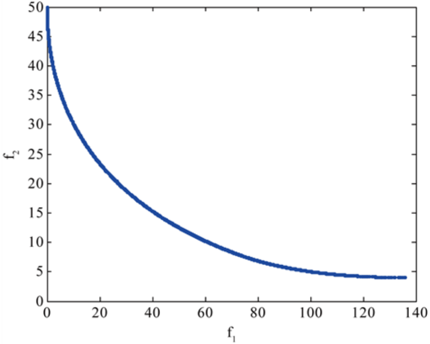
2.2. Pareto Local Search

Algorithm 2: Pareto Local Search

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

Output: A

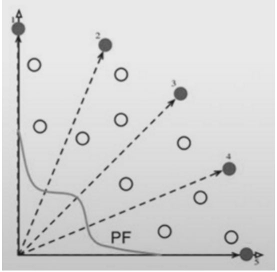
$A := A_0$;
explored $:= A_0$;
evaluation $:= 0$;
repeat
    $s :=$ select randomly a solution $\notin A$;
    foreach $s' \in V(s)$ **do**
        if $s' \notin$ *explored* **then**
            $A := A + s'$;
            $A :=$ getNonDominated(A);
            evaluation $:=$ evaluation $+ 1$;
        end
        explored $:=$ explored $+s'$;
    end
until *evaluation* $>=$ *nbEval*;



2.3. MOEA/D - Weighted sum

Multiobjective Evolutionary Algorithm Based on Decomposition

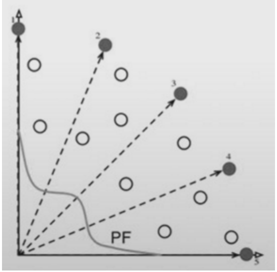
Method which decomposes multiobjective problems into a number of scalar sub problems and optimizes them simultaneously.



2.3. MOEA/D - Weighted sum

Multiobjective Evolutionary Algorithm Based on Decomposition

Method which decomposes multiobjective problems into a number of scalar sub problems and optimizes them simultaneously.



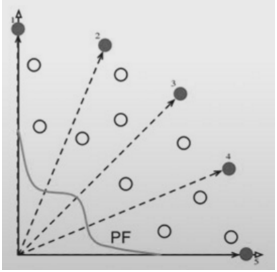
Weighted sum single objective scalarizing method

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

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

Multiobjective Evolutionary Algorithm Based on Decomposition

Method which decomposes multiobjective problems into a number of scalar sub problems and optimizes them simultaneously.



Tchebycheff single objective scalarizing method

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

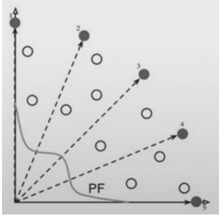
where r is a reference point in the objective space, as example $r(0,0)$.

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*

$EP := \theta$;
$\lambda :=$ *computeWeightVectors*(N);
$B :=$ generating with $B(i) = \{i_1, ..., i_T\}$ where $\lambda_{i1}, ..., \lambda_{iT}$ 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*;



1 Problem presentation

2 Algorithms studied

- Random walk
- Pareto Local Search
- MOEA/D
- Two-Phase Local Search

3 Algorithms comparisons

4 Web platform

5 Conclusion

2.4. Two-Phase Local Search

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

$A :=$ MOEAD_Algo(nbEvalMOEAD, N, T, g);
$A :=$ PLS_Algo(nbEvalPLS, A);

3.1. Test context

Language

All algorithms source code is in Scala multi paradigm language. Scala has been selected to get benefit of its functional paradigm for this mQAP.

Test platform

The platform used for test suites is a Cloud platform solution with 1 vCPU and 1.7 GB of RAM

3.1. Test context

Album photo size and disposition

- $N = \{1, 2, ..., 16\}$

3.1. Test context

Album photo size and disposition

- $N = \{1, 2, ..., 16\}$
- 4 pages which each contains a 2 per 2 photos matrix.

3.1. Test context

Album photo size and disposition

- $N = \{1, 2, ..., 16\}$
- 4 pages which each contains a 2 per 2 photos matrix.

Multiobjective function to minimize with $k \in [1, 2]$

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

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

3.1. Test context

Album photo size and disposition

- $N = \{1, 2, ..., 16\}$
- 4 pages which each contains a 2 per 2 photos matrix.

Multiobjective function to minimize with $k \in [1, 2]$

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

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

Criteria choice

- $f_1 \rightarrow$ Grey AVG

3.1. Test context

Album photo size and disposition

- $N = \{1, 2, ..., 16\}$
- 4 pages which each contains a 2 per 2 photos matrix.

Multiobjective function to minimize with $k \in [1, 2]$

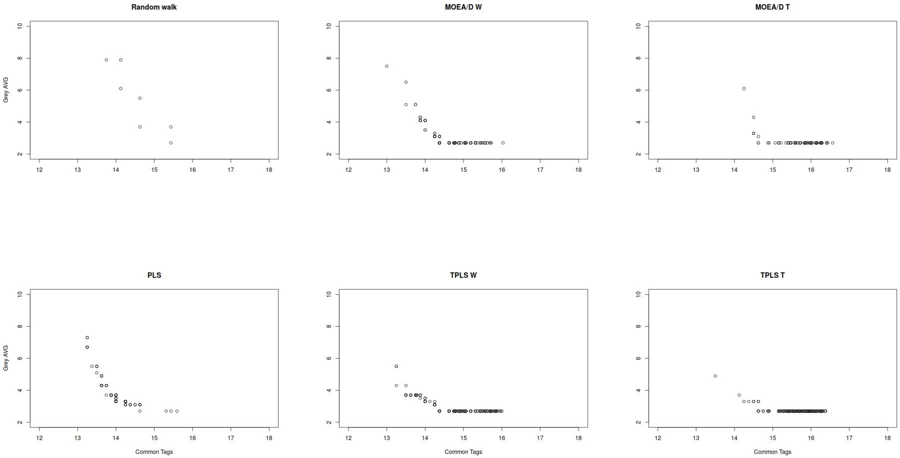
$$\min f_1(\phi) = \sum_{i=1}^{n} \sum_{j=1}^{n} s_{ij}^1 . d_{\phi(i)\phi(j)}$$

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

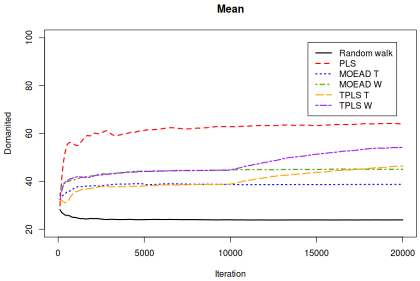
Criteria choice

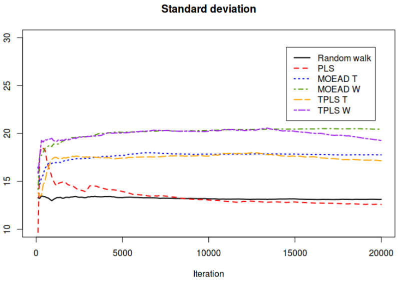
- $f_1 \rightarrow$ Grey AVG
- $f_2 \rightarrow$ Common Tags

3.2. Landscapes

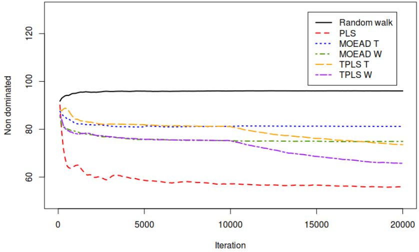


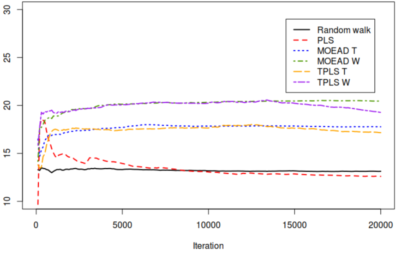
3.3. Features - Dominated feature



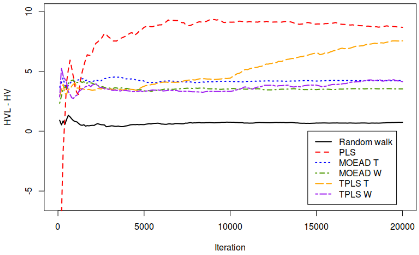


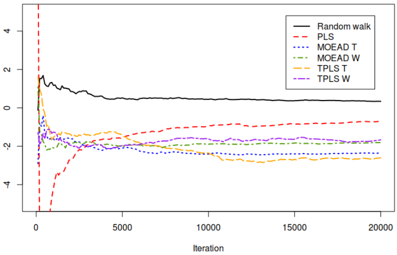
3.3. Features - Non dominated





3.3. Features - (HVL - HV)





4. Web platform

5. Conclusion

- Two-phase Local Search a good compromised
- Algorithm complexity
- Other programming language
- Client customization