HAL
open science

Experiments on local search for bi-objective unconstrained binary quadratic programming

Arnaud Liefooghe, Sébastien Verel, Luis Paquete, Jin-Kao Hao

To cite this version:

Arnaud Liefooghe, Sébastien Verel, Luis Paquete, Jin-Kao Hao. Experiments on local search for bi-objective unconstrained binary quadratic programming. EMO - 2015 8th International Conference on Evolutionary Multi-Criterion Optimization, Mar 2015, Guimarães, Portugal. pp.171-186, 10.1007/978-3-319-15934-8_12 . hal-01093261

HAL Id: hal-01093261

https://hal.science/hal-01093261v1

Submitted on 19 Dec 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Experiments on Local Search for Bi-objective Unconstrained Binary Quadratic Programming

Arnaud Liefooghe[1], Sébastien Verel[2], Luís Paquete[3], and Jin-Kao Hao[4]

[1] CRIStAL, UMR CNRS 9189, Université Lille 1 — Inria Lille-Nord Europe, France
arnaud.liefooghe@univ-lille1.fr
[2] LISIC, Université du Littoral Côte d'Opale, France
verel@lisic.univ-littoral.fr
[3] CISUC, Department of Informatics Engineering, University of Coimbra, Portugal
paquete@dei.uc.pt
[4] LERIA, Université d'Angers, France
jin-kao.hao@univ-angers.fr

Abstract. This article reports an experimental analysis on stochastic local search for approximating the Pareto set of bi-objective unconstrained binary quadratic programming problems. First, we investigate two scalarizing strategies that iteratively identify a high-quality solution for a sequence of sub-problems. Each sub-problem is based on a static or adaptive definition of weighted-sum aggregation coefficients, and is addressed by means of a state-of-the-art single-objective tabu search procedure. Next, we design a Pareto local search that iteratively improves a set of solutions based on a neighborhood structure and on the Pareto dominance relation. At last, we hybridize both classes of algorithms by combining a scalarizing and a Pareto local search in a sequential way. A comprehensive experimental analysis reveals the high performance of the proposed approaches, which substantially improve upon previous best-known solutions. Moreover, the obtained results show the superiority of the hybrid algorithm over non-hybrid ones in terms of solution quality, while requiring a competitive computational cost. In addition, a number of structural properties of the problem instances allow us to explain the main difficulties that the different classes of local search algorithms have to face.

1 Introduction

The unconstrained binary quadratic programming (UBQP) problem is one of the most challenging problem from single-objective combinatorial optimization [11]. Given a collection of n items such that each pair of items is associated with a profit value that can be positive, negative or zero, the UBQP problem seeks a subset of items that maximizes the sum of their paired values. The value of a pair is summed up only if the two corresponding items are selected. From a computational point-of-view, a feasible solution to a UBQP instance can be represented as a binary string of size n. Each position from the binary string maps to a particular variable that indicates whether the corresponding item is included

in the subset of selected items or not. Beyond its theoretical significance [8], the utility of UBQP has been demonstrated on a wide variety of application fields [11]. Furthermore, a number of NP-hard combinatorial optimization problems can be recast as UBQP problems, such as graph coloring, max-cut, set packing, set partitioning, or maximum clique, among others [11]. The single-objective UBQP problem has received a growing interest in recent years [9, 11], and a multi-objective extension of UBQP has been proposed recently [12].

In this paper, we focus on bi-objective UBQP, where two profit values are associated with each pair of items. By optimizing both sums of profit values simultaneously, we can improve the descriptive power of the conventional single-objective UBQP problem, and provide a more general formulation. However, as for many problems from multi-objective combinatorial optimization, the bi-objective UBQP problem raises several difficulties for heuristics design. In particular, the number of optimal solutions can be very large [12], and determining whether a candidate solution is optimal is NP-complete, even in the single-objective case [8]. For these reasons, we design and experiment with multi-objective stochastic local search algorithms, and measure their efficiency and their effectiveness on instances with different dimensions and correlation degrees between the objective function values. Furthermore, we analyze the problem structure to learn more about those difficulties, and to improve the design of algorithms. The contributions of the paper are two-fold.

(*i*) We characterize the features of small-size, enumerable bi-objective UBQP instances. More particularly, we analyze the number of global and local optimal solutions, based on scalarizing functions and on the Pareto dominance relation; and we examine the connectedness between optimal solutions.

(*ii*) We design and analyze local search algorithms for bi-objective UBQP, including two scalarizing approaches, a Pareto-based approach, and a hybrid approach combining these two complementary search strategies. The designed algorithms substantially improve over the previous attempts in solving large-size bi-objective UBQP instances [12]. More importantly, our experimental analysis allows us to better understand how the performance of these classes of algorithms relates to the structural properties of the search space, explaining the high efficiency and effectiveness of the proposed approaches.

The remainder of the paper is organized as follows. In Section 2, we present the bi-objective UBQP problem. In Section 3, we study the characteristics of small-size instances. In Section 4, we introduce four stochastic local search algorithms for bi-objective UBQP. In Section 5, we analyze the performance of these algorithms on a set of large-size bi-objective UBQP instances. In Section 6, we finally conclude the paper and discuss further research directions.

2 Bi-objective UBQP

This section presents the problem formulation, some definitions related to multi-objective combinatorial optimization, and the problem instances that are investigated in the paper.

2.1 Problem Formulation

The bi-objective UBQP (bUBQP) problem can be formalized as follows [12].

$$\begin{aligned} \max \quad & f_1(x) = \sum_{i=1}^{n}\sum_{j=1}^{n} q_1^{ij} x_i x_j \\ \max \quad & f_2(x) = \sum_{i=1}^{n}\sum_{j=1}^{n} q_2^{ij} x_i x_j \\ \text{subject to} \quad & x \in \{0,1\}^n \end{aligned} \tag{1}$$

where (f_1, f_2) is the pair of objective functions to be maximized, n is the number of items, $Q_1 = (q_1^{ij})$ and $Q_2 = (q_2^{ij})$ are both an $n \times n$ matrix of constant values, either positive, negative or zero. As in the single-objective case, the solution space $X = \{0,1\}^n$ is defined on binary strings of size n; its size is then 2^n.

2.2 Definitions

We denote by $Z \subseteq \mathbb{R}^2$ the feasible region in the *objective space*, *i.e.* the image of feasible solutions when using the maximizing function vector $f = (f_1, f_2)$ such that $Z = f(X)$. The Pareto dominance relation is defined as follows. A solution $x \in X$ is dominated by a solution $x' \in X$, denoted as $x \prec x'$, if $f_k(x) \leq f_k(x')$ for all $k \in \{1,2\}$, with at least one strict inequality. If neither $x \not\prec x'$ nor $x' \not\prec x$ holds, then both solutions are *mutually non-dominated*. A solution $x \in X$ is *Pareto optimal* if there does not exist any other solution $x' \in X$ such that $x \prec x'$. The set of all Pareto optimal solutions is the *Pareto set*, and its mapping in the objective space is the *Pareto front*. One of the most challenging issues in multi-objective optimization is to identify a minimal complete Pareto set, *i.e.* one Pareto optimal solution mapping to each point from the Pareto front. Since the bUBQP problem is both NP-hard and intractable [12], approximate algorithms like stochastic local search are well suited to identify a *Pareto set approximation*.

2.3 Problem Instances

Following [12], the definition of each bUBQP objective function is based on a matrix Q_k, $k \in \{1,2\}$. As in the single-objective UBQP instances available in the OR-lib [3], non-zero matrix integer values are randomly generated following a uniform distribution in $[-100, +100]$. The density $d \in [0,1]$ gives the expected proportion of non-zero entries in the matrix. Following a Bernoulli distribution of parameter d, a given entry at position (i,j) is set to zero on *both* matrices, *i.e.* $q_1^{ij} = q_2^{ij} = 0$. Moreover, we define a correlation coefficient ρ between the data contained in the two matrices. The positive (respectively negative) data correlation decreases (respectively increases) the degree of conflict between the objective function values. The generation of correlated data follows a multivariate uniform distribution of dimension 2 [12]. As reported in Fig. 1(a), the coefficient ρ allows to tune the correlation between objective function values with a high accuracy. The considered bUBQP problem instances as well as an instance generator are available at the following URL: `http://mocolib.sf.net/`.

3 Characteristics of Small-Size bUBQP Instances

Thereafter, we study the impact of the density d and of the objective correlation ρ on the number of Pareto optimal solutions, Pareto local optimal solutions, supported solutions, and on the connectedness property of bUBQP instances. More particularly, we consider a density $d \in \{0.2, 0.4, 0.6, 0.8, 1.0\}$ and an objective correlation $\rho \in \{-0.9, -0.7, -0.4, -0.2, 0.0, +0.2, +0.4, +0.7, +0.9\}$. The problem size is set to $n = 18$ in order to enumerate the solution space exhaustively. For each parameter setting, 30 independently generated random instances are considered. Experimental results are given in Fig. 1(b–f). In the following, we provide a detailed analysis of these statistics.

3.1 Pareto Optimal Solutions

Fig. 1(b) shows the proportion of Pareto optimal solutions in the solution space. Interestingly, the density d does not affect the size of the Pareto set. However, the objective correlation ρ modifies the number of Pareto optimal solutions to several orders of magnitude. Indeed, almost 0.05% of the solution space correspond to non-dominated solutions for conflicting objectives ($\rho = -0.9$), whereas this number drops to less than 0.003% for correlated objective ($\rho = +0.9$). As a consequence, the larger the objective correlation ρ, the lower the cardinality of the Pareto set. This means that an algorithm is expected to take more time to identify the whole Pareto set when the objectives are in conflict.

3.2 Supported Solutions

In multi-objective optimization, *scalarizing* approaches consist in transforming the original problem into a single-objective one by means of an aggregation of the objective function values. A typical example is the weighted-sum scalarizing function [6] that can be defined as follows.

$$g_\lambda(x) = \lambda_1 \cdot f_1(x) + \lambda_2 \cdot f_2(x) \tag{2}$$

where $x \in X$ is a candidate solution, and $\lambda = (\lambda_1, \lambda_2)$, such that $\lambda_1, \lambda_2 \geq 0$, is a weighting coefficient vector. Supported solutions are non-dominated solutions which are optimal with respect to a weighted-sum aggregation of the objective functions. Their corresponding objective vectors are located on the boundary of the convex hull of the Pareto front [6]. On the contrary, non-supported solutions are not optimal for any setting of the weighting coefficient vector λ. In order to explain the ability of scalarizing multi-objective optimization approaches to identify a large portion of Pareto optimal solutions, we should put the problem-related properties in relation with the proportion of *supported* solutions. Fig. 1(c) shows the proportion of supported solutions in the Pareto set. Once again, the matrix density d has a very small influence. However, when the objective correlation increases, and despite the absolute number of supported solution actually gets lower, the proportion of supported solutions on the Pareto set increases.

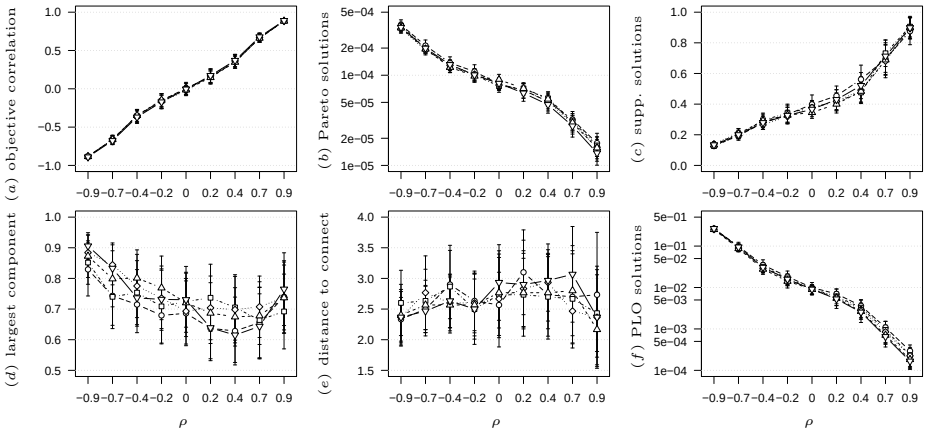

Fig. 1. (a) Spearman correlation coefficient between the objective function values, (b) ratio of the number of Pareto optimal solutions to the solution space size, (c) ratio of the number of supported solutions to the Pareto set size, (d) ratio of the size of the largest connected component of the Pareto graph for Hamming distance 1 to the Pareto set size, (e) minimal Hamming distance to connect the Pareto graph, and (f) ratio of Pareto local optimal solutions to the solution space size, with respect to the objective correlation ρ, for $d = 0.2$ ($\circ$), $d = 0.4$ ($\square$), $d = 0.6$ ($\diamond$), $d = 0.8$ ($\triangle$) and $d = 1.0$ ($\triangledown$). For each parameter setting, average values and confidence intervals (with a significance level of 10^{-2}) are reported over 30 independently generated random instances. The problem size is $n = 18$. Notice the log-scale on the y-axis for (b) and (f).

For highly correlated objectives ($\rho = +0.9$), nearly all Pareto optimal solutions are supported (this is even the case for some of the instances). On the contrary, for conflicting objectives ($\rho = -0.9$), only 15% of Pareto optimal solutions are supported. By putting this property in relation with algorithm design, we can assume that scalarizing approaches should be more suited to approximate the Pareto set of bUBQP instances with correlated objectives.

3.3 Connectedness

In the following, we describe some properties related to the *connectedness* of the Pareto set [7]. We follow the definition of *k-Pareto graph* from [16]. The k-Pareto graph is a graph $PG_k = (V, E)$, where each vertex in V corresponds to a Pareto optimal solution, and there is an edge $e_{ij} \in E$ between two nodes i and j only if the shortest distance between solutions x_i and $x_j \in X$, with respect to a given neighborhood, is below a bound k. For bUBQP, we adopt the Hamming distance on binary strings. This corresponds to the number of moves performed with the *bit-flip* neighborhood operator. Fig. 1(d) shows the ratio between the size of the largest connected component in the 1-Pareto graph ($PG_{k=1}$) and the size of the Pareto set. The objective correlation ρ has a clear impact on this feature. Indeed, the proportion of Pareto optimal solutions in the largest connected component decreases from $\rho = -0.9$ to $\rho = 0.4$, and then slightly increases from $\rho = 0.4$ to $\rho = 0.9$. Overall, we can expect to reach 50% to 95% of the whole Pareto set by

iteratively exploring the neighborhood of an approximation set initialized with at least one non-dominated solution. However, when there are several connected components in the 1-Pareto graph, it may happen that the distance between those components is small. Fig. 1(e) reports the smallest distance k such that the k-Pareto graph becomes connected, *i.e.* for all pairs of vertices $x_i, x_j \in V$ in PG_k, there is a path between x_i and x_j. When this minimal distance k is around 9, which is the average distance between random solutions for $n = 18$, we can conclude that the distance between Pareto optimal solutions is large. Actually, for bUBQP instances, this minimal distance is clearly smaller (between 2 and 3 in average). This means that finding a subset of non-dominated solutions can actually help to identify additional ones, which then may constitute a valuable asset for initializing local search algorithms.

3.4 Pareto Local Optimal Solutions

In Fig. 1(f), we report the proportion of Pareto Local Optimal (PLO) solutions [14] in the solutions space. A solution $x \in X$ is a PLO with respect to a neighborhood structure $\mathcal{N}$ if there does not exist any neighboring solution $x' \in \mathcal{N}(x)$ such that $x \prec x'$. As above, the neighborhood structure is taken as the 1-*bit-flip*, which is directly related to a Hamming distance 1. Once again, the distribution d does not seem to affect the number of PLO. However, similar to the trend observed on the Pareto set cardinality, the objective correlation ρ modifies the number of PLO to several orders of magnitude, from 20% of the solution space for $\rho = -0.9$ to less than 0.02% for $\rho = +0.9$. Therefore, by assuming that the difficulty for Pareto-based local search gets higher when the number of PLO is large, the difficulty of bUBQP instances might increase with the degree of conflict between the objectives.

4 Local Search for bUBQP

In this section, we give the working principles of four stochastic local search algorithms for identifying a Pareto set approximation to bUBQP instances. We start by introducing the algorithmic components shared by the different approaches. Then, we present the search strategies of two scalarizing local search algorithms, one Pareto-based local search algorithm, as well as a hybrid approach where a scalarizing and a Pareto local search phases are sequentially applied.

4.1 Main Ingredients

Neighborhood Relation. Similarly to the previous analysis, the neighborhood structure of the proposed local search algorithms is based on the 1-*bit-flip* operator: Two feasible solutions are neighbors if they differ exactly on one variable. In other words, a given neighbor can be reached by changing the value of a binary variable to its complement from the current solution. The size of the 1-*bit-flip* neighborhood structure is equal to the problem size n. As in the

single-objective UBQP, each bUBQP objective function can be evaluated incrementally. We follow the fast incremental evaluation procedure proposed in [9] to calculate the move gain of a given neighbor. For each objective, the whole set of neighbors is evaluated in linear time. As a consequence, the objective values of *all* neighboring solutions are evaluated in $\mathcal{O}(n)$ in the two-objective case.

Tabu Search. The tabu search algorithm proposed in [10] is reported to be one of the best-performing approaches for single-objective UBQP. In order to extend it to the multi-objective case, we consider a simple weighted-sum aggregation, as presented in Section 3.2, so that the initial objective vector values are (temporarily) transformed into a single scalar fitness value. Once the objective values of a given neighboring solution have been (incrementally) evaluated, we compute its scalar fitness value with respect to the weighted-sum problem (Eq. 2) for a given definition of the weighing coefficient vector. As a short-term memory, we maintain the tabu list as follows: Revisiting solutions is avoided within a certain number of iterations, called the *tabu tenure*. The tabu tenure of a given variable x_i is denoted by $tenure(i)$. Hence, variable x_i will *not* be flipped again for a number of $tenure(i)$ iterations. Following [9], we set the tabu tenure of a given variable x_i after it has been flipped as $tenure(i) = tt + rand(10)$, where tt is a user-given parameter and $rand(10)$ gives a random integer value in $[1, 10]$. From the set of neighbors produced by all non-tabu moves, we select the one with the best (highest) fitness value. However, all the neighbors are always evaluated, and a tabu move can still be selected if it produces a better solution than the current global best; this is called an *aspiration criterion* in tabu search [10]. The stopping condition is satisfied when no improvement has been performed within a given number of moves α, called the *improvement cutoff*. For more details on the tabu search algorithm for single-objective UBQP, the reader is referred to [10].

4.2 Scalarizing Local Search with Uniform Weights ($\texttt{SLS}_{\texttt{unif}}$)

The first approach consists in solving different settings of the weighted-sum problem (Eq. 2) by means of multiple weighting coefficient vectors defined in a way that the whole region of the Pareto front is covered in the objective space. For solving each scalarizing sub-problem, any algorithm for the resulting single-objective problem version can potentially be applied. In our case, we use the tabu search algorithm detailed above as a (single-objective) solver. Let us consider a set of μ uniformly defined weighting coefficient vectors $(\lambda^0, \ldots, \lambda^i, \ldots, \lambda^{\mu-1})$, such that $\lambda_1^i = i/(\mu - 1)$ and $\lambda_2^i = 1 - \lambda_1^i$. Each weighting coefficient vector λ^i corresponds to a scalarizing sub-problem WS^i. We start by identifying a high-quality solution with respect to the first objective function, corresponding to the scalarizing sub-problem WS^0, associated with the weighting coefficient vector $\lambda^0 = (0, 1)$. The final solution is then used as a seeding solution for solving the next sub-problem WS^1. We iterate this principle, each time the initial solution for sub-problem WS^i being the one that is returned by the tabu search algorithm for the previous sub-problem WS^{i-1}. At last, in order to avoid a bias in the

search process towards one objective, we re-run the same strategy by considering the reversed sequence of weighting coefficient vectors $(\lambda^{\mu-1}, \lambda^{\mu-2}, \ldots, \lambda^0)$. The algorithm outputs the union of non-dominated solutions generated during these two phases. The resulting scalarizing local search with uniform weights ($\texttt{SLS}_{\texttt{unif}}$) adapts the *"double two-phase local search"* from [15] to bUBQP by using the single-objective tabu search procedure for solving scalarizing sub-problems.

4.3 Dichotomic Scalarizing Local Search ($\texttt{SLS}_{\texttt{dicho}}$)

Similarly, the second approach is based on solving a sequence of scalarizing sub-problems, by means of a weighted-sum aggregation function, with the single-objective tabu search algorithm. However, unlike $\texttt{SLS}_{\texttt{unif}}$ that defines them *a priori*, the weighting coefficient vectors are now iteratively determined based on the solutions identified at previous steps. The resulting $\texttt{SLS}_{\texttt{dicho}}$ approach follows the principles of *dichotomic search* from exact bi-objective optimization [1], and adapt them to a local search engine strategy, similarly to [5]. Notice that, by using any exact algorithm instead of the tabu search procedure, such a dichotomic search would output the (exact) set of supported solutions [6]. Unfortunately, this would require to solve an NP-hard problem for each scalarizing sub-problem. Indeed, each sub-problem corresponds to a single-objective UBQP instance.

We start by identifying a high-quality solution for each separate objective. Let x^1 (resp. x^2) be the approximate solution found by tabu search for objective f_1 (resp. f_2). Both solutions are then added to a sequence $U_F = \{x^1, x^2, \ldots\}$, arranged in the decreasing order of f_1-values. Next, at each step of the algorithm, we define a weighting coefficient vector $\lambda = \big(f_2(x^2) - f_2(x^1), f_1(x^1) - f_1(x^2)\big)$, corresponding to the sub-problem to be solved in the current iteration. It gives a search direction that is perpendicular to the segment defined by $f(x^1)$ and $f(x^2)$ in the objective space. Let x be the solution identified by tabu search for this definition of λ. If $f_1(x^1) > f_1(x) > f_1(x^2)$ and $f_2(x^2) > f_2(x) > f_2(x^1)$, then x is added to the sequence U_F. Otherwise, we remove x^1 from U_F and add it to an external set U_T. Following [5], for each scalarizing sub-problem, we use the solutions found in previous iterations to seed the search process of the current iteration. Based on preliminary experiments, both x^1 and x^2 are here used as an initial solution for two independent runs of the tabu search procedure based on λ. The $\texttt{SLS}_{\texttt{dicho}}$ algorithm iterates this principle until U_F contains less than two elements, and returns the non-dominated solutions from $U_F \cup U_T$.

4.4 Pareto Local Search (PLS)

Let us now consider a Pareto approach based on a set of solutions and local search principles. In contrast to scalarizing approaches, the selection process is here directly based on the Pareto dominance relation. A typical example is the Pareto Local Search (PLS) algorithm [14]. An archive of mutually non-dominated solutions found so far is maintained in two different sets: V_F for non-dominated solutions whose neighborhood has not yet been explored, and V_T for solutions whose neighborhood has already been explored. These two sets are used in order

to avoid a useless re-evaluation of a solution's neighborhood. The algorithm starts with a set of mutually non-dominated solutions to initialize V_F, typically a single random solution. At each iteration, one unvisited solution is chosen at random from V_F. All its neighboring solutions are (incrementally) evaluated and checked for insertion in the archive. The current solution is then discarded from V_F and added to V_T, and dominated solutions are removed from $V_F \cup V_T$. The algorithm stops once V_F is empty, *i.e.* all solutions from the archive are *visited*. PLS always terminates and returns a maximal Pareto local optimum set [14].

4.5 Two-Phase Local Search (TP-LS)

The final algorithm consists in a hybrid two-phase approach, where $\texttt{SLS}_{\texttt{dicho}}$ and PLS are applied in a sequential way. It combines two fundamentally different and complementary search strategies: a scalarizing and a Pareto-based approach. In the first phase, $\texttt{SLS}_{\texttt{dicho}}$ is applied to identify a set of approximate supported solutions, as described in Section 4.3. This set of mutually non-dominated solutions is then used to initialize the archive V_F of the PLS algorithm, and is further improved by exploring the neighborhood of its own content until no improvement is possible. Hence, contrary to the conventional PLS, the search process does not start with a single random solution, but with a set of good-quality solutions identified by a scalarizing approach. The performance of the designed two-phase local search algorithm (TP-LS) should be impacted by the connectedness property for the problem under consideration; the more connected the Pareto optimal solutions, the easier to identify new non-dominated solutions from identified ones. Notice that TP-LS shares similar principles with existing approaches proposed for other problem classes [4, 13, 15].

5 Experimental Analysis

5.1 Experimental Design

We conduct an experimental study on the influence of the problem size (n) and of the objective correlation (ρ) over the performance of the proposed local search algorithms for approximating the Pareto set of bUBQP problem instances. In addition, we consider the best-known approximation sets (best-known) identified by multiple variants of evolutionary and memetic algorithms proposed in [12].

We investigate the following instance parameter setting: a problem dimension $n \in \{1000, 2000, 3000, 4000, 5000\}$ and a correlation between the objective function values $\rho \in \{-0.5, -0.2, 0.0, +0.2, +0.5\}$. The density of the matrices is set to $d = 0.8$. One instance, generated at random, is considered *per* parameter combination. This leads to a total of 25 problem instances. A set of 30 runs *per* instance is performed for each algorithm. All the algorithms start with a random solution. The tabu tenure tt is set to $n/150$ and the improvement cutoff α is set to n. At last, for each phase of $\texttt{SLS}_{\texttt{unif}}$, $\mu = 101$ weighting coefficient vectors $(\lambda^0, \ldots, \lambda^i, \ldots, \lambda^{100})$ are uniformly defined as $\lambda_1^i = i/100$ and $\lambda_2^i = 1 - \lambda_1^i$.

Since all the algorithms have a natural stopping condition, we measure their performance in terms of approximation set quality *and* computational cost. For each instance, we examine the quality of the Pareto set approximations identified by the competing algorithms in terms of hypervolume and epsilon indicators [17]. First, we compute the *hypervolume relative deviation* (`hypervolume`) as $(hv(R) - hv(A))/hv(R)$, where $A \subseteq Z$ is an approximation set and R is a reference set. The reference set is the best-found approximation over all tested configurations for the instance under consideration. Let z_k^- (resp. z_k^+) be the worst (resp. best) value obtained over all approximation sets for objective f_k, the reference point $\bar{z} = (\bar{z}_1, \bar{z}_2)$ for the hypervolume calculation is set to $\bar{z}_k = z_k^- - (z_k^+ - z_k^-) \cdot 10^{-2}$, $k \in \{1, 2\}$. Additionally, the epsilon indicator (`epsilon`) gives the minimum multiplicative factor by which an approximation set has to be shifted in the objective space in order to weakly dominate the reference set. In both cases, a lower indicator-value is better.

5.2 Experimental Results

A summary of our computational results is presented in Table 1, following the presentation from [2]. The first line corresponds to the bUBQP instance with $\rho = -0.5$ and $n = 1000$, and reports the quality of the Pareto set approximation obtained by the different algorithms with respect to `hypervolume`. The average hypervolume relative deviation obtained by $\texttt{SLS}_{\texttt{unif}}$, $\texttt{SLS}_{\texttt{dicho}}$, `PLS`, `TP-LS` and `best-known` over the 30 executions is respectively 0.009, 0.006, 0.002, 0.000 and 0.031. The ranking obtained by means of a pairwise Wilcoxon signed-rank non-parametric statistical test gives the following order for this particular setting: (1) `TP-LS`, (2) `PLS`, (3) $\texttt{SLS}_{\texttt{dicho}}$, (4) $\texttt{SLS}_{\texttt{unif}}$, and (5) `best-known`. Complementarily, Fig. 2 shows the average indicator-values for a subset of instances (the error bars indicate the confidence interval within a significance level of 10^{-2}). The results from `best-known` are omitted for a better readability.

Clearly, all the local search algorithms investigated in the paper largely improve over the previous `best-known` approximation sets from [12]. Indeed, for the 25 bUBQP instances under investigation, `best-known` obtains the lowest rank for 23 of them in terms of `hypervolume` and `epsilon`. A simple approach like $\texttt{SLS}_{\texttt{unif}}$ is able to obtain better `best-known` results in all the instances but one. Among the algorithms proposed in the paper, $\texttt{SLS}_{\texttt{unif}}$ is repeatedly dominated by the others with respect to both indicators. The only notable exceptions are for instances with correlated objectives where $\texttt{SLS}_{\texttt{unif}}$ performs better than `PLS` in terms of `hypervolume` (while it is slightly worse in terms of `epsilon`), and for large-size instances with conflicting objectives where `PLS` encounters more difficulties compared with other approaches. In both cases, the reason seems to be that the approximation sets identified by `PLS` badly covers the lexicographically optimal regions of the Pareto front. This is also the reason why $\texttt{SLS}_{\texttt{dicho}}$ outperforms `PLS` on more than half of the instances with respect to `hypervolume`, whereas the same only happens four times with respect to `epsilon`.

Interestingly, the quality of the approximation sets identified by scalarizing approches ($\texttt{SLS}_{\texttt{unif}}$, $\texttt{SLS}_{\texttt{dicho}}$) slightly increases with the objective correla-

Table 1. Comparison of the competing local search algorithms and of the previous best-known approximation [12] with respect to the hypervolume relative deviation (`hypervolume`) and to the unary multiplicative epsilon indicator (`epsilon`). The first value stands for the number of algorithms that statistically outperform the one under consideration with respect to a pairwise Wilcoxon signed-rank non-parametric statistical test with a p-value of 10^{-2} by using a Bonferroni correction (lower is better). The number in brackets stands for the average indicator-value, rounded to 10^{-3} (lower is better). **Bold** ranking values correspond to the best-performing algorithm for the instance and the indicator under consideration.

ρ	n	$\texttt{SLS}_{\texttt{unif}}$		$\texttt{SLS}_{\texttt{dicho}}$		PLS		TP-LS		best-known [12]	
						hypervolume					
−0.5	1000	3	(0.009)	2	(0.006)	1	(0.002)	**0**	(0.000)	4	(0.031)
−0.2	1000	3	(0.008)	2	(0.006)	1	(0.003)	**0**	(0.000)	4	(0.023)
0.0	1000	3	(0.007)	2	(0.006)	1	(0.004)	**0**	(0.000)	4	(0.016)
0.2	1000	1	(0.005)	1	(0.005)	3	(0.006)	**0**	(0.001)	4	(0.008)
0.5	1000	2	(0.002)	3	(0.003)	4	(0.008)	**0**	(0.002)	**0**	(0.002)
−0.5	2000	3	(0.007)	2	(0.004)	1	(0.002)	**0**	(0.000)	4	(0.053)
−0.2	2000	3	(0.007)	2	(0.004)	1	(0.003)	**0**	(0.001)	4	(0.047)
0.0	2000	3	(0.007)	1	(0.005)	1	(0.005)	**0**	(0.001)	4	(0.041)
0.2	2000	2	(0.005)	1	(0.004)	3	(0.006)	**0**	(0.001)	4	(0.023)
0.5	2000	1	(0.003)	1	(0.003)	4	(0.010)	**0**	(0.002)	3	(0.006)
−0.5	3000	3	(0.007)	2	(0.003)	1	(0.002)	**0**	(0.000)	4	(0.083)
−0.2	3000	3	(0.007)	1	(0.003)	1	(0.003)	**0**	(0.001)	4	(0.068)
0.0	3000	3	(0.007)	1	(0.004)	2	(0.006)	**0**	(0.001)	4	(0.062)
0.2	3000	2	(0.006)	1	(0.004)	3	(0.007)	**0**	(0.001)	4	(0.037)
0.5	3000	2	(0.003)	1	(0.002)	3	(0.010)	**0**	(0.001)	3	(0.010)
−0.5	4000	3	(0.007)	1	(0.003)	2	(0.006)	**0**	(0.000)	4	(0.092)
−0.2	4000	3	(0.007)	1	(0.003)	2	(0.004)	**0**	(0.001)	4	(0.077)
0.0	4000	3	(0.007)	1	(0.003)	2	(0.005)	**0**	(0.001)	4	(0.093)
0.2	4000	2	(0.004)	1	(0.002)	3	(0.006)	**0**	(0.001)	4	(0.047)
0.5	4000	2	(0.003)	1	(0.002)	3	(0.008)	**0**	(0.001)	4	(0.014)
−0.5	5000	2	(0.007)	1	(0.002)	3	(0.020)	**0**	(0.000)	4	(0.141)
−0.2	5000	2	(0.007)	1	(0.003)	3	(0.008)	**0**	(0.001)	4	(0.130)
0.0	5000	3	(0.006)	1	(0.003)	2	(0.006)	**0**	(0.001)	4	(0.130)
0.2	5000	2	(0.005)	1	(0.003)	3	(0.007)	**0**	(0.001)	4	(0.094)
0.5	5000	2	(0.003)	1	(0.002)	3	(0.010)	**0**	(0.001)	4	(0.021)
						epsilon					
−0.5	1000	3	(1.013)	2	(1.009)	1	(1.003)	**0**	(1.001)	4	(1.015)
−0.2	1000	2	(1.011)	2	(1.009)	1	(1.004)	**0**	(1.001)	4	(1.014)
0.0	1000	2	(1.010)	2	(1.010)	1	(1.005)	**0**	(1.001)	2	(1.010)
+0.2	1000	2	(1.009)	3	(1.009)	1	(1.005)	**0**	(1.001)	2	(1.008)
+0.5	1000	3	(1.011)	3	(1.012)	2	(1.008)	**0**	(1.002)	1	(1.005)
−0.5	2000	3	(1.009)	2	(1.007)	1	(1.003)	**0**	(1.001)	4	(1.026)
−0.2	2000	2	(1.008)	2	(1.008)	1	(1.003)	**0**	(1.001)	4	(1.027)
0.0	2000	3	(1.009)	2	(1.007)	1	(1.005)	**0**	(1.001)	4	(1.025)
+0.2	2000	3	(1.009)	2	(1.008)	1	(1.004)	**0**	(1.001)	4	(1.019)
+0.5	2000	2	(1.011)	1	(1.009)	1	(1.008)	**0**	(1.002)	4	(1.014)
−0.5	3000	3	(1.009)	2	(1.005)	1	(1.003)	**0**	(1.000)	4	(1.051)
−0.2	3000	3	(1.009)	2	(1.006)	1	(1.003)	**0**	(1.001)	4	(1.039)
0.0	3000	3	(1.008)	1	(1.006)	1	(1.005)	**0**	(1.001)	4	(1.034)
+0.2	3000	3	(1.007)	2	(1.006)	1	(1.004)	**0**	(1.001)	4	(1.025)
+0.5	3000	2	(1.005)	1	(1.004)	1	(1.004)	**0**	(1.001)	4	(1.011)
−0.5	4000	3	(1.008)	1	(1.004)	2	(1.007)	**0**	(1.000)	4	(1.055)
−0.2	4000	3	(1.008)	2	(1.005)	1	(1.003)	**0**	(1.001)	4	(1.042)
0.0	4000	3	(1.008)	2	(1.005)	1	(1.004)	**0**	(1.001)	4	(1.059)
+0.2	4000	3	(1.006)	2	(1.004)	1	(1.003)	**0**	(1.001)	4	(1.033)
+0.5	4000	3	(1.005)	1	(1.003)	1	(1.003)	**0**	(1.001)	4	(1.020)
−0.5	5000	2	(1.008)	1	(1.004)	3	(1.021)	**0**	(1.000)	4	(1.074)
−0.2	5000	2	(1.008)	1	(1.004)	2	(1.007)	**0**	(1.001)	4	(1.090)
0.0	5000	3	(1.008)	1	(1.005)	1	(1.004)	**0**	(1.001)	4	(1.064)
+0.2	5000	3	(1.007)	1	(1.004)	1	(1.004)	**0**	(1.001)	4	(1.050)
+0.5	5000	2	(1.004)	1	(1.003)	2	(1.004)	**0**	(1.001)	4	(1.025)

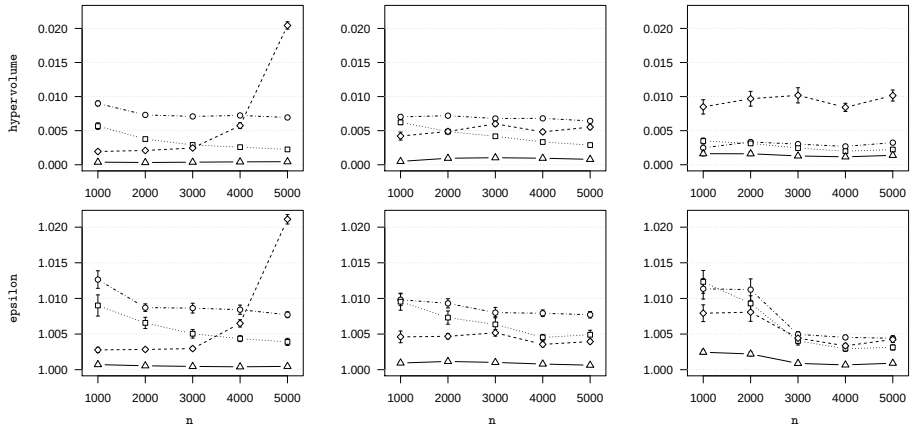

Fig. 2. Comparison of $\texttt{SLS}_{\texttt{unif}}$ (○), $\texttt{SLS}_{\texttt{dicho}}$ (□), PLS (◇) and TP-LS (△) with respect to hypervolume (top) and epsilon (bottom) for $\rho = -0.5$ (left), $\rho = 0.0$ (center), and $\rho = +0.5$ (right). A lower value is better.

tion ρ, as the proportion of supported solutions; see Fig. 1(c). By comparing $\texttt{SLS}_{\texttt{unif}}$ with $\texttt{SLS}_{\texttt{dicho}}$, the later is always at least as good as the former for all the instances we investigated but one ($\rho = +0.5$, $n = 1000$). The reason is that the $\texttt{SLS}_{\texttt{unif}}$ algorithm is limited on the approximation set size that it is able to identify (a fixed number of 101 weighting coefficient vectors times two phases, *i.e.* 202 solutions at most); see Fig. 3. On the contrary, $\texttt{SLS}_{\texttt{dicho}}$ adaptively determines a number of weighting coefficient vectors based on the solutions it iteratively identifies. As a consequence, it takes advantage of manipulating an unbounded approximation set, that allows $\texttt{SLS}_{\texttt{dicho}}$ to obtain better indicator-values overall. Still, as shown in Fig. 3, the number of solutions identified by both scalarizing approaches, which only seek for supported solutions, are lower than Pareto-based approaches by several orders of magnitude. However, the number of solutions found by all approaches reduces with the objective correlation ρ, as the number of Pareto optimal solutions reported in Section 3.1.

Overall, in terms of approximation quality, there is a clear advantage to TP-LS. Actually, hybridizing $\texttt{SLS}_{\texttt{dicho}}$ and PLS allows to obtain statistically better approximation sets, in terms of hypervolume and epsilon, than all the other competing algorithms, for all the instances. In particular, there is a substantial improvement of the indicator-values, showing that TP-LS is consistently able to identify a high-quality approximation set, which is very close to the reference set, especially for instances with conflicting objectives. Indeed, more than 99.67% of the best-found hypervolume is covered by the approximation set identified by TP-LS in the worst case. The epsilon indicator-value is always less than 1.003. The weakness of PLS in identifying good-quality lexicographical solutions seems to be overcome by initializing the archive with high-quality scalarizing solutions. This means that finding some non-dominated solutions can actually help to identify additional ones, as conjectured in Section 3.3.

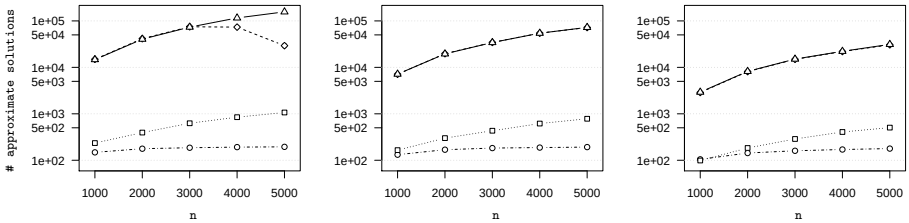

Fig. 3. Comparison of $\texttt{SLS}_{\texttt{unif}}$ (○), $\texttt{SLS}_{\texttt{dicho}}$ (□), PLS (◇) and TP-LS (△) with respect to the size of the approximation set found for $\rho = -0.5$ (left), $\rho = 0.0$ (center), and $\rho = +0.5$ (right). Notice the log-scale on the y-axis.

More surprisingly, the hybrid TP-LS approach also allows to improve the performance of PLS in terms of computational time. As reported in Fig. 4, the running time of TP-LS is below the one of PLS for most of the instances. By analyzing this more carefully, TP-LS actually allows to generate much more candidate solutions, and as many potentially non-dominated solutions as PLS, by performing much less pairwise comparisons, particularly for large-size instances. As well, the overhead of TP-LS compared to only performing the first phase as in $\texttt{SLS}_{\texttt{dicho}}$ is almost insignificant. At last, Fig. 4 also reveals that the loss of $\texttt{SLS}_{\texttt{unif}}$ in terms of quality is in fact counter-balanced by a very short computing time, which is lower than all other algorithms by several orders of magnitude. Actually, one single run of $\texttt{SLS}_{\texttt{unif}}$ allows to improve the *aggregated* best-known results from [12] by running faster than each single run of the algorithms from [12].

6 Conclusions

In this paper, we designed and analyzed stochastic local search algorithms for identifying a Pareto set approximation in bi-objective unconstrained binary quadratic programming. First, we designed a local search approach that iteratively identifies an approximate solution for multiple scalarizing sub-problems, such that the whole region of the Pareto front is covered in the objective space. The resulting $\texttt{SLS}_{\texttt{unif}}$ algorithm is based on a weighted-sum aggregation function, on a set of uniformly defined weighting coefficient vectors, and on a state-of-the-art tabu search procedure for the single-objective version of the problem under consideration. The scalarizing sub-problems are solved in sequence, such that the solution found at a given iteration is used to initialize the search process of the subsequent iteration. $\texttt{SLS}_{\texttt{unif}}$ allows to obtain a substantial improvement over the best-know solutions from previous studies with much less computations. Furthermore, the computational ressources required for instances with thousands of variables is less than a few minutes. However, the given number of weighting coefficient vectors severely limits the cardinality of the approximation set identified by $\texttt{SLS}_{\texttt{unif}}$. This user-defined parameter cannot be easily set *a priori* without performing an expensive experimental campaign. For this reason, we considered an improved scalarizing approach, based on the dichotomic search principles, that defines the weighting coefficient vector to be used in the current iteration based on solutions identified in previous iterations. As a consequence,

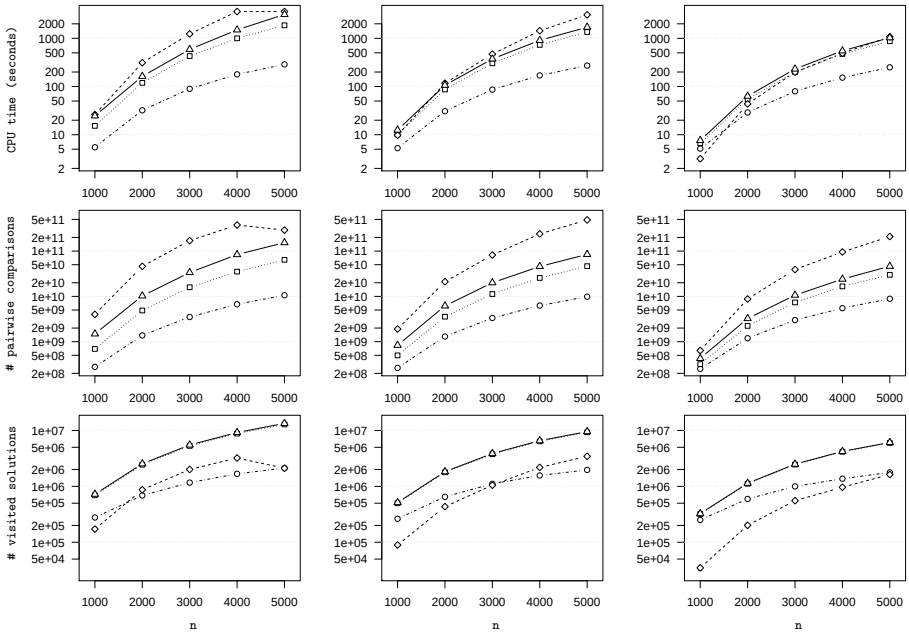

Fig. 4. Comparison of $\texttt{SLS}_{\texttt{unif}}$ (○), $\texttt{SLS}_{\texttt{dicho}}$ (□), PLS (◇) and TP-LS (△) with respect to to the CPU time (top), the number of comparisons (middle), and the number of visited solutions (bottom), for $\rho = -0.5$ (left), $\rho = 0.0$ (center), and $\rho = +0.5$ (right). Notice the log-scale on the y-axis.

there is *no* user-defined limit on the number of solutions manipulated by this search strategy. The designed $\texttt{SLS}_{\texttt{dicho}}$ algorithm allows to improve the results of $\texttt{SLS}_{\texttt{unif}}$ in terms of quality, while inducing an extra cost in terms of computing time. Still, for these two scalarizing approaches, that both seek *supported* optimal solutions only, the cardinality of the obtained approximation set is less than those of Pareto-based approaches. In addition, we highlighted that the performance of scalarizing local search decreases with the degree of conflict between the objectives, following the proportional number of supported solutions. Next, a Pareto local search algorithm, that iteratively explores the neighborhood of an archive of mutually non-dominated solutions, obtains similar results in terms of quality indicator-values, while requiring even more computational resources. Indeed, the PLS strategy obtains larger approximation sets while exploring much less candidate solutions. The bottleneck of its effectiveness is the number of comparisons required to maintain the archive, while the bottleneck of its efficiency is a poor quality in finding strong solutions at the extremes of the Pareto front. At last, we designed a two-phase algorithm that applies $\texttt{SLS}_{\texttt{dicho}}$ and PLS in a sequential way, the output of the former being the input of the latter. The TP-LS approach significantly surpasses the other algorithms over all the configurations we experimented. The computational overhead is negligible compared with the stand-alone $\texttt{SLS}_{\texttt{dicho}}$ approach, TP-LS being able to generate as many solutions as PLS while performing much less comparisons. This hybrid algorithm profits

from the closeness that exists between optimal solutions. Starting with a subset of approximate (supported) non-dominated solutions, `TP-LS` is able to identify additional ones, then improving the overall approximation set quality.

Extending the stochastic local search approaches investigated in the paper to unconstrained binary quadratic programming with more than two objectives would improve even more the expressive ability of the problem formulation. This would provide a more general unifying modeling and solution framework for multi-objective optimization that could potentially enable an efficient reformulation and resolution of a wide class of large-scale and NP-hard multi-objective combinatorial optimization problems.

References

1. Aneja, Y., Nair, K.: Bicriteria transportation problem. Manag Sci 25(1) (1979)
2. Bader, J., Zitzler, E.: HypE: An algorithm for fast hypervolume-based many-objective optimization. Evol Comp 19(1), 45–76 (2011)
3. Beasley, J.E.: OR-library: Distributing test problems by electronic mail. J Oper Res Soc 41(11), 1069–1072 (1990)
4. Dubois-Lacoste, J., López-Ibáñez, M., Stützle, T.: A hybrid TP+PLS algorithm for bi-objective flow-shop scheduling problems. Comp Oper Res 38(8), 1219–1236 (2011)
5. Dubois-Lacoste, J., López-Ibáñez, M., Stützle, T.: Improving the anytime behavior of two-phase local search. Ann Math Artif Intel 61(2), 125–154 (2011)
6. Ehrgott, M.: Multicriteria optimization. Springer, second edn. (2005)
7. Ehrgott, M., Klamroth, K.: Connectedness of efficient solutions in multiple criteria combinatorial optimization. Eur J Oper Res 97(1), 159–166 (1997)
8. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman & Co Ltd (1979)
9. Glover, F., Hao, J.K.: Efficient evaluations for solving large 0-1 unconstrained quadratic optimisation problems. Int J Metaheuristics 1(1), 3–10 (2010)
10. Glover, F., Lü, Z., Hao, J.K.: Diversification-driven tabu search for unconstrained binary quadratic problems. 4OR-Q J Oper Res 8(3), 239–253 (2010)
11. Kochenberger, G., Hao, J.K., Glover, F., Lewis, M., Lü, Z., Wang, H., Wang, Y.: The unconstrained binary quadratic programming problem: a survey. J Comb Optim 28(1), 58–81 (2014)
12. Liefooghe, A., Verel, S., Hao, J.K.: A hybrid metaheuristic for multiobjective unconstrained binary quadratic programming. Appl Soft Comp 16, 10–19 (2014)
13. Lust, T., Teghem, J.: Two-phase Pareto local search for the biobjective traveling salesman problem. J Heuristics 16(3), 475–510 (2010)
14. Paquete, L., Schiavinotto, T., Stützle, T.: On local optima in multiobjective combinatorial optimization problems. Ann Oper Res 156(1), 83–97 (2007)
15. Paquete, L., Stützle, T.: A study of stochastic local search algorithms for the biobjective QAP with correlated flow matrices. Eur J Oper Res 169(3), 943–959 (2006)
16. Paquete, L., Stützle, T.: Clusters of non-dominated solutions in multiobjective combinatorial optimization: An experimental analysis. In: Multiobjective Programming and Goal Programming, LNEMS, vol. 618, pp. 69–77. Springer (2009)
17. Zitzler, E., Thiele, L., Laumanns, M., Foneseca, C.M., Grunert da Fonseca, V.: Performance assessment of multiobjective optimizers: An analysis and review. IEEE Trans Evol Comput 7(2), 117–132 (2003)