A Hybrid Metaheuristic for Multiobjective Unconstrained Binary Quadratic Programming
Arnaud Liefooghe, Sébastien Verel, Jin-Kao Hao

To cite this version:
Arnaud Liefooghe, Sébastien Verel, Jin-Kao Hao. A Hybrid Metaheuristic for Multiobjective Unconstrained Binary Quadratic Programming. 2013. <hal-00801793v2>

HAL Id: hal-00801793
https://hal.archives-ouvertes.fr/hal-00801793v2
Submitted on 3 Jun 2013 (v2), last revised 15 Nov 2013 (v3)

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.
A Hybrid Metaheuristic for Multiobjective Unconstrained Binary Quadratic Programming

Arnaud Liefooghe\textsuperscript{a,b}, Sébastien Verel\textsuperscript{b,c}, Jin-Kao Hao\textsuperscript{d}

\textsuperscript{a}LIFL, Université Lille 1, UMR CNRS 8022, Cité scientifique, Bât. M3, 59655 Villeneuve d’Ascq cedex, France
\textsuperscript{b}Inria Lille-Nord Europe, Parc Scientifique de la Haute Borne, 40 avenue Halley, 59650 Villeneuve d’Ascq, France
\textsuperscript{c}LERIA, Université d’Angers, 2 bd. Lavoisier, 49045 Angers, France
\textsuperscript{d}I3S, Université Nice Sophia Antipolis, UMR CNRS 6070, 2000 route des Lucioles, BP 121, 06903 Sophia Antipolis cedex, France

Abstract

The conventional Unconstrained Binary Quadratic Programming (UBQP) problem is known to be a unified modeling and solution framework for many combinatorial optimization problems. This paper extends the single-objective UBQP to the multiobjective case (mUBQP) where multiple objectives are to be optimized simultaneously. We propose a hybrid metaheuristic which combines an elitist evolutionary multiobjective optimization algorithm and a state-of-the-art single-objective tabu search procedure by using an achievement scalarizing function. Finally, we define a formal model to generate mUBQP instances and validate the performance of the proposed approach in obtaining competitive results on large-size mUBQP instances with two and three objectives.

Key words: Unconstrained binary quadratic programming, Multiobjective combinatorial optimization, Hybrid Metaheuristic, Evolutionary Multiobjective Optimization, Tabu search, Scalarizing function

1. Introduction

Given a collection of \( n \) items such that each pair is associated with a profit value that can be positive, negative or zero, unconstrained binary quadratic programming (UBQP) seeks a subset of items that maximizes the sum of their paired values. The value of a pair is accumulated in the sum only if the two corresponding items are selected. A candidate solution to a UBQP instance can be specified by a binary string of size \( n \), such that each variable indicates whether the corresponding item is included in the selection or not. More formally, the conventional and single-objective UBQP problem is to maximize the following objective function.

\[
f(x) = x'Qx = \sum_{i=1}^{n} \sum_{j=1}^{n} q_{ij}x_{i}x_{j}
\]

where \( Q = (q_{ij}) \) is an \( n \) by \( n \) matrix of constant values and \( x \) is a vector of \( n \) binary (zero-one) variables, \( i.e., \ x_i \in \{0, 1\}, \ i \in \{1, \ldots, n\}. \)

The UBQP is known to be a general model able to represent a wide range of important problems, including those from financial analysis [1], social psychology [2], computer-aided design [3] and cellular radio channel allocation [4]. Moreover, a number of NP-hard problems can be conveniently transformed into the UBQP, such as graph coloring problems, maximum cut problem, set packing problem, set partitioning problem, maximum clique problem and so on [5, 6]. As a consequence, the UBQP itself is clearly a NP-hard problem [7]. During the past few decades, a large number of algorithms and approaches have been proposed for the single-objective UBQP in the literature. This includes several exact methods based on branch and bound or branch and cut [8, 9, 10] and a number of heuristic and metaheuristic methods like simulated annealing [11], tabu search [12, 13, 14, 15, 16], path-relinking [17], evolutionary and memetic algorithms [18, 19, 20, 21].

In this paper, we extend this conventional single-objective UBQP problem to the multiobjective case, denoted by mUBQP, where multiple objectives are to be optimized simultaneously. Such an extension naturally increases the expressive ability of the UBQP and provides a convenient formulation to fit situations where the single-objective UBQP cannot accommodate. For instance, UBQP can recast the vertex coloring problem (of determining the chromatic number of a graph) [5] and the sum coloring problem (of determining the chromatic sum of a graph) [22]. Still, UBQP is not convenient to formulate the bi-objective coloring problem which requires to determine a legal vertex coloring of a graph while minimizing simultaneously the number of colors used and the sum of colors. For this bi-objective coloring problem, the mUBQP formulation can be employed in a straightforward way.

In addition of introducing the mUBQP problem, the paper has two additional contributions. First, given that the single-objective UBQP is NP-hard, its generalized mUBQP formulation is also a difficult problem to solve in the general case. For the purpose of approximating the Pareto set of a given mUBQP instance, heuristic approaches are then compulsory. Following the studies on memetic algorithms for the UBQP and many other problems, we adopt as our solution approach the memetic framework and propose a hybrid metaheuristic which combines...
an elitist evolutionary multiobjective optimization algorithm with a state-of-the-art single-objective tabu search procedure based on an achievement scalarizing function. The last contribution of this work is to define a formal and flexible model to generate hard mUBQP instances. An experimental analysis validates the interest of the proposed hybrid metaheuristic by achieving a clear improvement over non-hybrid and conventional algorithms on large-size mUBQP instances with two and three objectives.

The paper is organized as follows. Section 2 introduces the multiobjective formulation of the UBQP problem (mUBQP). Section 3 presents the hybrid metaheuristic (HM) proposed for the mUBQP and its main ingredients, including the scalarizing evaluation function, the tabu search procedure, the initialization phase and the variation operators. Section 4 gives an experimental analysis of the HM algorithm on a large set of mUBQP instances of different structure and size. The last section concludes and suggests further research lines.

2. Multiobjective Unconstrained Binary Quadratic Programming

This section first introduces the multiobjective unconstrained binary quadratic programming problem. Some definitions related to multiobjective combinatorial optimization are then recalled, followed by problem complexity-related properties and a link with similar problem formulations. Last, the construction of problem instances, together with an experimental study on the objective values correlation and the cardinality of the Pareto set, are presented.

2.1. Problem Formulation

The multiobjective unconstrained binary quadratic programming (mUBQP) problem can be stated as follows.

\[
\begin{align*}
& \max \ f_k(x) = \sum_{i=1}^{n} \sum_{j=1}^{m} q_{ij}^k x_i x_j \quad k \in \{1, \ldots, m\} \\
& \text{subject to} \quad x_i \in \{0, 1\} \quad i \in \{1, \ldots, n\}
\end{align*}
\]

(2)

where \( f = (f_1, f_2, \ldots, f_m) \) is an objective function with \( m \geq 2 \), \( n \) is the problem size, and we have \( m \) matrices \( Q^k = (q_{ij}^k) \) of size \( n \times n \) with constant values, \( k \in \{1, \ldots, m\} \). The decision space \( X \) is defined on binary strings of size \( n \).

2.2. Definitions

Let \( X = [0, 1]^n \) be the set of feasible solutions in the decision space of Problem (2). We denote by \( Z \subseteq \mathbb{R}^n \) the feasible region in the objective space, i.e., the image of feasible solutions when using the vector maximizing function \( f \). The Pareto dominance relation is defined as follows. A solution \( x \in X \) is dominated by a solution \( x' \in X \), denoted by \( x \preceq x' \), if \( f_k(x) \leq f_k(x') \) for all \( k \in \{1, \ldots, m\} \), with at least one strict inequality. If neither \( x \not\preceq x' \) nor \( x' \not\preceq x \) holds, then both are mutually non-dominated. A solution \( x \in X \) is Pareto optimal (or efficient, non-dominated) if there does not exist any other solution \( x' \in X \) such that \( x' \) dominates \( x \). The set of all Pareto optimal solutions is called the Pareto set, denoted by \( X_{PSS} \), and its mapping in the objective space is called the Pareto front. One of the most challenging issues in multiobjective combinatorial optimization is to identify a minimal complete Pareto set, i.e., one Pareto optimal solution mapping to each point from the Pareto front. Note that such a set may not be unique, since multiple solutions can map to the same non-dominated vector.

2.3. Properties

For many multiobjective combinatorial optimization problems, computing the Pareto set is computationally prohibitive for two main reasons. First, the question of deciding if a candidate solution is dominated is known to be NP-hard for numerous multiobjective combinatorial optimization problems [23, 24]. This is also the case for the mUBQP problem since its single-objective counterpart is NP-hard [7]. Second, the number of Pareto optimal solutions typically grows exponentially with the size of the problem instance [24]. In that sense, most multiobjective combinatorial optimization problems are said to be intractable. In the following, we prove that the mUBQP problem is intractable.

**Proposition 1.** The multiobjective unconstrained binary quadratic programming problem (2) is intractable, even for \( m = 2 \).

**Proof.** Consider the following bi-objective mUBQP instance.

\[
q_{ij}^1 = \begin{cases} 
2^{m(i-1)+j-1} & \quad \text{if } i \geq j \\
0 & \quad \text{if } i < j
\end{cases}, \quad \text{i, j \in \{1, \ldots, n\}}
\]

Let \( q_{ij}^2 = -q_{ij}^1 \) for all \( i, j \in \{1, \ldots, n\} \). As illustrated in Figure 1 for \( n = 3 \), it is obvious that all solutions are mutually non-dominated. Therefore, all feasible solutions are Pareto optimal, and \( |X_{PSS}| = |X| = 2^n \).

In order to cope with NP-hard and intractable multiobjective combinatorial optimization problems, researchers have developed approximate algorithms that identify a Pareto set approximation having both good convergence and distribution properties [25, 26]. To this end, metaheuristics in general, and evolutionary algorithms in particular, have received a growing interest since the late eighties [27].

2.4. Links with Existing Problem Formulations

The single-objective UBQP problem is of high interest in practice, since many existing combinatorial optimization problems can be formalized in terms of UBQP [5]. As a consequence, multiobjective versions of such problems can potentially be defined in terms of mUBQP. However, to the best of our knowledge, the UBQP problem has never been explicitly defined in the multiobjective formulation given in Eq. (2). Existing multiobjective formulations of classical combinatorial optimization problems with binary variables include multiobjective linear assignment problems [24, 28], multiobjective knapsack problems [29, 30], multiobjective maxcut problems [31], or multiobjective set covering and partitioning problems [28], just to mention a few. Nevertheless, the objective
functions of such formulations are linear, and not quadratic as in mUBQP. Still, they often contain additional constraints; typically the unimodularity of the constraint matrix for linear assignment, or the capacity constraint for knapsack. This means that many existing binary multiobjective combinatorial optimization problems can be formalized in terms of mUBQP by adapting and generalizing the techniques from [5] to the multiobjective case, whereas the opposite does not hold in general due to the quadratic nature of mUBQP. The mUBQP problem is also different from the multiobjective quadratic assignment problem (mQAP) [32, 33], which seeks an assignment of \( n \) objects to \( n \) locations under multiple flow matrices. The solution representation is then usually based on a permutation for mQAP whereas it is based on a binary string for mUBQP.

### 2.5. Problem Instances

We propose to define correlated mUBQP problem instances as follows. Each objective function is defined by means of a matrix \( Q^k \), \( k \in \{1, \ldots , m\} \). Based on the single-objective UBQP instances available in the OR-lib [34], non-null matrix integer values are randomly generated according to a uniform distribution in \([-100, +100] \). As in the single-objective case, the density \( d \) gives the proportion of non-null numbers in the matrix. In order to define matrices of a given density \( d \), we set \( q_{ij}^k = 0 \) for all \( k \in \{1, \ldots , m\} \) at the same time, following a Bernoulli distribution of parameter \( d \).

Moreover, we define a correlation between the data contained in the \( m \) matrices \( Q^k \), \( k \in \{1, \ldots , m\} \). The positive (respectively negative) data correlation decreases (respectively increases) the degree of conflict between the objective function values. For simplicity, we use the same correlation between all pairs of objective functions, given by a correlation coefficient \( \rho \). The generation of correlated data follows a multivariate uniform law of dimension \( m \) [35]. In order to validate the behavior of the objective correlation coefficient experimentally, we conduct an empirical study for \( n = 18 \) in order to enumerate the decision space exhaustively. Figure 2 reports the average value of the Spearman correlation coefficient over 30 different and independent instances for different parameter combinations: \( \rho \), \( m \), and \( d \). Clearly, the correlation coefficient \( \rho \) tunes the objective correlation with a high accuracy.

To summarize, the four parameters used to define a mUBQP instance are (i) the problem size \( n \), (ii) the matrix density \( d \), (iii) the number of objective functions \( m \), and (iv) the objective correlation coefficient \( \rho \). The mUBQP problem instances used in the paper and an instance generator are available at the following URL: http://mocobench.sf.net/.

### 2.6. Cardinality of the Pareto Set

In this section, we analyze the impact of the mUBQP problem instance features (in particular, \( d \), \( m \) and \( \rho \)) on the number of Pareto optimal solutions. The Pareto set cardinality plays an important role on the problem complexity (in terms of intractability), and then on the behavior and the performance of solution approaches. Indeed, the higher the number of Pareto optimal solutions, the more computational resources required to identify a minimal complete Pareto set.

We set \( n = 18 \) in order to enumerate the decision space exhaustively. We report the average values over 30 different and independent mUBQP instances of same structure. Figure 3 gives the proportion of Pareto optimal solutions. Unsurprisingly, the matrix density \( d \) has a low influence on the results. However, the number of objective functions \( m \) and the objective correlation \( \rho \) both modify the proportion of Pareto optimal solutions to several orders of magnitude. Indeed, this proportion decreases from \( 10^{-4} \) for \( \rho = -0.9 \) to \( 10^{-3} \) for \( \rho = +0.9 \) for two- and three-objective mUBQP problem instances. As well, for a negative objective correlation \( \rho = -0.2 \), this proportion goes from \( 10^{-4} \) up to \( 10^{-1} \), whereas it goes from \( 10^{-5} \) up to \( 10^{-3} \) for a positive objective correlation \( \rho = +0.9 \), for \( m = 2 \) and \( m = 5 \), respectively. Figure 4 shows three examples of mUBQP problem instances represented in a two-objective space. When the objective correlation is negative, the objective functions are in conflict, and the Pareto set is large (left). When the objective correlation is null, the objective space can be embedded in a...
Figure 2: Average value of the Spearman correlation coefficient between the objective function values and the correlation coefficient $\rho$. The decision space is enumerated exhaustively for $n=18$ on a set of 30 independent random instances. The number of objectives is $m=2$ (left) and $m=3$ (right).

Figure 3: Average ratio of the minimal complete Pareto set cardinality ($|X_{PS}|$) to the decision space size ($|X|=2^{18}$) according to the objective correlation $\rho$ (top left $m=2$, right $m=3$), and according to the Q-matrix density $d$ (bottom left $\rho=-0.2$, right $\rho=0.9$). The problem size is $n=18$. Notice the log-scale.
multidimensional ball (middle). Last, when the objective correlation is positive, there exist few solutions in the Pareto set (right).

3. A Hybrid Metaheuristic for mUBQP

The hybrid metaheuristic proposed for the mUBQP problem is based on a memetic algorithm framework [36], which is known to be an effective approach for discrete optimization [37, 38]. Our approach uses one of the best performing local search algorithms for single-objective UBQP as one of its main components [12, 13].

3.1. General Principles

Memetic algorithms are hybrid metaheuristics combining an evolutionary algorithm and a local search algorithm. Multiobjective memetic algorithms [39] seek an approximation of the Pareto set (not only a subset of it). A simple elitist multiobjective population-based evolutionary algorithm operates as the main metaheuristic, whereas an advanced single solution-based local search is used as an improvement operator in place of the mutation step. Keeping the exploration vs. exploitation trade-off in mind, the idea behind such an approach is that the evolutionary algorithm will offer more facilities for diversification, while the local search algorithm will provide more capabilities for intensification.

The search space is composed of all binary vectors of size $n$. The size of the search space is then equal to $2^n$. The evaluation function is the canonical objective function given in Eq. (2). An unbounded archive of mutually non-dominated solutions found so far is maintained with respect to the Pareto dominance relation defined in Section 2.2. Throughout the search process, solutions are discarded as soon as they are detected to be equivalent to, or dominated by, at least one other solution from the archive. At each iteration, two parents are selected at random from the archive and recombined to produce a single offspring solution (Section 3.5). The offspring solution is further improved by means of a tabu search algorithm (Section 3.3). The evaluation function used by tabu search is based on a scalarizing technique of the initial objective function values (Section 3.2). The corresponding achievement scalarizing function is defined in such a way that the tabu search procedure focuses its search within the objective space area enclosed by the positions of parent solutions. Another crucial component of the HM algorithm appears at the initial phase (Section 3.4), where a computational effort is made in order to identify close-to-optimal solutions located at the extreme regions of the objective space. The algorithm is iterated until a user-given stopping condition is satisfied. An outline of the hybrid metaheuristic (HM) is given in Algorithm 1. The main components of the HM algorithm are detailed below.

3.2. Achievement Scalarizing Function

The tabu search procedure, that will be presented later in the paper, is known to be well-performing for solving single-objective UBQP instances of different structures and sizes [12, 13, 14, 15, 16]. Of course, given that it manipulates a single solution only, a scalarization of the multiple objective functions is required due to the multiobjective nature of the mUBQP. The goal is to temporarily transform the mUBQP problem into a single-objective one so that the tabu search algorithm can be used in a straightforward way. Many general-purpose scalariz-
ing functions have been proposed for multiobjective optimization [40], generally with the aim of incorporating preference information coming from a decision-maker. The matter is here different since we are interested in approximating the whole Pareto set. Hence, the parameters required by the scalarizing function are dynamically set according to the current state of the search process. This will be discussed in Section 3.5.

In multiobjective memetic algorithms, the most popular scalarizing function is the weighted-sum aggregation [39, 41], where a weighting coefficient vector represents the relative importance of each objective function. However, this approach cannot identify a number of Pareto optimal solutions, whose corresponding non-dominated objective vectors are located on the convex hull of the Pareto front [24, 40]. Another example is the achievement scalarizing function, proposed by Wierzbicki [42]. This technique is based on a reference point. A reference point gives desirable or acceptable values for each objective function. These objective values are called aspiration levels and the resulting objective vector is called a reference point and can be defined either in the feasible or in the infeasible region of the objective space. One of the families of achievement scalarizing functions can be stated as follows. Let us recall that the maximization of the objective functions are assumed.

\[
\sigma_{(\zeta, \lambda, \epsilon)}(x) = \max_{k \in \{1, \ldots, m\}} \left\{ \lambda_k (z^*_k - f_k(x)) \right\} + \epsilon \sum_{k=1}^{m} \lambda_k (z^*_k - f_k(x)) \tag{3}
\]

where \(\sigma\) is a mapping function from \(X\) to \(\mathbb{R}\), \(x \in X\) is a feasible solution, \(\zeta^* \in \mathbb{R}^m\) is a reference point vector, \(\lambda \in \mathbb{R}^m\) is a weighting coefficient vector, and \(\epsilon\) is an arbitrary small positive number \((0 < \epsilon \ll 1)\). We keep the \(\epsilon\) parameter constant throughout the search process. The following achievement scalarizing optimization problem can be formalized.

\[
\min \quad \sigma_{(\zeta, \lambda, \epsilon)}(x) \tag{4}
\]

subject to \(x \in X\)

Interestingly, two properties are ensured [43]:

(i) if \(x^* = \arg \min_{x \in X} \sigma_{(\zeta, \lambda, \epsilon)}(x)\), then \(x^*\) is a Pareto optimal solution;

(ii) if \(x^*\) is a Pareto optimal solution, then there exists a function \(\sigma_{(\zeta', \lambda)}\) such that \(x^*\) is a (global) optimum of Problem (4).

This makes the achievement scalarizing function attractive. Indeed, as noticed earlier, only a subset of Pareto optimal solutions, known as supported solutions [24], can be found within a weighted-sum aggregation function, since the second property (ii) is not satisfied. Those solutions are known as supported Pareto optimal solutions, and their corresponding non-dominated objective vectors are located on the convex hull of the Pareto front. On the contrary, the achievement scalarizing function potentially enables the identification of both supported and non-supported Pareto optimal solutions [40]. Successful integrations of the achievement scalarizing function into evolutionary multiobjective optimization algorithms can be found elsewhere [44, 45, 46]. However, in existing approaches, the parameters of the achievement scalarizing function are usually kept static or randomly chosen throughout the search process, whereas they are adapted to appropriate values according to the current state of the search process in the HM proposed in the paper, as will be detailed in Section 3.5.

3.3. Tabu Search

The following tabu search algorithm, used as a subroutine of the HM, is reported to be one of the best-performing approaches for the single-objective UBQP problem [13]. In order to extend it to the multiobjective case, we use the achievement scalarizing function, so that the initial objective vector values are transformed into a single scalar value. Notice, however, that the nature of the evaluation function considered in the paper has a different structure than the classical evaluation function of single-objective UBQP. We describe the main principles of the tabu search below.

The neighborhood structure is based on the 1-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-flip neighborhood structure is linear with the problem size \(n\). As in the single-objective UBQP, each mUBQP objective function can be evaluated incrementally. We follow the fast incremental evaluation procedure proposed by Glover and Hao [47] to calculate the move gain of a given neighboring solution. For a given objective function, the whole set of neighbors can be evaluated in linear time. As a consequence, the objective values of all neighboring solutions are evaluated in \(O(m \cdot n)\) in the multiobjective case. Once the objective values of a given neighboring solution have been (incrementally) evaluated, we compute its scalar fitness value with respect to Eq. (3).

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 Lü et al. [20], we set the tabu tenure of a given variable \(x_i\) after it has been flipped as follows.

\[
tenure(i) = tt + rand(10) \tag{5}
\]

where \(tt\) is a user-given parameter and \(rand(10)\) gives a random integer value between 1 and 10. From the set of neighboring solutions produced by all non-tabu moves, we select the one with the best (smallest) fitness value according to Eq. (3). Indeed, let us recall at this point that the aim of the tabu search algorithm is to find a good approximate solution for Problem (4), for a given definition of \(\zeta^*\) and \(\lambda\). However, all neighboring solutions 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. The stopping condition of the tabu search algorithm is met when no improvement
has been performed within a given number of moves \( \alpha \). The parameter \( \alpha \) is called the improvement cutoff. For more details on the tabu search algorithm for the single-objective UBQP, the reader is referred to Glover et al. [13, 20].

### 3.4. Initial Phase

The goal of the initial phase is to identify good-quality solutions with respect to each objective function of the mUBQP, i.e., solutions mapping to the extreme points of the Pareto front in the objective space. This set of solutions initializes the search process in order to ensure that the HM provides a good covering of the Pareto front. To this end, we define the following achievement scalarizing function parameter setting. We set the reference point \( \zeta' = \{z_{k,1}^{\text{max}}, \ldots, z_{k,m}^{\text{max}}\} \) such that \( z_{k,1}^{\text{max}} \) is higher than any possible \( f_k \)-value. This (unfeasible) objective vector can be seen as a rough approximation of the utopia point [24, 40]. Now, let us consider a particular objective function \( k \in \{1, \ldots, m\} \). We set \( \lambda_k = 1 \), and \( \lambda_l = 0 \) for all \( l \in \{1, \ldots, m\} \setminus \{k\} \).

The tabu search algorithm, seeded with a random solution, is then considered within the corresponding achievement scalarizing function as an evaluation function. Those initial solutions have a high impact on the performance of the HM, particularly in terms of diversification. As a consequence, we perform \( \gamma \) independent restarts of the tabu search per objective function in order to increase the chance of getting high-performing solutions in all extreme regions of the Pareto front. This process is iterated for every objective function of the mUBQP problem instance under consideration.

### 3.5. Variation Operator

At each iteration of the HM algorithm, a single offspring solution is created by a recombination operator. First, we select two mutually non-dominated parent solutions at random from the current archive \( x_i, x_j \in A \) such that \( x_i \neq x_j \). Then, an offspring solution is created with uniform crossover. Common variables between both parents are thus assigned to the offspring solution, while the remaining ones are assigned at random. The offspring solution is further improved by means of the tabu search procedure presented in Section 3.3. We aim at obtaining a new solution in an unexplored region of the Pareto front by defining the parameters of the achievement scalarizing function properly. The procedure attempts to find a non-dominated point that “fills the gap” between the objective vectors associated with \( x_i \) and \( x_j \). The region of the objective space where the tabu search algorithm operates is then delimited by the position of parent solutions, given by the following definition of the achievement scalarizing function.

\[
\begin{align*}
\zeta_k' & = \max(f_k(x_i), f_k(x_j)) \quad k \in \{1, \ldots, m\} \quad (6) \\
\lambda_k & = \frac{1}{|f_k(x_i) - f_k(x_j)|} \quad k \in \{1, \ldots, m\} \quad (7)
\end{align*}
\]

This procedure allows the HM to improve, at each iteration, a particular part of the Pareto front, dynamically chosen with respect to the pair of parent solutions under selection. The overall variation procedure is illustrated in Figure 5.

### 4. Experimental Analysis

This section presents an experimental analysis of the proposed approach on a broad range of mUBQP problem instances.

#### 4.1. Experimental Design

We conduct an experimental study on the influence of the problem size \( n \), the number of objectives \( m \), and the objective correlation \( \rho \) of the mUBQP problem on the performance of the HM algorithm proposed in the paper. In particular, we investigate the following parameter setting: \( n \in \{1000, 2000, 3000, 4000, 5000\} \), \( m \in \{2, 3\} \), and \( \rho \in \{-0.5, -0.2, 0.0, +0.2, +0.5\} \). 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 50 problem instances.

We compare the performance of our algorithm against a steady-state evolutionary algorithm that follows the same structure as the HM, but where the tabu search is replaced by a random mutation. This allows us to appreciate the impact of the tabu search and scalarizing procedure. The same initialization phase than in the HM is applied. Then, at each iteration, an offspring solution is created by uniform crossover and an independent bit-flip operator is applied, i.e., each variable is randomly flipped with a probability \( 1/n \). We refer to this algorithm as SSEA, for steady-state evolutionary algorithm. We also compare the results of the algorithms to a baseline algorithm, the well-known NSGA-II [48]. NSGA-II maintains a population of constant size, initialized at random, and produces the same number of offspring solutions at each iteration. Selection for reproduction and replacement is based on dominance-depth ranking first, and on crowding distance at second-level. At each iteration, non-dominated solutions from the current population are first
Table 1: Parameter setting for the experimental analysis.

<table>
<thead>
<tr>
<th>Description</th>
<th>Parameter</th>
<th>Value(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Problem size</td>
<td>( n )</td>
<td>([1000, 2000, 3000, 4000, 5000])</td>
</tr>
<tr>
<td>Matrix density</td>
<td>( d )</td>
<td>0.8</td>
</tr>
<tr>
<td>Number of objectives</td>
<td>( m )</td>
<td>([2, 3])</td>
</tr>
<tr>
<td>Objective correlation</td>
<td>( \rho )</td>
<td>([-0.5, -0.2, 0.0, +0.2, +0.5])</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Crossover rate</td>
<td>( \pi )</td>
<td>1.0</td>
</tr>
<tr>
<td>Mutation rate (SS-EA, NSGA-II)</td>
<td>( \frac{1.0}{n} )</td>
<td>100</td>
</tr>
<tr>
<td>Population size (NSGA-II)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tabu tenure</td>
<td>( tt )</td>
<td>( n/150 )</td>
</tr>
<tr>
<td>Tabu improvement cutoff</td>
<td>( \alpha )</td>
<td>5(n)</td>
</tr>
<tr>
<td>Number of restarts (initialization)</td>
<td>( \gamma )</td>
<td>5</td>
</tr>
<tr>
<td>reference point</td>
<td>( \epsilon )</td>
<td>adaptively set during the search process; see Section 3.5</td>
</tr>
<tr>
<td>weighting coefficient vector</td>
<td>( \lambda )</td>
<td>adaptively set during the search process; see Section 3.5</td>
</tr>
<tr>
<td>( \epsilon )-parameter (achievement scalarizing function)</td>
<td>( \epsilon )</td>
<td>(10^{-8})</td>
</tr>
<tr>
<td>Stopping condition (CPU time)</td>
<td></td>
<td>((n \cdot m \cdot 10^{-3})) minutes</td>
</tr>
</tbody>
</table>

Algorithms

<table>
<thead>
<tr>
<th>Description</th>
<th>Parameter</th>
<th>Value(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

assigned a rank of 1 and are discarded from consideration, non-dominated solutions from the remaining solutions of the population are then assigned a rank of 2 and are discarded from consideration, and so on. This process is iterated until the set of solutions with no rank is empty. The crowding distance estimates the density around a particular objective vector. The crowding value is computed among solutions with the same rank. A solution is said to be better than another solution if the former has a better rank, or in the case of equality, if it is less crowded. A binary tournament is used for selection, and an elitist strategy is used for replacement. The same crossover and mutation operators as for SS-EA are considered. In other words, the main differences between SS-EA and NSGA-II are: (i) SS-EA uses an unbounded population whereas NSGA-II maintains a fixed-size population, (ii) selection for reproduction is performed at random within SS-EA whereas it is based on dominance-depth and crowding distance within NSGA-II, and (iii) the archive is initialized as detailed in Section 3.4 for SS-EA whereas the NSGA-II initial population is generated at random. However, an external unbounded archive has been added to the canonical NSGA-II in order to prevent the loss of non-dominated solutions. We did not experience any memory issues by maintaining the whole set of non-dominated solutions found during the search process with any of the competing algorithms.

All the algorithms stop after \((n \cdot m \cdot 10^{-3})\) minutes of CPU time, \(i.e.,\) from 2 minutes per run for smaller instances up to 15 minutes for large-size instances. Since neighboring solutions are evaluated incrementally within HM during the tabu search phases, a maximum number of evaluations cannot be used as a stopping condition. Following [20], the tabu tenure constant is set to \(tt = n/150\), and the improvement cutoff to \(a = 5n\). During the initialization phase, the number of random restarts per objective function is set to \(\gamma = 5\). Last, the \(\epsilon\)-parameter of the achievement scalarizing function is set to \(\epsilon = 10^{-3}\). The population size of NSGA-II is set to 100 solutions. A summary of all the parameters is given in Table 1.

HM, SS-EA and NSGA-II have been implemented within the ParadisEO software framework [49, 50]. All the algorithms have been executed under comparable conditions and share the same base components for a fair comparison. The experiments have been conducted on an Intel Core 2 quad-core processor (2.40 GHZ, 4GB RAM) running under Ubuntu 10.04. All codes were compiled with g++ 4.4.3 using the -03 compilation option.

4.2 Performance Assessment

A set of 30 runs per instance has been performed for each algorithm. In order to evaluate the quality of the approximations found for each instance, we follow the performance assessment protocol proposed by Knowles et al. [26]. Such a way of comparing multiple stochastic multiobjective optimizers is a common practice in the specialized literature. Let us consider a given mUBQP problem instance. Let \(Z^{\text{null}}\) be the set of objective vectors from all the Pareto set approximations we obtained during all our experiments. Note that \(Z^{\text{null}}\) may contain both dominated and non-dominated objective vectors, since a given approximation may contain points dominating the ones of another approximation, and vice versa. We define \(z_{\min} = (z_{\min}^{1}, \ldots, z_{\min}^{m})\) and \(z_{\max} = (z_{\max}^{1}, \ldots, z_{\max}^{m})\), where \(z_{\min}^{k}\) (respectively \(z_{\max}^{k}\)) denotes the lower (respectively upper) bound of the \(k^{\text{th}}\) objective for all the points contained in \(Z^{\text{null}}\), \(\forall k \in \{1, \ldots, m\}\). In order to give a roughly equal range to the objective functions, values are normalized between 1 and 2 with respect to \(z_{\min}\) and \(z_{\max}\). Then, we compute a reference set \(Z^*\) containing the non-dominated points of \(Z^{\text{null}}\). In order to measure the quality of Pareto front approximations, we use both the Pareto dominance relation extended to sets and the difference hypervolume indicator \((I_H)\) [25]. They are illustrated in Figure 6 and Figure 7, respectively. The Pareto dominance relation over sets can be defined as follows. A given Pareto front approximation \(A^1\) is dominated by another approximation \(A^2\), if for all objective vector \(z^1 \in A^1\), there exists an objective vector \(z^2 \in A^2\) such that \(z^1\) is dominated by \(z^2\). The \(I_H\)-indicator value of a given approximation \(A\) gives the portion of the objective space that is dominated by \(Z^*\) and not by \(A\), \(z^2 \in \{0.9, \ldots, 0.9\}\) being the reference point. Note that \(I_H\)-values are to be minimized. The
The contributions of the paper are three-fold. First, the unconstrained binary quadratic programming (UBQP) problem has been extended to the multiobjective case (mUBQP) which involves an arbitrary number of UBQP objective functions to be maximized simultaneously over the same decision space of binary strings of size $n$. In the single-objective case, the UBQP problem is one of the most challenging problem from combinatorial optimization, and is known to enable the formulation of a large number of practical applications in many areas. The multiobjective UBQP problem introduced in this paper will allow more practical applications to be formulated and solved.

Second, multiobjective UBQP problem instances and an instance generator have been made available at the following URL: http://mocobench.sf.net. These problem instances are characterized by a problem size, a matrix density, a number of objective functions, and a correlation coefficient between the objective values. In particular, the objective correlation can be tuned precisely, allowing one to study the impact of this feature on the size of the Pareto front, and then on the performance of solution approaches. These instances are useful for performance assessment and comparison of new algorithms for the general mUBQP problem.
Third, we have presented an hybrid evolutionary-tabu search algorithm for the multiobjective UBQP. The proposed approach integrates a state-of-the-art tabu search algorithm for the single-objective UBQP, together with Pareto-based evolutionary optimization principles. Based on the achievement scalarizing function, the proposed algorithm is able to generate both supported and unsupported solutions, with the aim of finding a well-converged and well-diversified Pareto set approximation. We have showed that this hybrid metaheuristic obtains significantly better results than two conventional evolutionary multi-objective optimization techniques for large-size multiobjective UBQP problem instances of different structure and size.

A better understanding of the main problem characteristics would allow us to improve the design of heuristic search algorithms by incorporating a deeper problem knowledge. To this end, we plan to study the correlation between the main problem features and the algorithm performance through fitness landscapes analysis in multiobjective combinatorial optimization [35, 51]. Last, we hope that the challenge proposed by multi-objective UBQP will gain the attention of other researchers. In particular, a stronger link is required between multiobjective UBQP formulations and existing combinatorial optimization problems like the multiobjective variants of assignment, covering, partitioning, packing and quadratic knapsack problems. This would enable the identification of a Pareto front approximation for many problems from multiobjective combinatorial optimization under a unified modeling, either as a stand-alone methodology, or to provide a fast computation of a lower bound set for improving the performance of exact approaches.

References

P. M. Pardalos, G. P. Rodgers, Computational aspects of a branch and bound algorithm for quadratic zero-one programming, Computing 45 (2) (1990) 131–144.


Y. Wang, Z. Liu, F. Glover, J. Hao, Solving the minimum sum coloring problem via binary quadratic programming, submitted.


