Combining Local Search and Elicitation for Multi-Objective Combinatorial Optimization

Nawal Benabbou, Cassandre Leroy, Thibaut Lust, Patrice Perny

▶ **To cite this version:**

Nawal Benabbou, Cassandre Leroy, Thibaut Lust, Patrice Perny. Combining Local Search and Elicitation for Multi-Objective Combinatorial Optimization. ADT 2019 - 6th International Conference on Algorithmic Decision Theory, Oct 2019, Durham, NC, United States. hal-02170910

HAL Id: hal-02170910

https://hal.sorbonne-universite.fr/hal-02170910v1

Submitted on 2 Jul 2019

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.

Combining Local Search and Elicitation for Multi-Objective Combinatorial Optimization

Nawal Benabbou, Cassandre Leroy, Thibaut Lust, and Patrice Perny

Sorbonne Université, CNRS, LIP6, F-75005 Paris, France
firstname.lastname@lip6.fr

Abstract. In this paper, we propose a general approach based on local search and incremental preference elicitation for solving multi-objective combinatorial optimization problems with imprecise preferences. We assume that the decision maker's preferences over solutions can be represented by a parameterized scalarizing function but the parameters are initially not known. In our approach, the parameter imprecision is progressively reduced by iteratively asking preference queries to the decision maker 1) before the local search in order to identify a promising starting solution and 2) during the local search but only when preference information are needed to discriminate between the solutions within a neighborhood. This new approach is general in the sense that it can be applied to any multi-objective combinatorial optimization problem provided that the scalarizing function is linear in its parameters (e.g., a weighted sum, an OWA aggregator, a Choquet integral) and that a (near-)optimal solution can be efficiently determined when preferences are precisely known. For the multi-objective traveling salesman problem, we provide numerical results obtained with different query generation strategies to show the practical efficiency of our approach in terms of number of queries, computation time and gap to optimality.

Keywords: Multi-objective combinatorial optimization, local search, preference elicitation, minimax regret, traveling salesman problem.

1 Introduction

Designing efficient preference elicitation procedures to support decision making in combinatorial domains is one of the hot topics of algorithmic decision theory. On non-combinatorial domains, various model-based approaches are already available for preference learning. The elicitation process consists in analyzing preference statements provided by the decision maker (DM) to assess the parameters of the decision model and determine an optimal solution (see, e.g., [5, 7, 9, 10, 14, 31]). Within this stream of work, incremental approaches are of special interest because they aim to analyze the set feasible solutions to identify the critical preference information needed to find the optimal alternative. By a careful selection of preference queries, they make it possible to determine the optimal choice within large sets, using a reasonably small number of questions, see e.g.,

[10] for an example in a Bayesian setting, and [7] for another approach based on a progressive reduction of the uncertainty attached to the model parameters.

The incremental approach was efficiently used in various decision contexts such as multiattribute utility theory or multicriteria decision making [8, 30, 34], decision making under risk [10, 17, 26, 32] and collective decision making [23]. However, extending these approaches for decision support on combinatorial domains is more challenging due to the implicit definition of the set of solutions and the huge number of feasible solutions. In order to overcome this problem, several contributions aim to combine standard search procedures used in combinatorial optimization with incremental preference elicitation. Examples can be found in various contexts such as constraint satisfaction [15], committee election [3], matching [13], sequential decision making under risk [28, 33], fair multiagent optimization [6] and multicriteria optimization [1, 19]. In multicriteria optimization, the search procedure combines the implicit enumeration of Pareto-optimal solutions with preferences queries allowing a progressive reduction of the uncertainty attached to the parameters of the preference aggregation model, in order to progressively focus the exploration on the most attractive solutions. Various attempts to interleave incremental preference elicitation methods and constructive algorithms have been proposed. The basic principle consists in constructing the optimal solution from optimal sub-solutions using the available preference information, and to ask new preference information when necessary. This has been tested for greedy algorithms, for dynamic programming, for A^* and branch-and-bound search, see [4] for a synthesis.

In this paper we explore another way by considering non-constructive algorithms. We propose to interleave the elicitation with local search for multicriteria optimization. To illustrate our purpose, let us consider the following example:

Example 1 *Let us consider the instance of the multi-objective traveling salesman problem (TSP) depicted in Figure 1, including 5 nodes and two additive cost functions to be minimized (one looks for a cycle passing exactly once through each node of the graph and minimizing costs). Let us start a local search from*

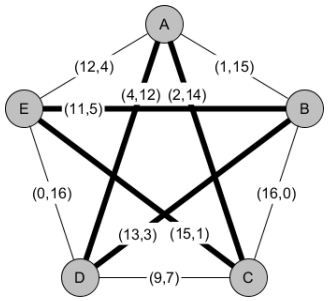

Fig. 1. An instance of the TSP with two criteria.

the boldface tour $x_0 = ADBECA$ *whose cost vector is* $(45, 35)$*, using a neighbor-*

hood definition based on the simple exchange of two consecutive nodes. Among the neighbors of x_0, there is $x'_0 = ABDECA$ whose cost vector is $(31, 49)$. Assume that the DM declares that x_0 is better than x'_0 (denoted $x_0 \succ x'_0$).

Let us show what could be a local search on this instance using partial preference information interpreted under a linear model assumption:

Using a linear model. We assume here that DM's preferences can be represented by a linear model of the form: $f_\omega(y_1, y_2) = \omega y_1 + (1 - \omega)y_2$ for some unknown $\omega \in (0, 1)$, where (y_1, y_2) is the cost vector associated to a feasible tour. In this case the $x_0 \succ x'_0$ condition implies $f_\omega(45, 35) < f_\omega(31, 49)$ and therefore $\omega \in (0, 1/2)$. After this restriction of the set of possible values for ω it can easily be checked that the optimal neighbor of x_0 is solution $x_1 = ADBCEA$ of cost $(60, 20)$. Then by exploring the neighborhood of x_1 it can easily be checked that no other solution can improve x_1 given that $\omega < 1/2$. We get a local optimum which is actually the optimal solution for this instance.

We can see here that, under the linear model assumption, an optimal solution has been obtained using a single preference query. However, the linear model is not always suitable. For example, when one looks for a compromise solution between the two criteria, one could prefer resorting to a decision model favoring the generation of solutions having a balanced cost vector. For this reason we consider now another elicitation session using a non-linear weighted aggregation function commonly used to control the balance between criterion values, namely the *Ordered Weighted Average* (OWA, [22, 35]).

Using the OWA model. Now, let us assume that the DM's preferences are represented by a non-linear model of the form: $f_\omega(y_1, y_2) = \omega \max\{y_1, y_2\} + (1 - \omega) \min\{y_1, y_2\}$ for some unknown $\omega \in (0, 1)$. In this case the $x_0 \succ x'_0$ condition implies $45\omega + 35(1 - \omega) < 49\omega + 31(1 - \omega)$ and therefore $\omega \in (1/2, 1)$ (note that although OWA is not linear in y, it is linear in ω and therefore any preference statement translates into a linear constraint on ω). After this restriction of the set of possible values for ω it can easily be checked that the optimal neighbor of x_0 is solution $x_2 = ABECDA$ whose cost vector is $(40, 40)$. Then, by exploring the neighborhood of x_2, it can easily be checked that no other solution can improve x_2 given that $\omega > 1/2$. We obtain a local optimum which is actually the OWA-optimal solution for this instance, given the restriction on ω.

These simple executions of local search using partial preference information show the potential of interactive local search combining local exploration of neighborhoods and model-based preference elicitation. For a given class of preference models, the successive answers from the DM to preference queries make it possible to progressively reduce the set of possible parameters and to discriminate the solutions belonging to the neighborhood of solutions found so far.

The combination of preference elicitation has several specific advantages. In particular, preference elicitation is based on very simple queries because they involve neighbor solutions that are cognitively simpler to compare than pairs of

solutions varying in all aspects. Moreover, preference queries only involve complete solutions. This provides two advantages: 1) solutions are easier to compare, and 2) no independence assumption is required in the definition of preferences (no need to reason on partial descriptions under the assumption that preferences hold everything all being equal). The latter point is of special interest when preferences are represented by *non-linear* decision models. With such models, the cost of partial solutions is a poor predictor of the actual cost of their extensions. This seriously reduces possibilities of pruning sub-solutions in constructive algorithms. Since we consider only complete solutions in local search, this problem vanishes. Another interest of studying incremental elicitation approaches in local search is to tackle preference-based combinatorial optimization problems for which no efficient exact algorithm is known.

The paper is organized as follows. Section 2 introduces some preliminary background and notations. Then, we present a general interactive local search in Section 3. In Section 4 we further specify our approach for application to the multi-objective TSP and provide numerical tests showing the practical efficiency of the proposed incremental elicitation process.

2 Background and Notations

In this section, we present the necessary background on multi-objective combinatorial optimization and regret-based incremental elicitation.

2.1 Multi-objective Combinatorial Optimization

We consider a multi-objective combinatorial optimization (MOCO) problem with n objectives/criteria $y_i, i \in \{1,\dots,n\}$, that need to be minimized. This problem can be formulated as follows: $\underset{x\in\mathcal{X}}{\text{minimize}}\big(y_1(x),\dots,y_n(x)\big)$ where $\mathcal{X}$ is the feasible set in the decision space (e.g., for the TSP, $\mathcal{X}$ is the set of all Hamiltonian cycles). In this problem, any solution $x \in \mathcal{X}$ is associated with a vector $y(x) = (y_1(x),\dots,y_n(x)) \in \mathbb{R}^n$ that gives its evaluations on all criteria. Solutions are usually compared through their images in the objective space (also called points) using the *Pareto dominance* relation: point $a = (a_1,\dots,a_n) \in \mathbb{R}^n$ is said to *Pareto dominate* point $b = (b_1,\dots,b_n) \in \mathbb{R}^n$ (denoted by $a \prec_P b$) if and only if $a_i \leq b_i$ for all $i \in \{1,\dots,n\}$ and $a_i < b_i$ for some $i \in \{1,\dots,n\}$. A solution $x \in \mathcal{X}$ is said to be *efficient* if there is no other feasible solution $x' \in \mathcal{X}$ such that $y(x') \prec_P y(x)$ and the set $\mathcal{X}_E$ of all efficient solutions is called the *efficient set* (their images are respectively called *non-dominated point* and *Pareto front*).

We assume here that the DM needs to select a single solution. Without any preference information, we only know that her preferred solution is an element of the efficient set. However, it is well-known that the number of efficient solutions (and the number of non-dominated points) can be exponential in the size of the problem (e.g., [18] for the multicriteria spanning tree problem); in such situations, identifying the Pareto front is not enough to help the DM in making a decision. One way to address this issue is to reduce the size of the Pareto front by constructing a "well-represented" set; for instance, this set can be obtained

by dividing the objective space into different regions (e.g., [20]) or by using some approximate dominance relation (e.g., ϵ-dominance [21]). However, in situations where the DM needs to select only one solution, it seems more appropriate to refine the Pareto dominance with preferences in order to determine the optimal solution according to the DM's subjective preferences.

In this work, we assume that the DM's subjective preferences can be represented by a parameterized scalarizing function f_ω that is linear in its preference parameters ω. For example, function f_ω can be a weighted sum (i.e., $f_\omega(a) = \sum_{i=1}^{n} \omega_i a_i$), an OWA aggregator ($f_\omega(a) = \sum_{i=1}^{n} \omega_i a_{(i)}$ where $a_{(i)} \geq \ldots \geq a_{(n)}$ are the components of a sorted in non-increasing order, see e.g. [35]) or even a Choquet integral with capacity ω (see e.g. [11, 16]). In this context, solution $x \in \mathcal{X}$ is preferred to solution $x' \in \mathcal{X}$ by the DM if and only if $f_\omega(y(x)) \leq f_\omega(y(x'))$. Thus any solution $x \in \mathcal{X}$ that minimizes function f_ω is optimal according to the DM's preferences.

2.2 Regret-based incremental elicitation

For the purpose of elicitation, we assume that preference parameters ω are not known initially. More precisely, we are given a (possibly empty) set Θ of preference statements of type $(a, b) \in \mathbb{R}^n \times \mathbb{R}^n$, meaning that the DM prefers point a to point b, and we consider the set Ω_Θ of all parameters ω that are compatible with Θ the available preference information. Formally, set Ω_Θ is defined by $\Omega_\Theta = \{\omega : \forall (a, b) \in \Theta, f_\omega(a) \leq f_\omega(b)\}$. Note that Ω_Θ is a convex polyhedron since function f_ω is assumed to be linear in its parameters ω.

Our goal now is to determine the most promising solution under the parameter imprecision characterized by Ω_Θ. To this aim, we consider the minimax regret approach (e.g., [7]) which is based on the following definitions:

Definition 1 (Pairwise Max Regret) *The Pairwise Max Regret (PMR) of solution $x \in \mathcal{X}$ with respect to solution $x' \in \mathcal{X}$ is:*

$$PMR(x, x', \Omega_\Theta) = \max_{\omega \in \Omega_\Theta} \Big\{ f_\omega(y(x)) - f_\omega(y(x')) \Big\}$$

By definition, $PMR(x, x', \Omega_\Theta)$ is the worst-case loss when recommending solution x instead of solution x' to the DM[1].

Definition 2 (Max Regret) *The Max Regret (MR) of solution $x \in \mathcal{X}$ is:*

$$MR(x, \mathcal{X}, \Omega_\Theta) = \max_{x' \in \mathcal{X}} PMR(x, x', \Omega_\Theta)$$

In other words, $MR(x, \mathcal{X}, \Omega_\Theta)$ is the worst-case loss when choosing x instead of any other solution $x' \in \mathcal{X}$. Finally, the minimax reget is defined as follows:

Definition 3 (Minimax Regret) *The MiniMax Regret (MMR) is:*

$$MMR(\mathcal{X}, \Omega_\Theta) = \min_{x \in \mathcal{X}} MR(x, \mathcal{X}, \Omega_\Theta)$$

[1] Note that $PMR(x, x', \Omega_\Theta)$ values can be computed using a LP solver since Ω_Θ is described by linear constraints and f_ω is linear in its parameters ω.

A solution $x^* \in \mathcal{X}$ is optimal according to the minimax regret decision criterion if x^* achieves the minimax regret, i.e. if $x^* \in \arg\min_{x \in \mathcal{X}} MR(x, \mathcal{X}, \Omega_\Theta)$. Recommending such a solution guarantees that the worst-case loss is minimized (given the imprecision surrounding the DM's preferences). Moreover, if $MMR(\mathcal{X}, \Omega_\Theta) = 0$, then we know that any optimal solution for the minimax regret criterion is necessarily optimal according to the DM's preferences.

Note that we have $MMR(\mathcal{X}, \Omega_{\Theta'}) \leq MMR(\mathcal{X}, \Omega_\Theta)$ for any set $\Theta' \supseteq \Theta$, as already observed in previous works (see e.g., [5]). Thus the general principle of regret-based incremental elicitation is to iteratively collect preference information by asking preference queries to the DM until $MMR(\mathcal{X}, \Omega_\Theta)$ drops below a given tolerance threshold $\delta \geq 0$ (representing the maximum allowable gap to optimality); if we set $\delta = 0$, then we obtain the (optimal) preferred solution at the end of the execution.

3 An Interactive Local Search Algorithm

For MOCO problems, regret-based incremental elicitation may induce prohibitive computation times since it may require to compute the pairwise max regrets for all pairs of distinct solutions in $\mathcal{X}$ (see Definitions 2 and 3). This observation has led a group of researchers to propose a new approach consisting in combining regret-based incremental elicitation and search by asking preference queries during the construction of the (near-)optimal solution (e.g. [2]). In this paper, we combine incremental elicitation and search in a different way. More precisely, we propose an interactive local search procedure that generates preference queries 1) before the local search to determine a promising starting point and 2) during the local search to help identifying the best solution within a neighborhood.

Our interactive algorithm takes as input a MOCO problem P, two thresholds $\delta = (\delta_1, \delta_2), (\delta_1, \delta_2 \geq 0)$, a scalarizing function f_ω with unknown parameters ω, an initial set of preference statements Θ (possibly empty), and m the number of possible starting solutions (generated at the beginning of the procedure). First, our algorithm identifies a promising starting solution as follows:

1. A set of m admissible preference parameters ω^k, $k \in \{1, \ldots, m\}$, are randomly generated within Ω_Θ.
2. Then, for every $k \in \{1, \ldots, m\}$, a (near-)optimal solution is determined for the *precise* scalarizing function f_{ω^k} using an existing efficient algorithm. Let X_0 be the set of generated solutions.
3. Finally, preference queries are generated in order to discriminate between the solutions in X_0. More precisely, while $MMR(X_0, \Omega_\Theta) > \delta_1$, the DM is asked to compare two solutions $x, x' \in X_0$ and the set of admissible parameters is updated by inserting the constraint $f_\omega(x) \leq f_\omega(x')$ (or $f_\omega(x) \geq f_\omega(x')$ depending on her answer); once $MMR(X_0, \Omega_\Theta)$ drops below δ_1, the starting solution x^* is arbitrarily chosen in $\arg\min_{x \in X_0} MR(x, X_0, \Omega_\Theta)$.

Then, our algorithm moves from solution to solution by considering local improvements. More precisely, it iterates as follows:

1. Firstly, a set X^* of solutions is generated from x^* using a neighborhood function defined on the search space; we add x^* to X^* and remove from X^* any solution that is Pareto-dominated by another solution in this set.
2. Secondly, while $MMR(X^*, \Omega_\Theta) > \delta_2$, the DM is asked to compare two solutions $x, x' \in X^*$ and Ω_Θ is restricted by inserting the constraint $f_\omega(x) \leq f_\omega(x')$ (or $f_\omega(x) \geq f_\omega(x')$).
3. Finally, if $MR(x^*, X^*, \Omega_\Theta) > \delta_2$ holds, solution x^* is then replaced by a neighbor solution minimizing the max regret in X^*; otherwise, the algorithm stops by returning solution x^*.

Our algorithm is called ILS for Interactive Local Search and is summarized in Algorithm 1.

Algorithm 1: `ILS`

IN $\downarrow P$: a MOCO problem; δ_1, δ_2: thresholds; f_ω: an aggregator with unknown parameters; Θ: a set of preference statements; m: number of initial solutions.

--| Initialization of the admissible parameters:
$\Omega_\Theta \leftarrow \{\omega : \forall (a,b) \in \Theta, f_\omega(a) \leq f_\omega(b)\}$
--| Generation of m initial solutions:
$X_0 \leftarrow$ `Select&Optimize`(P, Ω_Θ, m)
--| Determination of the starting solution:
while $MMR(X_0, \Omega_\Theta) > \delta_1$ **do**
  --| Ask the DM to compare two solutions in X_0:
  $(x, x') \leftarrow$ `Query`(X_0)
  --| Update preference information:
  $\Theta \leftarrow \Theta \cup \{(y(x), y(x'))\}$
  $\Omega_\Theta \leftarrow \{\omega : \forall (a,b) \in \Theta, f_\omega(a) \leq f_\omega(b)\}$
end while
$x^* \leftarrow$ `Select`$(\arg\min_{x \in X_0} MR(x, X_0, \Omega_\Theta))$
--| Interactive Local Search:
improve $\leftarrow$ **true**
while improve **do**
  $X^* \leftarrow$ `Neighbors`$(P, x^*) \cup \{x^*\}$
  while $MMR(X^*, \Omega_\Theta) > \delta_2$ **do**
    --| Ask the DM to compare two solutions in X^*:
    $(x, x') \leftarrow$ `Query`(X^*)
    --| Update preference information:
    $\Theta \leftarrow \Theta \cup \{(y(x), y(x'))\}$
    $\Omega_\Theta \leftarrow \{\omega : \forall (a,b) \in \Theta, f_\omega(a) \leq f_\omega(b)\}$
  end while
  --| Move to another solution:
  if $MR(x^*, X^*, \Omega_\Theta) > \delta_2$ **then**
    $x^* \leftarrow$ `Select`$(\arg\min_{x \in X^*} MR(x, X^*, \Omega_\Theta))$
  else
    improve $\leftarrow$ **false**
  end if
end while
return x^*

Note that procedures `Select&Optimize` and `Neighbors` depend on the problem considered; for instance, for the multicriteria spanning tree problem with a weighted sum model, the optimization part can be performed using Prim algorithm [27] and the neighborhood function can be defined by edge swaps. Note also that procedure `Query`(X) can implement any query generation strategy that selects two solutions in X and asks the DM to compare them; in the numerical section, we propose and compare different query generation strategies.

It is well-known that local search is a heuristic search that may stuck at a locally optimal point that is not globally optimal; the problem obviously remains when using our interactive local search. However, it is worth noting that our algorithm with $\delta_2 = 0$ provides the same performance guarantees as the corresponding local search algorithm with precise preferences. To give an example, when using the 2-opt neighborhood function [12], our algorithm approximately solves the TSP within a differential-approximation ratio bounded above by $1/2$ (see [24] for further details); in the numerical section, we will see that the error is even lower in practice.

For illustration purposes, we now present an execution of our algorithm on a small instance of the multi-objective TSP:

Example 2 *Consider the 3-objective TSP with* 6 *vertices defined by Figure 2. In this problem, the set $\mathcal{X}$ of feasible solutions is the set of all Hamiltonian cycles, i.e. cycles that include every node exactly once. We now apply ILS algorithm with $\delta = (0, 0)$ on this instance considering the neighborhood function defined by 2-opt swaps [12]; in other words, the neighbors of cycles are all the cycles that can be obtained by deleting two edges and adding two other edges from the graph (see Figure 2 for an example).*

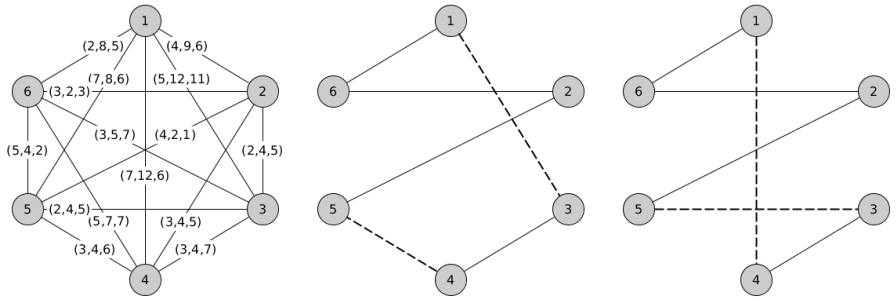

Fig. 2. The graph on the left side of the figure represents an instance of the 3-objective TSP with 6 vertices. The two others graphs give an example of a 2-opt movement: the dashed edges of the cycle in the middle are deleted and then replaced by the dashed edges in the right side of the figure.

We assume here that the DM's preferences can be represented by a weighted sum with the hidden weight $\omega^* = (0.2, 0.1, 0.7)$ and we start the execution with an empty set of preference statements (i.e. $\Theta = \emptyset$). Hence Ω_Θ is initially the set of all weighting vectors $\omega = (\omega_1, \omega_2, \omega_3) \in [0, 1]^3$ such that $\omega_1 + \omega_2 + \omega_3 = 1$. In

Figure 3, we represent Ω_Θ by the triangle ABC in the space (ω_1, ω_2), ω_3 being implicitly defined by $\omega_3 = 1 - \omega_1 - \omega_2$.

Identification of a promising starting solution: *First, we generate $m = 2$ weighting vectors ω^1 and ω^2 at random and we determine then the corresponding optimal solutions x^1 and x^2. If $\omega^1 = (0.6, 0.3, 0.1)$ and $\omega^2 = (0.3, 0.6, 0.1)$, we obtain the following evaluation: $y(x^1) = (19, 34, 30)$ and $y(x^2) = (21, 32, 27)$. Let $X_0 = \{x^1, x^2\}$. Since $MMR(X_0, \Omega_\Theta) = 2 > \delta_1$, we ask the DM to compare x^1 and x^2. Since $f_{\omega^*}(y(x^1)) = 28.2 > f_{\omega^*}(y(x^2)) = 26.3$, the DM prefers solution x^2 to x^1. Therefore we set $\Theta = \{((21, 32, 27), (19, 34, 30))\}$ and Ω_Θ is restricted by imposing the constraint $f_\omega(y(x^2)) \leq f_\omega(y(x^1))$, i.e. $\omega_2 \leq -5\omega_1 + 3$ (see Figure 4 where Ω_Θ is represented by ABDE). Now we have $MMR(X_0, \Omega_\Theta) = MR(x^2, X_0, \Omega_\Theta) = 0 \leq \delta_1$, and therefore x^2 is chosen to be the starting solution (i.e. $x^* = x^2$).*

Local Search: *At the first iteration step, three neighbors of x^* are Pareto non-dominated, and the set X^* contains four solutions, denoted by x^1, $x^2(= x^*)$, x^3 and x^4 evaluated as follows: $y(x^1) = (23, 34, 26)$, $y(x^2) = (21, 32, 27)$, $y(x^3) = (19, 34, 30)$ and $y(x^4) = (20, 31, 30)$. Since $MMR(X^*, \Omega_\Theta) = 1 > \delta_2$, we ask the DM to compare two solutions in X^*, say x^1 and x^*. As $f_{\omega^*}(y(x^1)) = 26.2 < f_{\omega^*}(y(x^*)) = 26.3$, the DM prefers x^1 to x^*. Therefore we obtain $\Theta = \{((21, 32, 27), (19, 34, 30)), ((23, 34, 26), (21, 32, 27))\}$ and Ω_Θ is restricted by the linear constraint $f_\omega(y(x^1)) \leq f_\omega(y(x^*))$, i.e. $\omega_2 \leq -\omega_1 + 1/3$ (see Figure 5 where Ω_Θ is represented by AGF). Then we stop asking queries at this step since we have $MMR(X^*, \Omega_\Theta) = MR(x^1, X^*, \Omega_\Theta) = 0 \leq \delta_2$. We move from $x^* = x^2$ to solution x^1 for the next step (i.e., we now set $x^* = x^1$).*

At the second iteration step, X^* only includes three solutions denoted by $x^1(= x^*)$, x^2 and x^3 with $y(x^1) = (23, 34, 26)$, $y(x^2) = (21, 32, 27)$ and $y(x^3) = (19, 33, 31)$. Since $MMR(X^*, \Omega_\Theta) = 0 \leq \delta_2$, no query is generated at this step. Moreover, $MR(x^*, X^*, \Omega_\Theta) = 0 \leq \delta_2$ (that is $x^* \in \arg\min_{x \in X^*} MR(x, X^*, \Omega_\Theta)$) and x^* is thus a local optimum (variable improve is set to **false** *and the while loop ends). Therefore, after two iteration steps, ILS algorithm stops by returning the solution $x^* = x^1$ which is the preferred solution in this problem. Note that only two preference queries were needed to discriminate between the 60 feasible solutions (among which 10 are Pareto-optimal).*

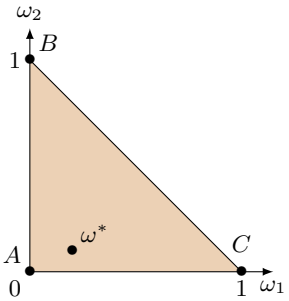

Fig. 3. Initial set Ω_Θ.

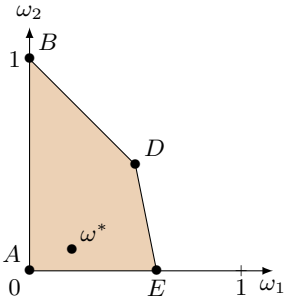

Fig. 4. Ω_Θ after 1 query.

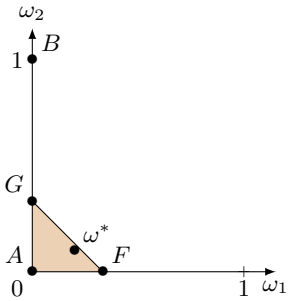

Fig. 5. Ω_Θ after 2 queries.

To illustrate ILS and the impact of m (the number of initial solutions), we show the evolution of the local search on a randomly generated instance of the bi-criteria TSP with 100 cities (see Figure 6). The left part of the figure ($m = 1$) shows that the neighborhood function enables to go straight to the optimal solution instead of following the Pareto front. However the number of iterations can still be very large when the starting solution is far from the optimal solution in the objective space. The right part of the figure ($m = 2$) shows that the number of iterations can be much reduced when increasing the number of possible starting solutions and selecting the most preferred one.

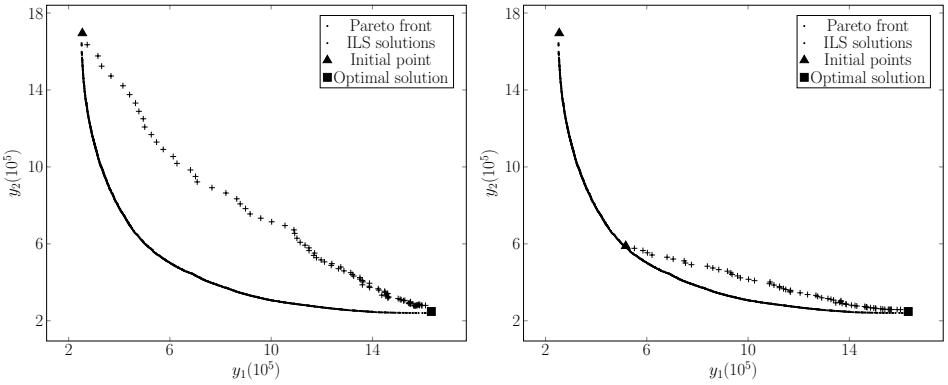

Fig. 6. Results obtained for an instance of the 2-criteria TSP.

4 Numerical Tests

In this section, we provide numerical results aiming to estimate the performance of our algorithm. In these experiments, we use existing Euclidean instances[2] of the multi-objective TSP with 25 to 100 cities, and $n = 3$ to 6 objectives. Numerical tests were performed on a Intel Core i7-8550U CPU with 16 GB of RAM, with a program written in C++[3].

4.1 Preferences represented by a weighted sum

We assume here that the preferences are represented by a weighted sum f_ω with imprecise weights, with an empty set of preference statements (i.e. $\Theta = \emptyset$). The answers to queries are simulated using a weighting vector ω randomly generated before running the algorithm, using the procedure presented in [29], to guarantee a uniform distribution of the weights.

[2] `https://eden.dei.uc.pt/~paquete/tsp/`

[3] PMRs values are computed using CPLEX Optimizer (`https://www.ibm.com/analytics/cplex-optimizer`) and the optimization part of `Select&Optimize` is performed by the exact TSP solver Concorde (`http://www.math.uwaterloo.ca/tsp/concorde`).

In ILS algorithm, procedure `Query`(X) selects two solutions from X and then asks the DM to compare them. We first want to estimate the impact of the query generation strategy on the performances of ILS algorithm. To do so, we consider the following two query generation strategies:

- **Random:** the set of possibly optimal solutions in X are computed and then two of them are selected at random at each iteration step; note that the set of possibly optimal solutions in X can be determined in polynomial time in the size of X (see e.g., [1]).
- **CSS:** The Current Solution Strategy (CSS) selects a solution $x^* \in X$ that minimizes the max regret $MR(x^*, X, \Omega_\Theta)$ and then the DM is asked to compare solution x^* with one of its adversary's choice (i.e. a solution in $\arg\max_{x \in X} PMR(x^*, x, \Omega_\Theta)$) [7].

These strategies are compared in terms of computation time (given in seconds), number of generated queries and the error (expressed in terms of percentage from the optimal solution). Results averaged over 100 runs are given in Table 1 for the instance with 50 cities. In this table, "/" means that the timeout is exceeded (the timeout is set to 900 seconds).

		CSS			Random		
n	m	time	queries	error	time	queries	error
4	10	28.52	35.90	1.96	15.33	77.54	1.90
4	50	8.65	27.31	1.11	317.11	380.18	0.92
4	100	6.71	24.34	0.69	894.93	375.83	0.70
6	10	537.07	85.67	2.06	67.91	168.71	2.31
6	50	133.08	68.45	1.64	899.39	338.98	1.49
6	100	43.06	54.40	1.35	> 900	/	/

Table 1. Comparison between CSS and Random strategies, 50 cities, $\delta = (0, 0)$.

First, we see that the query strategy has an important impact on the quality of the results: with the random strategy the number of queries and computation time are much higher than with the CSS strategy, showing that preference queries must be carefully chosen when designing incremental elicitation methods. Then we see that ILS with CSS achieves better results when increasing the number of possible initial solutions, in terms of computation times, queries and error (as the selected starting solution is becoming closer to the best solution). Although the error is very low (about less than 2%) for both strategies, the number of queries is quite high for instances with 6 criteria (at least 54 queries). This is due to the fact that we use the tolerance thresholds $\delta = (0, 0)$.

We now compare the results obtained with $\delta = (0, 0)$ and $\delta = (0.1, 0.4)$ (see the left part of Table 2); we set $\delta_1 < \delta_2$ since the starting point selection has a significant impact on local search performances. We also give the results obtained by ILS when a ranking of the objectives can be provided by the DM prior to the search (see the right part of Table 2). We vary n the number of criteria between 3 and 6, and we set m the number of initial solutions to 100.

In Table 2, we see that using strictly positive thresholds enables to reduce both the number of queries and the computation time without having much impact on the error. For instance, for $n = 6$, the error is equal to 1.35% with $\delta = (0, 0)$ whereas it is equal to 1.77% with $\delta = (0.1, 0.4)$, while the number of queries is reduced from 54.40 to 32.24. We also remark that knowing the ranking of the objectives allows to further reduce the number of queries (only 23.75 queries for $n = 6$, with an error of 1.51%).

	$\delta = (0,0)$			$\delta = (0.1, 0.4)$			$\delta = (0,0)$			$\delta = (0.1, 0.4)$		
n	time	queries	error	time	queries	error	time	queries	error	time	queries	error
3	2.67	13.50	0.20	3.81	13.65	2.01	2.10	10.53	0.20	2.30	9.66	0.80
4	6.71	24.34	0.69	8.32	19.68	1.84	5.01	17.01	0.66	4.89	13.39	1.13
5	19.96	38.38	0.96	19.30	26.21	2.08	14.81	25.08	0.95	10.97	19.24	1.38
6	43.06	54.40	1.35	36.25	32.24	1.77	31.19	34.52	1.36	29.68	23.75	1.51

Table 2. Performances of ILS combined with CSS, with (left) and without the ranking of the objectives (right), 50 cities, $m = 100$, 100 runs.

In Figure 7, we show the evolution of the number of queries according to the number of criteria (left part, 50 cities) and number of cities (right), with $\delta = (0.1, 0.4)$ and $m = 100$. We note that the number of queries evolves more or less linearly according to the number of criteria/cities.

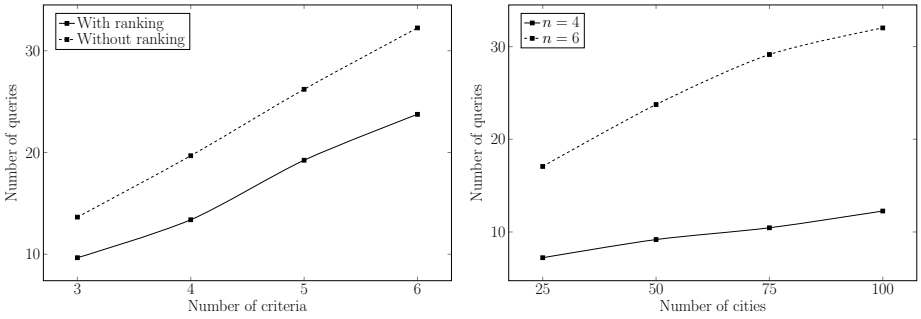

Fig. 7. Evolution of the queries according to number of criteria and number of cities.

4.2 Preferences represented by an OWA aggregator

Now we assume that the DM's preferences can be represented by an OWA aggregator, with decreasing weights, favoring well-balanced solutions [35] (as the larger weights are associated with the worst values). Contrary to the weighted sum, when the weights ω are known, we cannot reduce the multi-objective TSP to a single-objective TSP as the OWA aggregator is not a linear operator. Therefore, to obtain the optimal solution with known weights (to be able to compute the error), we have used a well-known linearization of the OWA operator with decreasing weights (see [25]); for information purposes, we also provide the computation time needed when solving the corresponding linear program with the LP-solver (see "LP time").

In Table 3, we give the results obtained by ILS combined with the CSS for the instances with 50 cities, 3 to 6 criteria and $m = 10$ starting solutions (obtained

by optimizing 10 randomly generated weighted sum). The results show that the number of queries is much reduced compared to the weighted sum, with an error also around 2%. Moreover, we observe that using our algorithm to solve the problem with unknown weights is much faster than using the LP-solver to obtain the optimal solution when the weights are known. This shows that our algorithm can be used to efficiently solve multi-objective optimization problems with complex decision models, even for problems such that there is no efficient algorithm for the determination of the optimal solution with known preference parameters.

		$\delta = (0, 0)$			$\delta = (0.1, 0.4)$		
n	LP time	time	queries	error	time	queries	error
3	164.80	1.14	9.96	2.31	1.19	6.16	2.41
4	330.17	1.28	10.57	1.98	1.06	6.13	2.00
5	730.19	0.98	6.40	1.17	0.83	4.09	1.42
6	7870.03	1.12	16.00	1.41	0.89	9.13	1.44

Table 3. ILS combined with CSS with OWA, 50 cities, $m = 10$.

Finally, in Figure 8, we illustrate the iterations of ILS when preferences are represented by an OWA operator with decreasing weights (which favors well-balanced solutions).

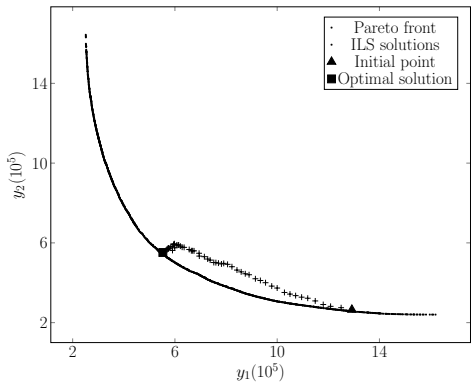

Fig. 8. Results obtained for an instance of the 2-criteria TSP with OWA operator.

5 Conclusion

In this paper, we have proposed a general approach based on local search and incremental elicitation for solving multi-objective combinatorial optimization problems with unkown preference parameters. We have applied the method to a NP-hard combinatorial optimization problem and we have shown that, by combining the generation of promising starting solutions with an adaptive preference-based local search, we are able to rapidly obtain high quality solutions, even with a non-linear aggregation function like OWA. The approach can

be applied to any multi-objective combinatorial optimization problem provided that the scalarizing function used to compare solutions is linear in its parameters.

References

1. N. Benabbou and P. Perny. Incremental weight elicitation for multiobjective state space search. In *Proceedings of AAAI'15*, pages 1093–1098, 2015.
2. N. Benabbou and P. Perny. On possibly optimal tradeoffs in multicriteria spanning tree problems. In *Proceedings of ADT'15*, pages 322–337, 2015.
3. N. Benabbou and P. Perny. Solving multi-agent knapsack problems using incremental approval voting. In *Proceedings of ECAI'16*, pages 1318–1326, 2016.
4. N. Benabbou and P. Perny. Interactive resolution of multiobjective combinatorial optimization problems by incremental elicitation of criteria weights. *EURO Journal on Decision Processes*, 6(3):283–319, Nov 2018.
5. N. Benabbou, P. Perny, and P. Viappiani. Incremental elicitation of Choquet capacities for multicriteria choice, ranking and sorting problems. *Artificial Intelligence*, 246:152180, 2017.
6. N. Bourdache and P. Perny. Active preference elicitation based on generalized Gini functions: Application to the multiagent knapsack problem. In *Proceedings of AAAI'19*, 2019.
7. C. Boutilier, R. Patrascu, P. Poupart, and D. Schuurmans. Constraint-based Optimization and Utility Elicitation using the Minimax Decision Criterion. *Artifical Intelligence*, 170(8–9):686–713, 2006.
8. D. Braziunas and C. Boutilier. Minimax regret based elicitation of generalized additive utilities. In *Proc. of UAI-07*, pages 25–32, 2007.
9. E. Brochu, F. Nando D, and G. Abhijeet. Active preference learning with discrete choice data. In *Advances in neural information processing systems*, pages 409–416, 2008.
10. U. Chajewska, D. Koller, and R. Parr. Making rational decisions using adaptive utility elicitation. In *Proceedings of AAAI-00*, pages 363–369, 2000.
11. G. Choquet. Theory of capacities. *Annales de lInstitut Fourier*, 5:31–295, 1953.
12. G.A. Croes. A method for solving traveling-salesman problems. *Operations research*, 6(6):791–812, 1958.
13. J. Drummond and C. Boutilier. Preference elicitation and interview minimization in stable matchings. In *Proceedings of AAAI'14*, pages 645–653, 2014.
14. J. Fürnkranz and E. Hüllermeier. *Preference learning*. Springer, 2010.
15. M. Gelain, M.S. Pini, F. Rossi, K. B. Venable, and T. Walsh. Elicitation strategies for soft constraint problems with missing preferences: Properties, algorithms and experimental studies. *Artificial Intelligence*, 174(3):270–294, 2010.
16. M. Grabisch and C. Labreuche. A decade of application of the Choquet and Sugeno integrals in multi-criteria decision aid. *Annals of Operations Research*, 175(1):247–286, 2010.
17. V. Ha and P. Haddawy. Problem-focused incremental elicitation of multi-attribute tility models. In *Proceedings of UAI-97*, pages 215–222. Morgan Kaufmann Publishers Inc., 1997.
18. H.W. Hamacher and G. Ruhe. On spanning tree problems with multiple objectives. *Annals of Operations Research*, 52:209–230, 1994.
19. S. Kaddani, D. Vanderpooten, J-M. Vanpeperstraete, and H. Aissi. Weighted sum model with partial preference information: application to multi-objective optimization. *European Journal of Operational Research*, 260:665–679, 2017.

20. E. Karasakal and M. Köksalan. Generating a representative subset of the nondominated frontier in multiple criteria decision making. *Operations Research*, 57(1):187–199, 2009.
21. M. Laumanns, L. Thiele, K. Deb, and E. Zitzler. Combining convergence and diversity in evolutionary multiobjective optimization. *Evol. Comput.*, 10(3):263–282, 2002.
22. J. Lesca and P. Perny. LP solvable models for multiagent fair allocation problems. In *ECAI 2010*, pages 393–398, 2010.
23. T. Lu and C. Boutilier. Robust approximation and incremental elicitation in voting protocols. In *IJCAI 2011*, pages 287–293, 2011.
24. J. Monnot, V.T. Paschos, and . Toulouse. Approximation algorithms for the traveling salesman problem. *Mathematical methods of operations research*, 56(3):387–405, 2003.
25. W. Ogryczak and T. liwiski. On solving linear programs with the ordered weighted averaging objective. *European Journal of Operational Research*, 148(1):80 – 91, 2003.
26. P. Perny, P. Viappiani, and A. Boukhatem. Incremental preference elicitation for decision making under risk with the rank-dependent utility model. In *Proceedings of UAI*, pages 597–606, 2016.
27. R. C. Prim. Shortest connection networks and some generalizations. *Bell System Technical Journal*, 36:1389–1401, 1957.
28. K. Regan and C. Boutilier. Eliciting additive reward functions for markov decision processes. In *Proceedings of IJCAI'11*, pages 2159–2164, 2011.
29. R.Y. Rubinstein. Generating random vectors uniformly distributed inside and on the surface of different regions. *European Journal of Operational Research*, 10(2):205 – 209, 1982.
30. A. Salo and R.P. Hämäläinen. Preference ratios in multiattribute evaluation (prime)-elicitation and decision procedures under incomplete information. *IEEE Trans. Systems, Man, and Cybernetics, Part A*, 31(6):533–545, 2001.
31. A.F. Tehrani, W. Cheng, K. Dembczyński, and E. Hüllermeier. Learning monotone nonlinear models using the Choquet integral. *Machine Learning*, 89(1-2):183–211, 2012.
32. T. Wang and C. Boutilier. Incremental Utility Elicitation with the Minimax Regret Decision Criterion. pages 309–316, 2003.
33. P. Weng and B. Zanuttini. Interactive Value Iteration for Markov Decision Processes with Unknown Rewards. In *Proceedings of IJCAI'13*, pages 2415–2421, 2013.
34. C. C. White, A. P. Sage, and S. Dozono. A model of multiattribute decisionmaking and trade-off weight determination under uncertainty. *IEEE Transactions on Systems, Man, and Cybernetics*, 14(2):223–229, 1984.
35. R.R. Yager. On Ordered Weighted Averaging aggregation operators in multicriteria decision making. *IEEE Transactions on systems, Man, and Cybernetics*, 18(1):183–190, 1988.