A Genetic Algorithm Approach for an Equitable Treatment of Objective Functions in Multi-objective Optimization Problems

Mohammed Gabli, El Miloud Jaara, and El Bekkaye Mermri

***Abstract*—A reasonable solution to a multi-objective problem is to determine an entire Pareto optimal solution set. Another general approach is to transform a multi-objective optimization problem into a mono-objective one. Determination of a single objective is possible with methods such as weighted sum method, but the problem lies in the right selection of the weights to characterize the decision makers preferences. In this paper, we study the problem where the decision maker tries to balance the objective function weights. This task is not easy for both decision maker and system analyser. To remedy this problem we introduce a solution method based on a genetic algorithm which automates the choice of the weights by varying them at each iteration of the algorithm. Our algorithm is tested on five academic problems and is applied to a UMTS base station location planning problem. The obtained results show that the proposed approach ensures an equitable treatment of each objective function.**

***Index Terms*—genetic algorithm, multi-objective optimization, weighted sum method, UMTS problem.**

I. Introduction

MOST of the real-life decision-making problems have more than one conflicting objective function. Researchers studied multi-objective optimization problems from different viewpoints and, thus, there exist different solution methods and goals for solving them. The goal may be finding a representative set of Pareto optimal solutions, or quantifying the trade-offs in satisfying the different objectives, or finding a single solution that satisfies the preferences of a decision maker. The approaches used for solving multi-objective problems (MOPs) can be classified into three categories: approaches based on the transformation of the problem into a mono-objective problem, non-Pareto approaches and Pareto approaches (see [12], [20], [24], for instance).

In the literature, a special and particular attention is given to multi-objective problems using exact and approximate algorithms. Exact methods such as Branch and Bound, the A* algorithm and Dynamic Programming are effective for problems of small sizes. When problems become harder, usually because of their NP-hard complexity, approximate algorithms are mandatory. Several adaptations of metaheuristics have been proposed in the literature for solving multi-objective problems [21], [31]. In [5] authors described a simulated annealing based multi-objective optimization algorithm (AMOSA) that incorporates the concept of archive in order to provide a set of trade-off solutions for the problem under consideration. In [19], an improved multi-objective genetic algorithm (MOGA), named SMGA, was proposed for solving multi-objective optimization problems. In [27] author presented genetic algorithms for solving fuzzy multi-objective optimization. In [1] authors proposed three multi-objective Artificial Bee Colony (ABC) algorithms based on synchronous and asynchronous models using Pareto-dominance and non-dominated sorting. Some proposal for extending particle swarm optimization algorithms to treat MOPs, have been published in [6] and [36]. A new multi-objective algorithm based on Monte Carlo tree search is proposed in [26]. In [4] authors presented a new method of solving non-linear multi-objective optimization problems by adding a control function that guides the optimization process over the Pareto set which does not need to be found explicitly. In [37] authors proposed a method, called MOEA/D-EGO, to deal with expensive multi-objective optimization. Recently, new metaheuristic search algorithms have been developed. In [33] and [32] authors extended, respectively, the flower algorithm and the firefly algorithm to solve multi-objective optimization problems.

Several methods and approaches using genetic algorithms (GAs) have been developed for solving MOP, for example, Vector Evaluated Genetic Algorithm (VEGA) [29], Multi-Objective Genetic Algorithm (MOGA) [13], Niched Pareto Genetic Algorithm (NPGA) [18], Strength Pareto Evolutionary Algorithm (SPEA) [38], Fast Non-dominated Sorting Genetic Algorithm (NSGA-II) [8], Multi-objective Evolutionary Algorithm (MEA) [28], Dynamic Multi-objective Evolutionary Algorithm (DMOEA) [35], Pareto Fitness Genetic Algorithm (PFGA) [11], Sharing Mutation Genetic Algorithm(SMGA) [19] and others. Coello [7] maintains an updated list with more than 5000 titles of publications involving different genetic algorithms.

The weighted sum method is the simplest approach and probably the most widely used classical method [25]. If multiple solutions are desired, the problem should be solved multiple times with different weight combinations. The main difficulty with this approach is the selection of a weight vector for each run of the program. In [23] researchers have proposed a multi-objective genetic algorithm based on a weighted sum of multiple objective functions where a normalized weight vector is randomly generated for each solution during the selection phase at each generation. In [34] the results show that weighted sum method combined with genetic algorithm can quickly search the solution of a high-speed gasoline engine problem.

In this paper we are interested in the weighted sum method which transforms the multi-objective optimization problem

Manuscript received September 10, 2013; revised February 11, 2014.
M. Gabli, E. M. Jaara and E. B. Mermri are with Department of Mathematics and Computer Science, Faculty of Science, University Mohammed Premier, BV Mohammed 6, Oujda, Morocco. e-mails: medgabli@yahoo.fr; emjaara@yahoo.fr; b.mermri@fso.ump.ma.

into a mono-objective one. More precisely we are interested in the interactive method. The problem lies in the right selection of the weights to characterize the decision makers preferences. Here, we concentrate on the case where the decision maker tries to ensure an equitable treatment of each objective function. For this we introduce a solution method based on a genetic algorithm which automates the choice of the weights by varying them at each iteration of the algorithm. As an application of our solution method we consider a telecommunication base station location problem.

The remaining of the paper is outlined as follows. We describe the problem in Section II. In Section III, we introduce a solution method based on a genetic algorithm (GA) which automates the choice of the weights at each iteration of the algorithm. Section IV provides the obtained experimental results corresponding to five academic problems. As application, we study a UMTS base station location planning problem. The results of this problem are presented in section V. Finally, in Section VI we give some concluding remarks.

II. Problem Presentation

A multi-objective optimization problem is a problem that involves more than one objective function to be optimized simultaneously. In mathematical terms, a multi-objective optimization problem can be formulated, in the context of maximization, as (see [36], for instance):

$$\begin{aligned} &\underset{x}{\text{Maximize}}\ (f_1(x), f_2(x), \cdots, f_I(x))^T \\ &\text{S.t.}\quad g_j(x) \geq 0,\ \ j = 1, 2, \cdots, J \\ &\qquad h_m(x) = 0,\ \ m = 1, 2, \cdots, M \\ &\qquad x_n \in [x_{nl}, x_{nu}],\ \ n = 1, 2, \cdots, N \end{aligned} \tag{1}$$

where the integer $I \geq 2$ is the number of objective functions, $f_i(x)$ is the i-th objective or criterion, $g_j(x)$ is the j-th inequality constraint, $h_m(x)$ is the m-th equality constraint, $x = (x_1, x_2, \cdots, x_N)^T \in \Re^N$ (search space) is the vector (solution) of decision variable, x_{nl} and x_{nu} are the superior boundary value and the inferior boundary value of each component x_n of the vector x, respectively. In the following, we replace all these constraints by $x \in C$, C is the feasible set of decision variable, called also constrained set. The maximization in (1) is understood compenentwise.

In this paper we are interested in the weighted sum method which transforms the multi-objective optimization problem (1) into a mono-objective one as follows.

$$\begin{aligned} &\underset{x}{\text{Maximize}}\ h(x) = \sum_{i=1}^{I} w_i f_i(x) \\ &\text{S.t.}\ x \in C, \end{aligned} \tag{2}$$

where the weights w_i, $1 \leq i \leq I$, are positive values satisfying

$$0 \leq w_i \leq 1 \ \text{ and } \ \sum_{i=1}^{I} w_i = 1. \tag{3}$$

When we transform problem (1) into problem (2), the choice of the weights w_i is not an easy task for both the decision maker and the system analyser. Assume that the decision maker tries to balance the objective function weights, then problem (2) becomes

$$\begin{aligned} &\underset{x}{\text{Maximize}}\ h(x) = \sum_{i=1}^{I} w_i f_i(x) \\ &|w_i f_i(x) - w_j f_j(x)| \prec \epsilon,\ \ i, j = 1, 2, \cdots, I \\ &\text{S.t}\ x \in C, \end{aligned} \tag{4}$$

where ϵ is a positif number in the vicinity of 0.
In the literature, the weights are usually taken constants. In Section III-A, we show that, when using genetic algorithm (GA) to solve problem (4), it is not always appropriate to take w_i as constants. Instead, we introduce dynamic weights. Then, the problem (4) becomes:

$$\begin{aligned} &\underset{x}{\text{Maximize}}\ h(x,t) = \sum_{i=1}^{I} w_i(t) f_i(x) \\ &|w_i(t) f_i(x) - w_j(t) f_j(x)| \prec \epsilon,\ \ i, j = 1, 2, \cdots, I \\ &\text{S.t}\ x \in C, \end{aligned} \tag{5}$$

where t is a time-step, and $w_i(t)$ is the dynamic weight satisfying

$$0 \leq w_i(t) \leq 1 \ \text{ and } \ \sum_{i=1}^{I} w_i(t) = 1. \tag{6}$$

In this paper, t represents an iteration step of the GA.

III. Genetic Algorithm and Dynamic weighted Method

Genetic algorithm (GA) is a search and optimization technique that mimics natural evolution. GA has already a relatively old history since the first work of John Holland on the adaptive systems goes back to 1962 [17]. The work of David Goldberg [15] largely contributed to popularize the GA. GA is inspired by the evolutionist theory explaining the origin of species. The main components of a GA are: selection, crossover and mutation.

A. Limits of Choosing the Weights as Constants

In this section we will show the drawback of taking the weights w_i, in problem (4), as constants when solving this problem by using GA. Without loss of generality, we consider a multi-objective maximization problem with two objective functions ($I = 2$).

$$\begin{aligned} &\underset{x}{\text{Maximize}}\ h(x) = w_1 f_1(x) + w_2 f_2(x) \\ &|w_1 f_1(x) - w_2 f_2(x)| \prec \epsilon, \\ &\text{S.t.}\ x \in C. \end{aligned} \tag{7}$$

Assume that f_1 is much greater than f_2. When applying GA to maximize the objective function h, if we take the weights w_1 and w_2 as constants, then there is a great risk that the GA selection procedure chooses only solutions which improve f_1 by neglecting f_2, since the function f_1 dominates f_2. We illustrate the drawback through the following example.
Let $[a, b]$ be a real interval, and f_1 and f_2 be two real functions satisfying:

$$10^3 \leq f_1(x) \leq 5 \times 10^4 \ \text{ and } \ 0 \leq f_2(x) \leq 1, \quad x \in [a, b].$$

The problem is to maximize:

$$h(x) = w_1 f_1(x) + w_2 f_2(x), \quad x \in [a, b].$$

For example, let $w_1 = 0.0001$ and $w_2 = 0.9999$. Suppose that in the iteration k of the GA, the solution x_k satisfying: $f_1(x_k) = 9 \times 10^3$ and $f_2(x_k) = 0.1$ is retained as a solution for h, i.e. the best among solutions of the current population. Then we have:

$$h(x_k) = w_1 f_1(x_k) + w_2 f_2(x_k) = 0.9 + 0.09999 \approx 1.$$

Assume that in the next iteration of the GA, we find two solutions $x_{k+1}^{(1)}$ and $x_{k+1}^{(2)}$ satisfying:

$$f(x_{k+1}^{(1)}) = 5 \times f(x_k) \quad \text{and} \quad g(x_{k+1}^{(1)}) \approx g(x_k)$$

$$f(x_{k+1}^{(2)}) \approx f(x_k) \quad \text{and} \quad g(x_{k+1}^{(2)}) = 5 \times g(x_k)$$

Then we have:

$$h(x_{k+1}^{(1)}) = w_1 f_1(x_{k+1}^{(1)}) + w_2 f_2(x_{k+1}^{(1)}) \approx 4.6$$

and

$$h(x_{k+1}^{(2)}) = w_1 f_1(x_{k+1}^{(2)}) + w_2 f_2(x_{k+1}^{(2)}) \approx 1.4.$$

It is clear that although $x_{k+1}^{(1)}$ and $x_{k+1}^{(2)}$ improve, respectively, the objective functions f_1 and f_2 in the same way, the probability that GA selects $x_{k+1}^{(2)}$ is weak compared to the probability of selecting $x_{k+1}^{(1)}$. When one takes w_1 and w_2 fixed, the fact that an objective function dominates another is extremely probable. In this example, a solution which improves the value of f_2 does not have the same influence on maximizing h as the one which improves the value of f_1. To remedy this problem we should not take the value of weights as constants, but rather this value must be dynamic and it changes in each iteration of the GA.

B. A GA Method with Dynamic Weights

Now, we present a GA approach applied to the optimization model (5), using dynamic weights.

Let $h = \sum_{i=1}^{I} w_i(t) f_i$, $I \geq 2$, be the fitness function of the GA. In each iteration t of the GA we take:

$$w_i(t) := \frac{\sum_{\substack{j=1 \\ j \neq i}}^{I} |f_j(x_{t-1})|}{(I-1) \times \sum_{j=1}^{I} |f_j(x_{t-1})|}, \quad i = 1 \ldots I,$$

where x_{t-1} is the best individual among solutions of the population $P(t-1)$, with respect to the fitness function h; if $f_j(x_{t-1}) = 0$ for any $1 \leq j \leq I$, we take $w_i(t) := w_i(t-1)$. It is easy to see that $0 \leq w_i(t) < 1$, $1 \leq i \leq I$, and $\sum_{i=1}^{I} w_i(t) = 1$. Then the algorithm is outlined as follows:

Step 0. At the initialization step of the GA, we assign arbitrary positive real numbers to $w_i(0)$, $i = 1, \ldots, I$, satisfying the condition (6);

Step 1. Run an iteration t of the GA, with the fitness function $h = \sum_{i=1}^{I} w_i(t) f_i$;

Step 2. Let x_t be the best solution among solutions of the current population;

Step 3. Calculate $f_i(x_t)$, $i = 1, \cdots, I$;

Step 4. If $\sum_{i=1}^{I} |f_i(x_t)| \neq 0$ then take

$$w_i(t+1) = \frac{\sum_{\substack{j=1 \\ j \neq i}}^{I} |f_j(x_t)|}{(I-1) \times \sum_{j=1}^{I} |f_j(x_t)|}, \; i = 1, \ldots, I;$$

Step 5. $t := t + 1$;

Step 6. Repeat steps 1 through 5 until a stopping criterion is satisfied.

Our algorithm has two immediate advantages:

- It automates the choice of the weights. The utility resides, therefore in the fact that we do not need to define this factor in advance. This task turns out to be a very delicate question.
- It ensures an equitable treatment of each objective function, so we have an equitable chance to maximize the functions f_i, $i = 1, \cdots, I$, simultaneously. For example, for $I = 2$, in the iteration t, the function $h(x,t) = w_1(t) f_1(x) + w_2(t) f_2(x)$ becomes:

$$h(x,t) = \frac{|f_2(x_{t-1})|}{|f_1(x_{t-1})| + |f_2(x_{t-1})|} f_1(x) + \frac{|f_1(x_{t-1})|}{|f_1(x_{t-1})| + |f_2(x_{t-1})|} f_2(x).$$

  Where x_{t-1} is the best solution of the iteration $(t-1)$ of the GA.

C. Explanation

In this section we will explain our algorithm for only two objective functions. The generalization is trivial. For this, we are interested in linear scaling method (see [15], [22], [27], for instance), where the fitness h is transformed into h_{sc} according to

$$h_{sc} = a \times h + b,$$

where the coefficients a and b are determined so that the mean fitness h_{mean} of the population should be a fixed point and the maximal fitness h_{max} of the population should be equal to $c \times h_{mean}$. The constant c, usually set as $1.2 \leq c \leq 2$, means the expected value of the number of the best individual in the current generation surviving in the next generation (see for instance [27]).

We use linear scaling method and we take the coefficient a very smaller than 1. For $I = 2$, we have

$$h(x,t) = w_1(t) f_1(x) + w_2(t) f_2(x).$$

Therefore,

$$h_{sc}(x,t) = w_1(t) \times a \times f_1(x) + w_2(t) \times a \times f_2(x) + b.$$

Then, the application of scaling at h induces an application of scaling at f_1 and f_2. Thus, the optimization model (5) becomes:

$$h_{sc}(x,t) = \frac{a \times |f_2(x_{t-1})|}{|f_1(x_{t-1})| + |f_2(x_{t-1})|} f_1(x) + \frac{a \times |f_1(x_{t-1})|}{|f_1(x_{t-1})| + |f_2(x_{t-1})|} f_2(x) + b.$$

Where x_{t-1} is the best solution of h_{sc} selected for the previous iteration $(i-1)$. Therefore, x_{t-1} survives in the current population. Then, the difference between $f_1(x)$ and

$f_1(x_{t-1})$, on the one hand, and the difference between $f_2(x)$ and $f_2(x_{t-1})$, on the other hand, are reduced. Therefore, the difference between the two terms of the objective function h_{sc} is reduced.

D. Example

Consider the example of the objective functions f_1 and f_2 presented in Section III-A. We apply the proposed algorithm, given in Section III-B, to maximize $h = w_1 f_1 + w_2 f_2$.

- Let x_{t-1} be the best solution among solutions of the population in the iteration $(t-1)$ of the GA. Assume that the solution x_{t-1} satisfies $f_1(x_{t-1}) = 9 \times 10^3$ and $f_2(x_{t-1}) = 0.1$. Then the weights, in the iteration t, are defined as:

$$w_1(t) = \frac{|f_2(x_{t-1})|}{|f_1(x_{t-1})| + |f_2(x_{t-1})|} \approx 1.11109 \times 10^{-5}$$

and

$$w_2(t) = \frac{|f_1(x_{t-1})|}{|f_1(x_{t-1})| + |f_2(x_{t-1})|} \approx 0.99998.$$

Therefore $h(x_{t-1}, t) = w_1(t) f_1(x_{t-1}) + w_2(t) f_2(x_{t-1})$

$$\approx 1.999961 \times 10^{-1}.$$

- Then, in the next iteration of the GA we have to maximize the function

$$h(x,t) = 1,11109 \times 10^{-5} f_1(x) + 0,99998 f_2(x).$$

Let x_1 and x_2 be two solutions such as:

$$f_1(x_1) = 5 \times f_1(x_{t-1}) \quad \text{and} \quad f_2(x_1) \approx f_2(x_{t-1})$$

and

$$f_1(x_2) \approx f_1(x_{t-1}) \quad \text{and} \quad f_2(x_2) = 5 \times f_2(x_{t-1}).$$

Then we have:

$$h(x_1,t) = w_1(t) f_1(x_1) + w_2(t) f_2(x_1) \approx 5.999885 \times 10^{-1}$$

and

$$h(x_2,t) = w_1(t) f_1(x_2) + w_2(t) f_2(x_2) \approx 5.999881 \times 10^{-1}.$$

It is clear that the probabilities to select x_1 and x_2 are nearly equal. The solution x_2, thus, has almost the same chance as x_1 to be selected in the next generation of the GA, which is not the case with the choice of fixed weights.

IV. Experimentation: Five Test Problems

We test our algorithm on five optimization problem given in the literature. In these experimentations:

- We will compare the two terms of the objective function using, first, classical weighted method and secondly our dynamic weighted method;
- We will observe the behavior of solutions for nine iterations of GA.

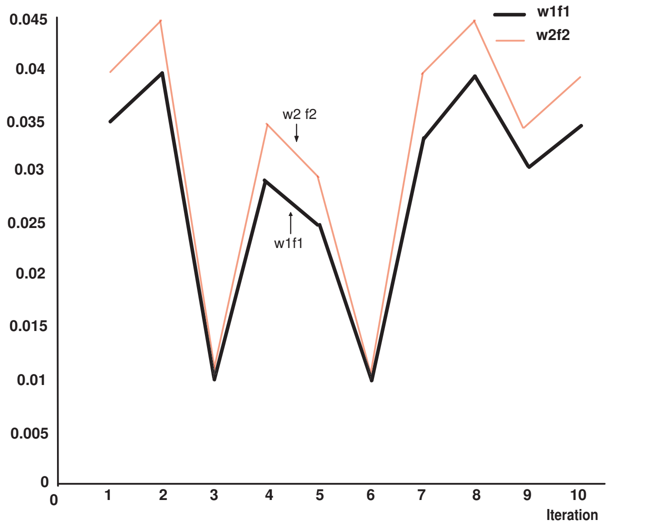

Fig. 1. SCH1: comparison between $w_1 f_1$ and $w_2 f_2$ for 10 tests using dynamic weighted method

A. Test Function: SCH1

The function $SCH1$ was proposed by Schaffer [30] and cited in [10] and [8] for instance.

$$SCH1: \begin{array}{ll} \text{Minimize} & f_1(x) = x^2 \\ \text{Minimize} & f_2(x) = (x-2)^2 \\ S.t. & -5 \le x \le 5. \end{array}$$

$SCH1$ has a Pareto optimal set in $0 \le x \le 2$. We transform this problem into the mono-objective optimization problem.

$$\begin{array}{ll} \text{Maximize} & h(x) = -w_1 f_1(x) - w_2 f_2(x) \\ S.t. & -5 \le x \le 5. \end{array}$$

The objective functions f_1 and f_2 satisfy: $0 \le f_1 \le 25$ and $0 \le f_2 \le 49$. f_1 has the minimum in the vicinity of $x = 0$. In this case, the function f_2 is far from its minimum. On the other hand, f_2 has the minimum in the neighborhood of $x = 2$. In this case, the function f_1 is far from its minimum.
We apply the GA with dynamic weights presented in Section III-B. The parameters of GA are set as follows: crossover probability $p_c = 0.4$, mutation probability $p_m = 0.01$, population size 20, and maximum number of generations 30. The experiment was conducted on ten times. The results are presented in Table I. The second column specifies the choice of w_i and the third column shows the best solution after ten trials of each experiment. The fourth column tests if the solution belongs to the Pareto set or not. The right hand part of Table I calculates $w_1 f_1$ and $w_2 f_2$ of the last iteration.
When comparing the two terms of the objective function h ($w_1 f_1$ and $w_2 f_2$), using dynamic weighted method, we find that our algorithm ensures an equitable treatment of each objective. Indeed, in 10 experiments, we always have $|w_1 f_1 - w_2 f_2| \le 0.006$, see Fig 1.
However, when we use classical weighted method to compare $w_1 f_1$ and $w_2 f_2$, we see that the difference between the two terms is very important. Indeed, this difference reached $|w_1 f_1 - w_2 f_2| \ge 0.6$ in the sixth iteration, see Fig 2.

TABLE I
RESULTS FOR TEST PROBLEM SCH1

Method	w_i	Best solution x	Pareto	Time in Secondes	$w_1 f_1$	$w_2 f_2$
classical Weighted Sum	$w_1 = 0,7$ and $w_2 = 0,3$	0,5999	Yes	0,67	0,2519	0,5880
	$w_1 = 0,3$ and $w_2 = 0,7$	1,4001	Yes	0,67	0,5880	0,2518
Dynamic Weighted Sum	Dynamic	1,094	Yes	0,86	0,0110	0,0113

TABLE II
RESULTS FOR TEST PROBLEM SCH2

Method	w_i	Best solution x	Pareto	Time in Secondes	$w_1 f_1$	$w_2 f_2$
Classical Weighted Sum	$w_1 = 0,7$ and $w_2 = 0,3$	4,00006	Yes	0,91	0.00048	0.29994
	$w_1 = 0.3$ and $w_2 = 0.7$	4.785	Yes	0,91	0.23571	0.03213
Dynamic Weighted Sum	Dynamic	4.2372	Yes	1,2	0.2609	0.2621

TABLE III
RESULTS FOR TEST PROBLEM MIN-EX

Method	w_i	Best solution x_1	Best solution x_2	Time in Secondes	$w_1 f_1$	$w_2 f_2$
Classical Weighted Sum	$w_1 = 0.7$ and $w_2 = 0.3$	0.949	0.053	1	0.664	0.332
	$w_1 = 0.3$ and $w_2 = 0.7$	0.909	0.048	1	0.27	0.80
Dynamic Weighted Sum	Dynamic	0.252	0.005	1.09	0.21	0.21

TABLE IV
RESULTS FOR TEST PROBLEM CONST-MIN-EX

Method	w_i	Best solution x_1	Best solution x_2	Time in Secondes	$w_1 f_1$	$w_2 f_2$
Classical Weighted Sum	$w_1 = 0.7$ and $w_2 = 0.3$	0.74	0.16	1.34	0.24	0.88
	$w_1 = 0.3$ and $w_2 = 0.7$	0.82	0.04	1.34	0.51	0.47
Dynamic Weighted Sum	Dynamic	0.66	0.017	1.4	0.46	0.46

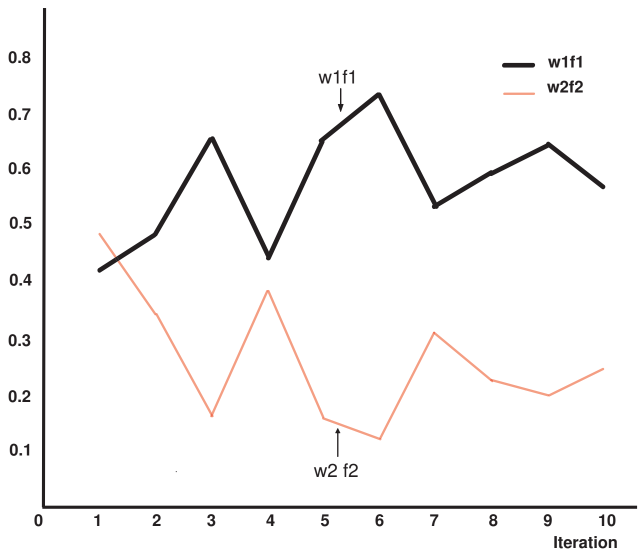

Fig. 2. SCH1: comparison between $w_1 f_1$ and $w_2 f_2$ for 10 tests using classical weighted method ($w_1 = 0.3$ and $w_2 = 0.7$)

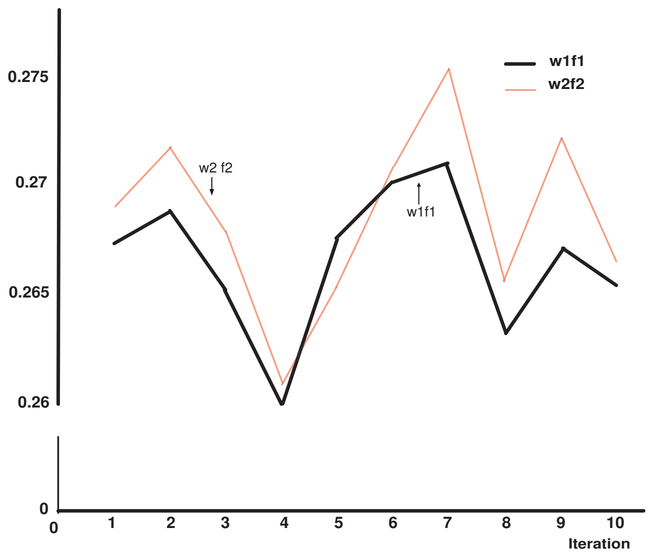

Fig. 3. SCH2: comparison between $w_1 f_1$ and $w_2 f_2$ for 10 tests using dynamic weighted method

B. Test Function: SCH2

This problem proposed by Schaffer [30] and cited in [10] for instance, aims to minimize two cost functions with a single variable. It is given by:

$$SCH2: \begin{array}{l} Minimize\ f_1(x) = \begin{cases} -x \text{ if } x \leq 1 \\ x-2 \text{ if } 1 < x \leq 3 \\ 4-x \text{ if } 3 < x \leq 4 \\ x-4 \text{ if } x > 4 \end{cases} \\ Minimize\ f_2(x) = (x-5)^2 \\ -5 \leq x \leq 10 \end{array}$$

TABLE V
RESULTS FOR TEST PROBLEM MAX-EX

Method	w_i	Best solution x_1	Best solution x_2	Time in Secondes	$w_1 f_1$	$w_2 f_2$
Classical Weighted Sum	$w_1 = 0.7$ and $w_2 = 0.3$	0.655	0	1,24	0.3112	17.5422
	$w_1 = 0.3$ and $w_2 = 0.7$	0.9992	0.0003	1,24	0.0302	41.2992
Dynamic Weighted Sum	Dynamic	0.3450	0.0120	1,85	0.9570	0.9339

TABLE VI
RESULTS FOR UMTS PROBLEM

Method	w_i	$w_1 f_1$	$w_2 f_2$	Time in Seconds
Classical Weighted Sum	$w_1 = 0.5$ and $w_2 = 0.5$	47.5	8.2	16,04
Dynamic Weighted Sum	Dynamic	54.97	55.88	17,93

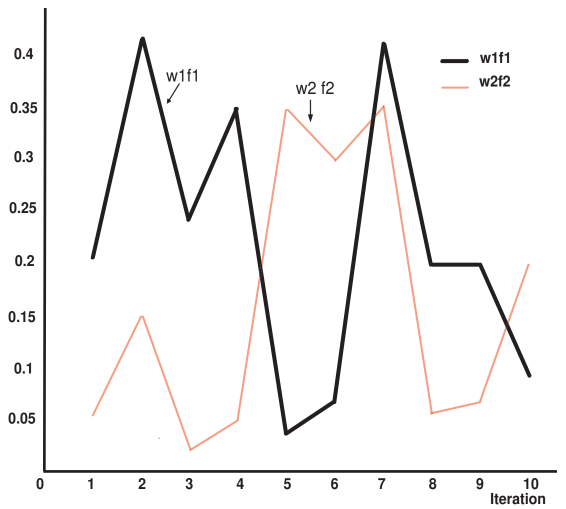

Fig. 4. SCH2: comparison between $w_1 f_1$ and $w_2 f_2$ for 10 tests using classical weighted method ($w_1 = 0.3$ and $w_2 = 0.7$)

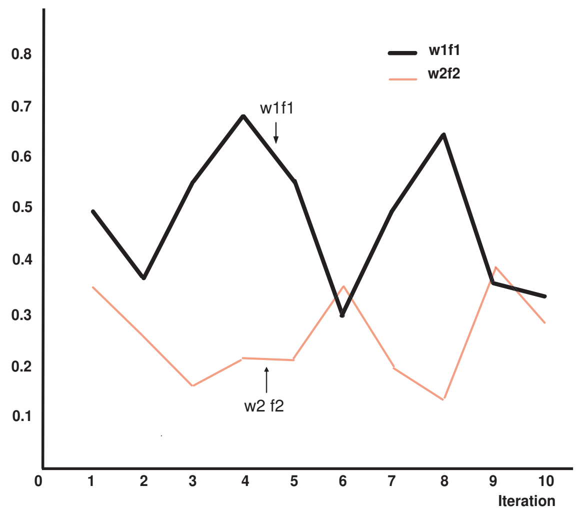

Fig. 6. Min-Ex: comparison between $w_1 f_1$ and $w_2 f_2$ for 10 tests using classical weighted method ($w_1 = 0.7$ and $w_2 = 0.3$)

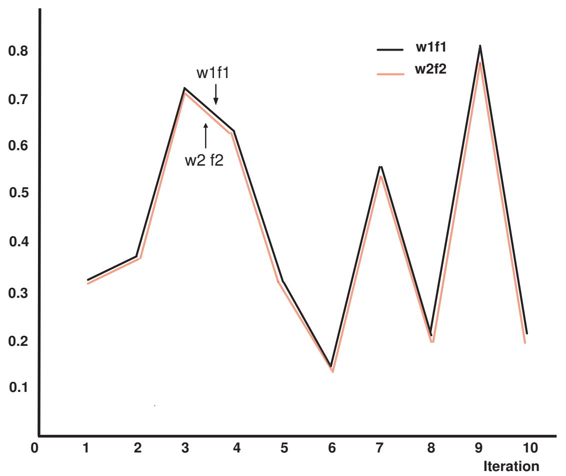

Fig. 5. Min-Ex: comparison between $w_1 f_1$ and $w_2 f_2$ for 10 tests using dynamic weighted method

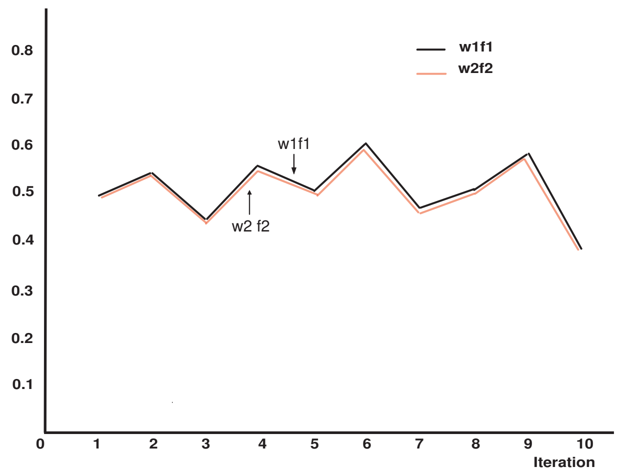

Fig. 7. Const-Min-Ex: comparison between $w_1 f_1$ and $w_2 f_2$ for 10 tests using dynamic weighted method

The Pareto optimal set is formed by two discontinuous areas: $x \in [1, 2] \cup [4, 5]$, which results in a discontinuous Pareto optimal front. This problem is transformed into the mono-objective optimization problem.

$$\text{Maximize } h(x) = -w_1 f_1(x) - w_2 f_2(x)$$
$$S.t. \quad -5 \le x \le 10.$$

The objective functions f_1 and f_2 satisfy: $-1 \le f_1 \le 6$ and $0 \le f_2 \le 100$. f_1 has the minimum in the vicinity of $x = 1$. In this case, the function f_2 is far from its minimum. On the other hand, f_2 has the minimum in the neighborhood of $x = 5$. In this case, the function f_1 is far from its minimum. We apply the GA with dynamic weights. We run our algorithm ten times. The maximum number of generations is 40.

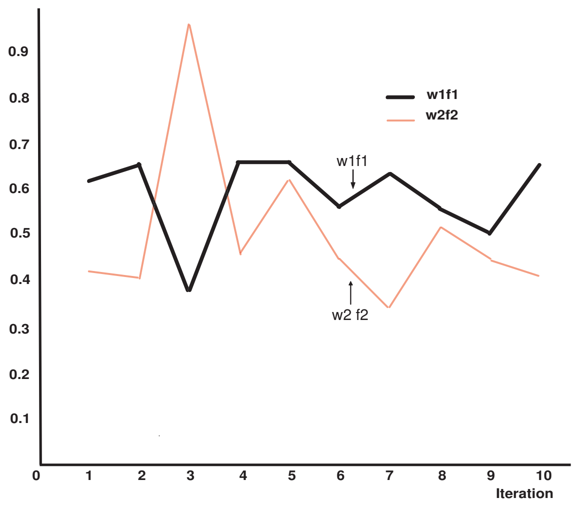

Fig. 8. Const-Min-Ex: comparison between $w_1 f_1$ and $w_2 f_2$ for 10 tests using classical weighted method ($w_1 = 0.7$ and $w_2 = 0.3$)

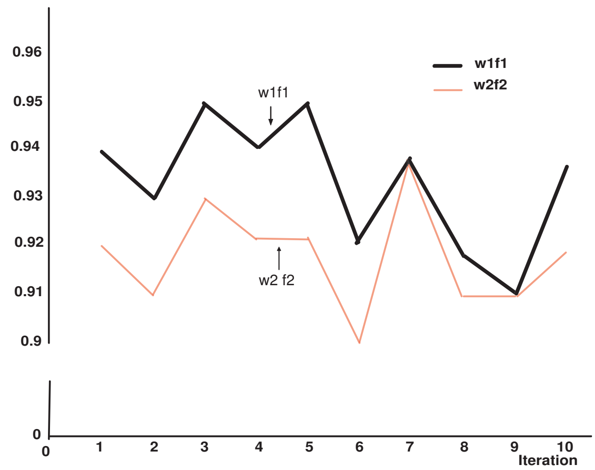

Fig. 9. Max-Ex: comparison of $w_1 f_1$ and $w_2 f_2$ for 10 tests using dynamic weighted method

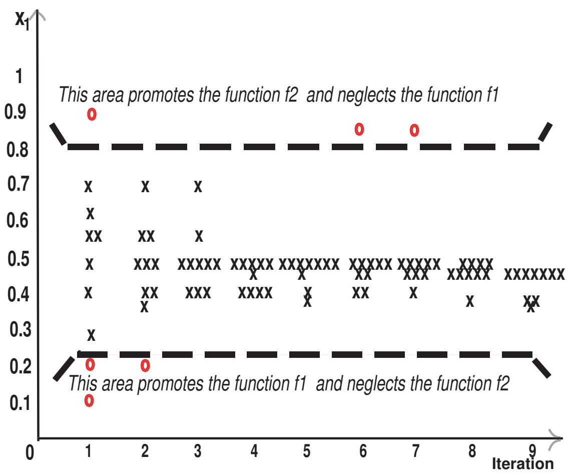

Fig. 10. Max-Ex: The behavior of solutions for nine iterations

The results are presented in Table II. When we compare the two terms $w_1 f_1$ and $w_2 f_2$ of the objective function h, using dynamic weighted method, we find that our algorithm, in this example also, ensures almost an equitable treatment of each objective. Indeed, in 10 experiments, we always have $|w_1 f_1 - w_2 f_2| \leq 0.008$, see Fig 3.
However, when we use classical weighted method to compare $w_1 f_1$ and $w_2 f_2$, we see that the difference between the two terms is very important. Indeed, this difference reached $|w_1 f_1 - w_2 f_2| \geq 0.3$ in the fifth iteration, see Fig 4.

C. Test Function: Min-Ex

Min-Ex is a minimization problem with two cost functions and two variables. It was proposed by Deb [9] and cited in [10] for instance. Its solution is a convex Pareto optimal front. It is given by:

$$\text{Min-Ex}: \begin{cases} \text{Minimize } f_1(x_1, x_2) = x_1 \\ \text{Minimize } f_2(x_1, x_2) = (1 + x_2)/x_1 \\ 0.1 \leq x_1 \leq 1 \\ 0 \leq x_2 \leq 5. \end{cases}$$

Since the optimal Pareto value of x_2 is exactly 0, then the analytical Pareto front is written as:

$$f_2(x_1, x_2) = \frac{1}{f_1(x_1, x_2)}$$

The problem Min-Ex is transformed into the mono-objective optimization problem.

$$\begin{cases} \text{Maximize } h(x_1, x_2) = -w_1 f_1(x_1, x_2) - w_2 f_2(x_1, x_2) \\ 0.1 \leq x_1 \leq 1 \\ 0 \leq x_2 \leq 5. \end{cases}$$

We apply the GA with dynamic weights presented in Section III-B. The parameters of GA are set as follows: crossover probability $p_c = 0.4$, mutation probability $p_m = 0.01$, population size 20, and maximum number of generations 30. The experiment was conducted on ten times. The results are presented in Table III.
When we compare the two terms $w_1 f_1$ and $w_2 f_2$ of the objective function h, using dynamic weighted method, we find that our algorithm ensures an equitable treatment of each objective. Indeed, in 10 experiments, we always have $|w_1 f_1 - w_2 f_2| \leq 0.0001$, see Fig 5.
However, when we use classical weighted method to compare $w_1 f_1$ and $w_2 f_2$, we see that the difference between the two terms is very important. Indeed, this difference reached $|w_1 f_1 - w_2 f_2| \geq 0.5$ in the eighth iteration, see Fig 6.

D. Test Function: Const-Min-Ex

Const-Min-Ex is a minimization problem with two cost functions and two variables with two constraints. It was proposed by Deb [9] and cited in [10] for instance. It is given by:

$$\text{Const-Min-Ex}: \begin{cases} \text{Minimize } f_1(x_1, x_2) = x_1 \\ \text{Minimize } f_2(x_1, x_2) = (1 + x_2)/x_1 \\ x_2 + 9x_1 \geq 6 \\ -x_2 + 9x_1 \geq 1 \\ 0.1 \leq x_1 \leq 1 \,;\, 0 \leq x_2 \leq 5. \end{cases}$$

The problem Const-Min-Ex is transformed into the mono-objective optimization problem.

$$\text{Maximize } h(x_1, x_2) = -w_1 f_1(x_1, x_2) - w_2 f_2(x_1, x_2)$$
$$x_2 + 9x_1 \geq 6$$
$$-x_2 + 9x_1 \geq 1$$
$$0.1 \leq x_1 \leq 1 \;;\; 0 \leq x_2 \leq 5.$$

We apply the GA with dynamic weights presented in Section III-B. The parameters of GA are set as follows: crossover probability $p_c = 0.5$, mutation probability $p_m = 0.01$, population size 20, and maximum number of generations 40. The experiment was conducted on ten times. The results are presented in Table IV.

When we compare the two terms $w_1 f_1$ and $w_2 f_2$ of the objective function h, using dynamic weighted method, we find that our algorithm ensures an equitable treatment of each objective. Indeed, in 10 experiments, we always have $|w_1 f_1 - w_2 f_2| \leq 0.0001$, see Fig 7.

However, when we use classical weighted method to compare $w_1 f_1$ and $w_2 f_2$, we see that the difference between the two terms is very important. Indeed, this difference reached $|w_1 f_1 - w_2 f_2| \geq 0.57$ in the third iteration, see Fig 8.

E. Test Function: Max-Ex

Max-Ex is a maximization problem with two cost functions and two variables. It was proposed by Deb [9] and cited in [10] for instance. Its solution is a non-convex Pareto optimal front. It is given by:

$$\text{Max-Ex}: \begin{cases} \text{Maximize } f_1(x_1, x_2) = 1.1 - x_1 \\ \text{Maximize } f_2(x_1, x_2) = 60 - ((1 + x_2)/x_1) \\ 0.1 \leq x_1 \leq 1 \\ 0 \leq x_2 \leq 5. \end{cases}$$

Max-Ex has a Pareto optimal set in $x_2 = 0$ and $0.1 \leq x_1 \leq 1$. Since the optimal Pareto value of x_2 is exactly 0, then the objective functions f_1 and f_2 satisfy: $0.1 \leq f_1 \leq 1$ and $50 \leq f_2 \leq 59$. The problem Max-Ex is transformed into the mono-objective optimization problem.

$$\text{Maximize } h(x_1, x_2) = w_1 f_1(x_1, x_2) + w_2 f_2(x_1, x_2)$$
$$0.1 \leq x_1 \leq 1$$
$$0 \leq x_2 \leq 5.$$

Using classical weighted sum method, the fact that the objective function f_2 dominates f_1 is extremely probable. Now we apply our algorithm presented in Section III-B to Max-Ex problem. For genetic algorithm, the parameters are set as follows: crossover probability $p_c = 0.4$, mutation probability $p_m = 0.01$, population size 20, and maximum number of generations 40. The experiment was conducted on ten times. First, we will compare the two terms of the objective function h. The results are presented in Table V. When comparing the two terms $w_1 f_1$ and $w_2 f_2$, we find that our algorithm ensures almost an equitable treatment of each objective. Indeed, in the 10 experiments, we always have $|w_1 f_1 - w_2 f_2| \leq 0.03$, see Fig 9.

The second task is to observe the behavior of individuals of the population in each generation of GA. For simplicity, we consider a population of only 10 individuals and we limit the observation for only nine iterations. The results are presented in Fig 10. In the first iteration, the solutions (individuals) are randomly distributed on the interval $[0.1, 1]$. We see that the two solutions $x_1^1 = 0.1$ and $x_1^2 = 0.2$ belong to the area which promotes the function f_1 and neglects the function f_2, and the solution $x_1^3 = 0.9$ belongs to the region which promotes the function f_2 and neglects the function f_1. In the second iteration, the solutions x_1^1 and x_1^3 disappear, and in the third iteration, the three solutions disappear. In the sixth iteration, a solution $x = 0.85$ appears because of the crossover and mutation operators. This solution disappears in the eighth iteration. In the ninth iteration, only the solutions that optimize simultaneously both functions f_1 and f_2 resist.

V. Application: UMTS Base Station Location Planning Problem

In this section, we address the problem of planning the universal mobile telecommunication system (UMTS) base stations location presented in [2], [3], [14].

A. Problem Statement and Model Presentation

Consider a territory to be covered by a UMTS service. Let $S = \{1, ..., m\}$ be a set of candidate sites (CS) where a base station (BS) can be installed and $I = \{1, ..., n\}$ be a set of test points (TPs). Each base station BS_j, $j \in S$, has a cost of installation denoted by $\mathbf{c_j}$. We denote by $\mathbf{u_i}$ the required number of simultaneously active connections for a TP of index i (TP_i). Let us define the two following classes of decision variables:

$$y_j = \begin{cases} 1 & \text{if a } BS \text{ is installed in a site } j, \\ 0 & \text{otherwise,} \end{cases} \quad \text{for } j \in S, \tag{8}$$

and

$$x_{ij} = \begin{cases} 1 & \text{if a } TP_i \text{ is assigned to a } BS_j, \\ 0 & \text{otherwise,} \end{cases} \tag{9}$$

for $i \in I$ and $j \in S$.

We consider a power-based PC mechanism. Suppose we have directive BSs with three identical 120 degree sectors and with an omnidirectional antenna diagram along the horizontal axis. Let the index set $I_j^\sigma \subseteq I$ denotes the set of all TPs that fall within the sector σ of the BS installed in the candidate site CS_j. Since we wish to maximize the total trafic covered and minimize the total installation cost subjected to some constraints, then the problem can be expressed as [2], [3]:

$$\begin{aligned} &\text{Maximize } f_1(x) = \sum_{i=1}^{n} \sum_{j=1}^{m} u_i x_{ij}, \\ &\text{Minimize } f_2(y) = \sum_{j=1}^{m} c_j y_j, \end{aligned} \tag{10}$$

subject to:

$$\textstyle\sum_{j=1}^{m} x_{ij} \leq 1 \,, \quad i \in I, \tag{11}$$

$$x_{ij} \leq \min\{1, \tfrac{g_{ij} P_{max}}{P_{target}}\} y_j, \quad i \in I \,,\, j \in S, \tag{12}$$

$$y_j \sum_{i \in I_j^\sigma} \sum_{t=1}^{m} (\frac{u_i g_{ij}}{g_{it}} x_{it} - 1) \leq \frac{SF}{SIR_{min}}, j \in S, \sigma \in \Sigma, \tag{13}$$

$$x_{ij} \,,\, y_j \in \{0, 1\}, \quad i \in I, j \in S. \tag{14}$$

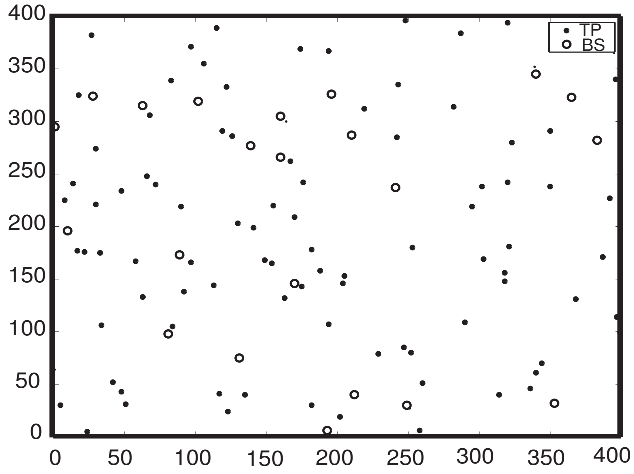

Fig. 11. Location of 95 TPs and 22 BSs in a service area of $0.4 \times 0.4 (Km)$

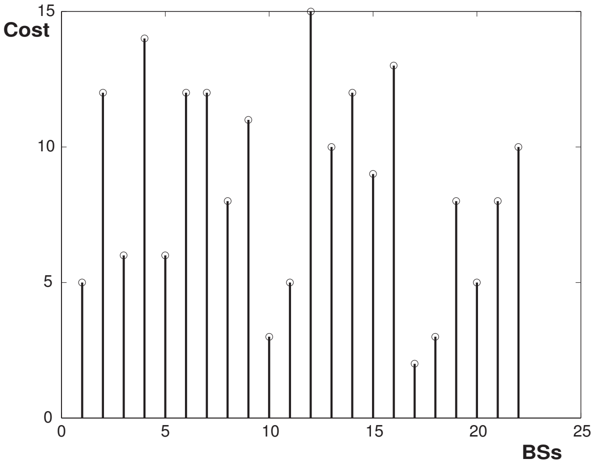

Fig. 12. Costs of 22 BSs

Where the propagation factor of the radio link between a TP_i and a candidate site CS_j is given by:

$$g_{ij} = (10^{\frac{L_u(d_{ij})}{10}})^{-1},$$

where the attenuation L_u is calculated by the Hata's propagation model presented in [16].

The multi-objective problem (10) can be transformed into a mono-objective one as follows:

$$\text{Maximize} \quad w_1 \sum_{i=1}^{n} \sum_{j=1}^{m} u_i x_{ij} - w_2 \sum_{j=1}^{m} c_j y_j, \qquad (15)$$

Subject to the constraints (11), (12), (13) and (14), where w_1 and w_2 are the weights of f_1 and f_2.

B. Data Description and Computational Results

We consider a rectangular service area, a number of candidate sites (CSs) in which to locate omnidirectional antennas, and a number of TPs. Using a pseudorandom number generator each CS and each TP is assigned a position with uniform distribution in the service area. We consider an instance of an urban environment. The simulation parameters are:

- Size of the service area (in *km*): 0.4×0.4;
- Number of TPs: 95, and number of BSs: 22;

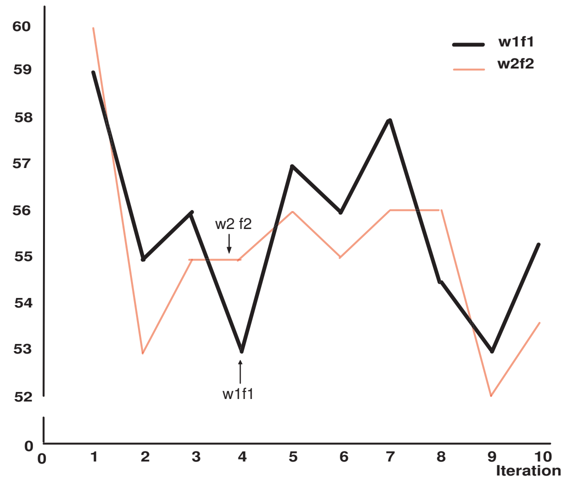

Fig. 13. UMTS: comparison of $w_1 f_1$ and $w_2 f_2$ for 10 tests

- $\mathbf{u_i} = 1$,
- Signal frequency F = 2000 *MHz*;
- Height of the mobile station H_m = 1 *meter*;
- Height of the base H_b =10 *meters*;
- Target power P_{target} = -100 *dBm*;
- Maximum power P_{max} = 30 *dBm*;
- Ratio between spread signal and user rate $SF = 128$;
- Target signal-to-Interference Ratio $SIR_{target} = 6$ *dB*;
- $SIR_{min} = 0.03125$ *dB*;
- Costs $\mathbf{c}_i$: are taken randomly between 1 and 20 units.

Figure 11 illustrates the distribution of the TPs and BSs in the area service, and Figure 12 shows the cost of installation of each BS. The GA parameters are set as follows: crossover probability $p_c = 0.4$, mutation probability $p_m = 0.01$, population size 30, and maximum number of generations 1000. We run our algorithm ten times. The best solution consists of installing 21 BSs instead of 22, which cover 93 TPs among 95, with a cost equal to 179. Then we have a gain of 10 (approximately 5.3% of costs of BSs), since the cost of installing all BSs is 189. When comparing the two terms of the objective function (15), we find that our algorithm ensures almost an equitable treatment of each objective, see Fig 13 and Table VI.

VI. Conclusion

In this paper we are interested in the weighted sum method where a multi-objective problem is transformed into a mono-objective one. The problem lies in the right selection of the weights to characterize the decision makers preferences. Here, we tried to solve the following main problem: How to ensure an equitable treatment of each objective function?

In order to solve this problem we have introduced a solution method based on a genetic algorithm approach which automates the choice of the weights by varying them at each iteration of the algorithm. The utility of choosing dynamic weights lies mainly in the two following points:

- It automates the choice of the weights. Therefore we do not need to define these factors a priori.
- It ensures an equitable treatment of each objective function.

Our algorithm is tested on five academic problems and is applied to a UMTS base station location planning problem. The obtained results validate the method proposed in this paper.

REFERENCES

[1] B. Akay, "Synchronous and asynchronous Pareto-based multi-objective Artificial Bee Colony algorithms," *Journal of Global Optimization*, Volume 57, Issue 2, 415-445, October 2013.

[2] E. Amaldi , A. Capone and F. Malucelli, "Planning UMTS Base Station Location: Optimization Models With Power Control and Algorithms," *IEEE Transactions on wireless communications*, 2:939-952, 2003.

[3] E. Amaldi , A. Capone, F. Malucelli and F. Signori, "Radio Planning and Optimization of W-CDMA Systems," *Personal Wireless Communications*, 437-447, 2003.

[4] O. B. Augusto, F. Bennis and S. Caro, "A New Method for Decision Making in Multi-Objective Optimization Problems," *Pesquisa Operacional*, 32(2), 331-369, 2012.

[5] S. Bandyopadhyay, S. Saha, U. Maulik and K. Deb, "A Simulated Annealing-Based Multiobjective Optimization Algorithm: AMOSA," *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION*, VOL. 12, NO. 3, JUNE 2008.

[6] L. C. Cagnina and S. C. Esquivel, "Solving Hard Multiobjective Problems with a Hybridized Method," *JCS&T*, Vol.10, No. 3, October 2010.

[7] C. A. C. COELLO, "List of References on Evolutionary Multi-objective Optimization, April 2010." *in URL: http://www.lania.mx /ccoello/EMOO/EMOObib.html*

[8] K. Deb , A. Pratap , S. Agarwal and T. Meyarivan, "A Fast and Elitist Multi-objective Genetic Algorithm: NSGA-II," *IEEE Trans. Evol. Comp.*, Vol. 6, No. 2, 182-197, 2002.

[9] K. Deb , L. Thiele , M. Laumanns and E. Zitzler, "Scalable Test Problems for Evolutionary Multi-Objective Optimization," *Tech. Rep. 112*, Zurich, Switzerland, 2001.

[10] M. Ejday, *Optimisation multi objectifs à base de méta modele pour les procédés de mise en forme*, 3rd ed. Ph.D. Thesis, cole nationale superieure des mines, Paris, 2011.

[11] S. Elaoud, T. Loukil and J. Teghem, "The Pareto fitness genetic algorithm: Test function study," *European Journal of Operational Research*, 177, 1703-1719, 2007.

[12] C. M. Fonseca and P. J. Fleming, "Multiobjective optimization" *IOP Publishing*, Bristol, U.K., 2000.

[13] C. M. Fonseca and P. J. Fleming, "Multiobjective genetic algorithms," *IEE colloquium on Genetic Algorithms for Control Systems Engineering*, 6:1-5, 1993.

[14] M. Gabli , E. M. Jaara and E. B. Mermri, "Planning UMTS Base Station Location Using Genetic Algorithm with a Dynamic Trade-Off Parameter," *Lecture Note of Computer Science, Springer, Heidelberg*, No 7853, 120-134, 2013.

[15] D. E. Goldberg, *Genetic algorithms in search, optimization, and machine learning*, 3rd ed. Addison-Wesley, 1989.

[16] M. Hata, "Empirical Formula for Propagation Loss in Land Mobile Radio Services," *IEEE Transactions on Vehicular Technology*, 29:317-325, 1980.

[17] J. Holland, "Outline for a logical theory of adaptive systems," *Journal of the Association of Computing Machinery*, 3, 1962.

[18] J. Horn , N. Nafpliotis and D. E. Goldberg, "A niched Pareto genetic algorithm for multiobjective optimization," *IEEE world congress on computational intelligence*, 82-87, 1994.

[19] S. T. Hsieh, S. Y. Chiu and S. J. Yen "An Improved Multi-Objective Genetic Algorithm for Solving Multi-objective Problems," *Applied Mathematics and Information Sciences*, 7, No. 5, 1933-1941, 2013.

[20] Y. Jin , T. Okabe and B. Sendhoff, "Adapting Weighted Aggregation for Multiobjective Evolution Strategies," *EMO 01 Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization*, 96-110, 2001.

[21] H. Meunier, *Algorithmes evolutionnaires paralleles pour l'optimisation multiobjectif de reseaux de telecommunications mobiles*, 3rd ed. PhD thesis, University of Sciences and Technologies, Lille, 2002.

[22] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, 3rd ed. Springer-Verlag, New York, 1994.

[23] T. Murata , H. Ishibuchi and H. Tanaka, "Multi-objective genetic algorithm and its applications to flowshop scheduling" *Computers & Industrial Engineering* 30(4), 957968, 1996.

[24] A. Nakibe, *Conception de metaheuristiques d'optimisation pour la segmentation d'images. Application à des images biomedicales.*, 3rd ed. PhD thesis, UFR of Sciences and Technology, University PARIS 12-VAL DE MARNE, 2007.

[25] G. Narzisi, *Classic Methods for Multi-Objective Optimization*,3rd ed. Courant Institute of Mathematical Sciences, New York University, 31 January 2008.

[26] D. Perez, S. Samothrakis and S. Lucas, "Online and Offline Learning in Multi-Objective Monte Carlo Tree Search," *Proceedings of the Conference on Computational Intelligence and Games (CIG)*, pp. 121-128, 2013.

[27] M. Sakawa, *Genetic Algorithm and Fuzzy Multiobjective Optimization*, 3rd ed. Kluwer Academic Publishers, 2001.

[28] R. Sarker, K. H. Liang and C. Newton, "A new multiobjective evolutionary algorithm," *European Journal of Operational Research*, 140(1), 1223, 2002.

[29] J. D. Schaffer, "Multiple objective optimization with vector evaluated genetic algorithms," *Proceedings of an international conference on genetic algorithms and their applications*, 93-100, 1985.

[30] J. D. Schaffer, *Some experiments in machine learning using vector evaluated genetic algorithms*, 3rd ed. Ph.D Thesis, Nashville, TN: Vanderbilt University, 1984.

[31] E. G. Talbi , M. Basseur , A. G. Nebro and E. Alba, "Multi-objective optimization using metaheuristics: non-standard algorithms," *International Transactions in Operational Research*, 19, 283-306, 2012.

[32] X. S. Yang, "Multiobjective firefly algorithm for continuous optimization," *Engineering with Computers*, Vol. 29, Issue 2, pp. 175184, 2013.

[33] X. S. Yang, M. Karamanoglua and X. He, "Multi-objective Flower Algorithm for Optimization," *Procedia Computer Science*, Vol. 18, 861868, 2013.

[34] J. Yang, Z. Zhang, L. chen and Y. Wang, "Optimization of a High Speed Gasoline Engine Using Genetic Algorithm," *SAE Technical Paper 2013-01-1626*, doi:10.4271/2013-01-1626, 2013.

[35] G. G. Yen and H. Lu, "Dynamic multiobjective evolutionary algorithm: adaptive cell-based rank and density estimation," *IEEE Transactions on Evolutionary Computation*, 7(3), 253274, 2003.

[36] H. Zhang "An Analysis of Multiple Particle Swarm Optimizers with Inertia Weight for Multi-objective Optimization," *IAENG International Journal of Computer Science*, Vol. 39 Issue 2, 190-199, 2012.

[37] Q. Zhang, W. Liu, E. Tsang and B. Virginas, "Expensive Multi-objective Optimization by MOEA/D with Gaussian Process Model," *Evolutionary Computation, IEEE Transactions on*, Vol. 14 , Issue. 3, 456-474, June 2010.

[38] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach," *IEEE transactions on evolutionary computation*, 257-271, 1999.

Mohammed Gabli was born in Oujda, Morocco. He is a PhD student of Laboratory of Research in Computer Science, Department of Mathematics and Computer Science, Faculty of Science, University Mohammed Premier, Oujda, Morocco.
He is an educative guidance counselor since 2006.
His research interests include metaheuristics, multi-objective optimization, education.

El Miloud Jaara is an Associate Professor at the Department of Mathematics and Computer Science, Faculty of Science, University Mohammed Premier, Oujda, Morocco.
He earned his PhD in parallelization of neural networks in 2000 from University Mohammed Premier, Oujda.
His research interests include the parallelization, the neural networks, image processing.

El Bekkaye Mermri is an Associate Professor at the Department of Mathematics and Computer Science, Faculty of Science, University Mohammed Premier, Oujda, Morocco.
He earned his PhD in Mathematics in 2002 from University Mohammed Premier, and his PhD in Artificial Complex Systems Engineering from Hiroshima University in 2006.
His research interests include numerical inequalities, computational intelligence, systems optimization.