A Comparison of Decoding Strategies for the 0/1 Multi-objective Unit Commitment Problem

Sophie Jacquin, Lucien Mousin, Igor Machado, El-Ghazali Talbi, Laetitia Jourdan

HAL Id: hal-01110753

https://hal.archives-ouvertes.fr/hal-01110753

Submitted on 28 Jan 2015

A Comparison of Decoding Strategies for the 0/1 Multi-objective Unit Commitment Problem

Sophie Jacquin[2,3], Lucien Mousin[3], Igor Machado[1,3], El-Ghazali Talbi[2,3], and Laetitia Jourdan[2,3]

[1] University of Flumisende, Brasil
[2] Inria Lille - Nord Europe, DOLPHIN Project-team, France
[3] Université Lille 1, LIFL, UMR CNRS 8022, France

Abstract. In the single objective Unit Commitment Problem (UCP) the problem is usually separated in two sub-problems : the commitment problem which aims to fix the on/off scheduling of each unit and the dispatching problem which goal is to schedule the production of each turned on unit. The dispatching problem is a continuous convex problem that can easily be solved exactly. For the first sub-problem genetic algorithms (GA) are often applied and usually handle binary vectors representing the solutions of the commitment problem.Then the solutions are decoded in solving the dispatching problem with an exact method to obtain the precise production of each unit. In this paper a multi-objective version of the UCP taking the emission of gas into account is presented. In this multi objective UCP the dispatching problem remains easy to solve whereas considering it separatly remains interesting. A multi-objective GA handling binary vectors is applied. However for a binary representation there is a set of solutions of the dispatching problem that are pareto equivalent. Three decoding strategies are proposed and compared. The main contribution of this paper is the third decoding strategy which attaches an approximation of the Pareto front from the associated dispatching problem to each genotypic solution. It is shown that this decoding strategy leads to better results in comparison to the other ones.

Keywords: UCP, metaheuristics, heuristic, multi-objective optimization

1 Introduction

The UCP is used to find the scheduling of commissioning and production of generating units that minimizes the production cost. However, environmental protection has become a major issue. In response to the growth of the negative impacts on the environment and due to the growing importance of environmental interest in society, governments have developed and implemented laws or technical standards in order to reduce the negative impacts of human activity on the environment. For this reason new modeling of the UCP taking into account some limitation constraints on gas emissions [2] has been proposed. A bi-objective model has also been proposed, but is usually solved by reducing the two objectives to one, using the weighted sum approach [17, 13]. In [15], a multi-objective version of the UCP is solved with a genetic algorithm, but it does not directly exploit the concept of Pareto dominance. The authors applied a classical genetic algorithm, with the exception that the selection process is based on a specific version of the tournament selection. Two individuals are randomly selected from the population and a stochastic competition of the objective that are chosen randomly, is performed to determine the winner that will survive into the next generation.

The UCP can be separated into two sub-problems. The commitment problem which is to give the on/off scheduling of each unit and then the dispatching problem which is to give the exact production for each turned on unit. In the single objective case, the dispatching problem is a quadratic continuous problem which is easy to solve exactly. For this reason evolutionary algorithms generally handle binary vectors giving the on/off scheduling of each unit to solve the single-objective UCP [3, 7, 8, 10, 16]. Then to obtain a complete description of the solution (the phenotypic solution), the production of each turned on unit is determined optimally using a λ-iteration method [14]. This binary representation takes advantage of a real one because the search space is considerably reduced. For the multi-objective the dispatching problem remains interesting to exploit separately because it has still good properties. All the solutions of the dispatching problem are supported and can be found by the scalarizing method. This method is to transform the problem with two objectives f_1 and f_2 into a single objective problem optimizing the function $\lambda f_1 + (1 - \lambda) f_2$, where $\lambda \in [0, 1]$. The quality of the results found in the single objective case and the fact that the multi-objective version of the dispatching problem has good properties lead us to a specific interest in considering a similar two level method of resolution for the multi-objective UCP.

In this paper a multi-objective GA based on NSGA-II [4] is proposed. This GA handle binary vectors representing the solutions of the commitment problem. Then the phenotypic solutions corresponding to the production of each unit has to be find by a decoding method. Since the dispatching problem is also a multi-objective problem there are many candidates of phenotypic solutions for one genotypic solution. Therefore three different decoding methods are proposed and compared. The first one is to construct the phenotypic solution by solving the dispatching problem by optimizing the function $f_1 + f_2$. The second is to add a real λ into the genotypic representation of a solution. The phenotypic solution is decoded by solving the dispatching problem minimizing $\lambda f_1 + (1 - \lambda) f_2$. The third is to associate a set of Pareto equivalent solutions of the corresponding multi-objective dispatching problem to each genotypic solution. For the last decoding system, the process of fitness and diversity assignment of NSGA-II has to be adapted.The fact that many phenotypic solutions are attached to a single genotypic solution must be taken into account. The main contribution of this paper is to show that the multi decoding embedded approach has an advantage over the two other less complex decoding systems that are proposed.

The paper is organized as follows. In the next section the multi-objective UCP is described in details. Then the three solving methods corresponding to the three binary genetic algorithms using the different decoding strategies are presented. Finally the experimental process and the obtained results are presented and discussed before the conclusion section.

2 Multi-objective Unit Commitment Problem - MO-UCP

In this section, the MO-UCP is presented in details. This problem is the same as the classical UCP but an objective is added to take into account the gas emission of SO_2 and CO_2.

2.1 Unit Commitment Problem

Unit Commitment Problem (UCP) [8] is to schedule generating units online or offline over a scheduling horizon. The goal is to minimize the power production cost while satisfying a set of operational constraints. The production cost includes the fuel and start-up costs. Constraints are capacity of production of each units, minimum up/down time and spinning reserve. UCP is usualy modeled as as a mixed integer non-linear problem. It consists of binary variables $u_{i,t}$ that takes value 1 if a unit i is turned on at time t and 0 otherwise, continuous variables $p_{i,t}$ that denotes their prodduction amounts. It is a very complex problem to solve because of its enormous dimension, a non-linear objective function, and time-dependent constraints. Indeed, the UCP is a NP-complete problem [6].

2.2 Multi-objective UCP

The first objective is the same as in single-objective UCP. It minimizes the cost of production. This production cost is divided into two components, the fuel cost and the start up cost. For a system of N units and a time horizon of T periods, the objective function can be described as follows:

$$f_1(u,p) = \sum_{t=1}^{T} \sum_{i=1}^{N} FC_i(p_{i,t}) \times u_{i,t} + CS_i(T_{i,t-1}^{off}) \times (1 - u_{i,t-1})u_{i,t},$$

where :

- FC_i is the fuel cost function of the unit i, which is modeled by a quadratic function:

  $$FC(p_{i,t}) = a_{1,i} + a_{2,i} \times p_{i,t} + a_{3,i} \times p_{i,t}^2,$$

  where $a_{1,i}$, $a_{2,i}$ and $a_{3,i}$ are real cost coefficients for the unit i.
- CS_i is the start-up cost for unit i, which depends on time $T_{i,t-1}^{off}$ the unit i has been turned off at time $t-1$:

  $$CS_i(T_{i,t-1}^{off}) = \begin{cases} CS_{cold} \;\; if \;\; T_{i,min}^{off} + T_{cs,i} \leq T_{i,t-1}^{off} \\ CS_{hot} \; else \end{cases},$$

where $T_{i,min}^{off} + T_{cs,i}$ is the time it takes the unit i to become cold.

The second objective function measures the SO_2 and CO_2 emissions:

$$f_2(p,u) = \sum_{t=1}^{T} \sum_{i,u_{i,t}=1} b_{0,i} + b_{1,i}p_{i,t} + b_{2,i}p_{i,t}^2$$

The coefficients $b_{0,i}$, $b_{1,i}$, $b_{2,i}$ used in this paper are the ones proposed in [13].

The minimization of the objectives of the UCP is subject to the following system and unit constraints:

1. Power balance constraints:

   $$\textstyle\sum_{i=1}^{N} p_{i,t}u_{i,t} = D_t \quad \forall t \ ,$$

   where D_t is a real number giving the load demand at time t.
2. Spinning reserve constraints:

   $$\textstyle\sum_{i=1}^{N} p_{i,max}u_{i,t} \geq D_t + R_t \quad \forall t \ ,$$

   where R_t is a real giving the minimal reserve at time t.
3. Unit output constraints:

   $$p_{i,min} \leq p_{i,t} \leq p_{i,max} \quad \forall t,$$

   where $p_{i,min}$ and $p_{i,max}$ are the lower and upper bounds on the energy production of unit i respectively.
4. Minimum up time limit:

   $$T_{i,t-1}^{on} \geq T_{i,min}^{on} \times (1 - u_{i,t})u_{i,t-1} \quad \forall t,$$

   where $T_{i,t-1}^{on}$ is the time from which the unit i is turned on at time $t-1$ and $T_{i,min}^{on}$ is the minimal time during which unit i has to stay turned on.
5. Minimum down time limit:

   $$T_{i,t-1}^{off} \geq T_{i,min}^{off} \times (1 - u_{i,t-1})u_{i,t} \quad \forall t,$$

   where $T_{i,t-1}^{off}$ is the time from which the unit i is turned off at time $t-1$ and $T_{i,min}^{off}$ is the minimal time during which unit i has to stay off.

The feasible outcome vectors of the *objective space* are compared using the *Pareto dominance* $\succ$. In this minimization context, a solution $x \in \Omega$ is said to dominate a solution $y \in \Omega$, denoted by $x \succ y$, if they satisfy relation (1).

$$\forall i \in \{1,2\},\ f_i(x) \leq f_i(y) \bigwedge \exists i \in \{1,2\},\ f_i(x) < f_i(y) \qquad (1)$$

3 Solving methods

Many evolutionary algorithms involving a decoding system have been proposed to solve the single objective version of the UCP [3, 7, 8, 10, 16]. A binary vector u of size $T \times N$ is used to represent the solutions. In this representation each $u_{i,t}$ gives the state of a unit i (on or off) at a given time period t of the scheduling. Then the exact production of each unit is decoded by solving the dispatching sub-problem ($\mathscr{D}(u)$) using the lambda-iteration method [14]:

$$\min_p \sum_{t=1}^{T} \sum_{\substack{i \\ \text{s.t } u_i=1}} f_1((p_{i,t})_i)$$

such that:

$$\sum_{\substack{i \\ \text{s.t } u_i=1}} p_{i,t} = D_t \tag{2}$$

$$p_{i,min} \leq p_{i,t} \leq p_{i,max} \quad \forall i \text{ s.t } u_i = 1 \tag{3}$$

The advantage of the binary vector representation over a real vector representation where the productions are directly given is obvious: the search space is considerably reduced. In the multi-objective case, the dispatching problem is also a multi objective one as f_1 and f_2 have to be minimized. But this subproblem has still very good properties that make it easy to solve. It is a convex and continuous bi-objective problem. As the objective functions are convex and the decision variables are defined in a convex set all the solutions are supported [5] . Then the Pareto front solution is convex and totally defined by the set:

$$\{f_{1,\lambda}, f_{2,\lambda} | \lambda \in [0, 1]\},$$

where $f_{1,\lambda} = f_1(p*_\lambda)$ and $f_{1,\lambda} = f_1(p^*_\lambda)$, with p^*_λ solution of the dispatching problem $\mathscr{D}(u, \lambda)$ defined as follows:

$$p^*_\lambda = arg(\min_p \sum_{t=1}^{T} \sum_{\substack{i \\ \text{s.t } u_i=1}} \lambda f_1((p_{i,t})_i)) + (1 - \lambda) f_2((p_{i,t})_i))$$

such that (2) and (3) are met.

It seems interesting to consider a two level based method in the multi-objective version which will be similar to those proposed for the single-objective UCP. The method proposed is a GA, NSGA-II, handling binary vectors representing the solutions of commitment problem. Then the production of the units are obtained by using a decoding method. However, since the dispatching problem is a multi-objective one, there are many pareto equivalent solution to fix the production values. It becomes difficult to choose a method to associate phenotypic solutions with genotypic ones. Then three approaches of decoding are proposed and compared. In the first approach the solution associated with a binary representation is the one obtained by solving the dispatching problem for

λ fixed to 0.5. This approach is a naive one because it might miss some possibly good solutions. The second approach consists of adding λ to the representation of an individual. Then a binary vector associated with an on/off scheduling can be present many times in the population with different value of λ and all the solutions are reachable. In the last approach, an approximation of the Pareto front of the dispatching sub-problem is associated with each individual. Therefore an adapted version of NSGA-II is proposed to manage the association of many phenotypic solutions to a single genotypic solution.

In the next subsection, the common components of the three methods, which are essentially evolutionary operators, are presented. Then each method are explained in details.

3.1 Common components

Each multi-objective GA proposed hereafter are based on NSGA-II [4]. In each case the following operators are used:

Crossovers: Two crossovers are used. The first one is the classical one-point crossover and the second one is an intelligent two-points crossover. It is to randomly choose a window size and if the window size is smaller than the remaining portion of the vector, a new individual is created with the window portion of the worst parent and the remaining portion from the best parent. The reverse is done if the window size is larger than the number of bits in the remaining vector.

Mutations: Two mutations are used. The first one is the standard 1-bit-flip-mutation. This operator randomly flips a bit of the vector with a low probability. The second one randomly chooses a window whose size is randomly chosen and flips all the bits of this window.

Repair operator: The aim of this operator is to correct a solution if it does not meet the constraints of demand or of minimum on/off time. Firstly, it corrects the violations of time constraint hour by hour in modifying the states of unities if necessary. Then corrections are done on the constraints of power balance. This is done hour-by-hour. If the maximal capacity of production of the turned on units is lower than the power balance then a unit is turned on. The unit to turn on is chosen randomly among the units that can be turned on depending on the past hour (i.e. if this unit was turned on the previous hour or if it was turned off for a time long enough). The reverse process is done if minimal capacity of the turned on units exceeds the load demand. Naturally this correction process does not guarantee obtaining a feasible solution but this operator speeds up the algorithm and hence increases the possibility of finding such solutions.

Objective function: The objective values are computed by using the exact production $p_{i,t}$ at each time period t and for each unit i which is a phenotypic solution. The way to obtain phenotypic solutions from genotypic ones is different for each approach. The objective functions correspond to f_1, the cost production and f_2, the quantity of SO_2 and CO_2 emission. Nevertheless some penalties have to be added on the violation of time and load constraints. Therefore the objectives are:

$$\begin{cases} obj_1 = f_1 + c_p \sum\limits_{t=1}^{T} (\sum\limits_{i=1}^{N} p_{i,t} - D_t) + c_p (T_{i,off}^{t-1}(u_{i,t} - x_{i,t-1}) + T_{i,on}^{t-1}(u_{i,t-1} - u_{i,t})) \\ obj_2 = f_2 + c_p \sum\limits_{t=1}^{T} (\sum\limits_{i=1}^{N} p_{i,t} - D_t) + c_p (T_{i,off}^{t-1}(u_{i,t} - u_{i,t-1}) + T_{i,on}^{t-1}(u_{i,t-1} - u_{i,t})), \end{cases} \tag{4}$$

where c_p is a constant positive number.

3.2 Naive approach

In this approach a genotypic solution is decoded in solving a dispatching problem that is reduced to a single objective one by scalarization. Then it is possible that some pareto optimal phenotypic solutions cannot be reached because they are not solution of the chosen scalarized sub-problem. The decoding process is explained in detail hereafter.

Decoding process: The phenotypic solution $(p_{i,t})_{i,t}$ associated to a genotypic is computed by solving the dispatching problem $\mathscr{D}(u, 0.5)$ thank to the λ-iteration method.

3.3 Scalarized decoding

In this approach an integer ($\lambda_u \in \llbracket 0, 100 \rrbracket$) is added to the genotypic representation of a solution. This value is chosen randomly during the initialization process and then can be modified by the evolutionary operators. It is used to define the coefficients of scalarization during the decoding process. Hence, the main difficulty of the previous approach becomes possible to overcome, which is the inaccessibility of some pareto optimal solutions. Actually as the pareto front solution of the dispatching problem will always be convex all the pareto optimal solutions are reatchable by the scalarization method.

Representation: A value λ_u is added at the end of the representation vector with a binary encoding. λ_u is an integer between 0 and 100. The representation is shown Fig. 1.

Decoding process: The phenotypic solution $(p_{i,t})_{i,t}$ associated to a genotypic is computed in solving the dispatching problem $\mathscr{D}(u, \frac{\lambda_u}{100})$ thank to the λ-iteration method.

$$\overbrace{\underbrace{010\ \cdots\ 1}_{\text{unit 1}}\ \underbrace{010\ \cdots\ 0}_{\text{unit 2}}\ \cdots\ \underbrace{110\ \cdots\ 0}_{\text{unit N}}}^{\text{u}}\ \underbrace{101\cdots1}_{\lambda_u}$$

Fig. 1. Representation genotypic of a solution: the scalar λ is included in the representation.

Adaptation on the crossover: The one point crossover is transformed in a two-points crossover with the first crossover point in a locus of the definition of u and the second point in a locus of definition of λ_u. It is done in order that the crossover can have a significant impact on λ_u.

Adaptation on the mutation: The 1-bit-flip-mutation is applied only on the bits corresponding to u. λ_u is mutated by being replaced by a value chosen randomly between 0 and 100 with a normal distribution centered on its original value.

3.4 Multi decoding embedded approach

In this approach a genotypic solution is associated with a set of phenotypic solutions. This set of solutions is from the optimal pareto front solution of the dispatching problem associated with the genotypic solution. Fig. 2 helps to understand how genotypic solutions are represented in the objective space. The

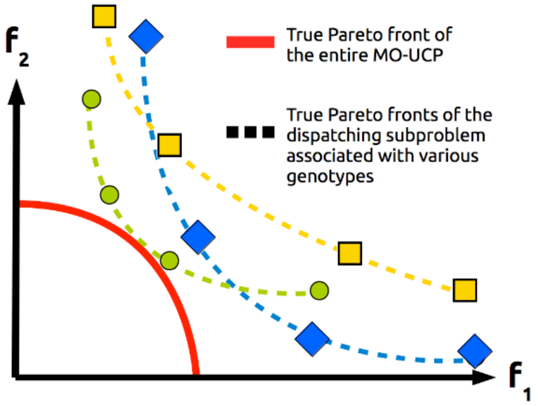

Fig. 2. Repressentation of the genotypic solutions in the objective space.

continuous line is the optimal pareto front of the entire MO-UCP. This front can be non-convex. Each one of the convex pareto fronts composed of the round, square or diamond points is derived from a single genotypic solution. They are composed of the phenotypic solutions found in solving the dispatching problem defined by the corresponding genotypic solution. The phenotypic solutions attached to a genotypic solution are pareto equivalent and form a convex front.

Decoding process: In this case a pareto set P_u will be associated with a genotypic solution u. This set is:

$$P_u = \{p_u^{\frac{k}{n_\lambda}}, k = 0...n_\lambda\}, \tag{5}$$

where n_λ is a fixed integer and $p_u^{\frac{k}{n_\lambda}}$ the optimal solution of $\mathscr{D}(u, \frac{k}{n_\lambda})$.

$$\overbrace{\underbrace{010\ \cdots\ 1}_{\text{unit 1}}\ \underbrace{010\ \cdots\ 0}_{\text{unit 2}}\ \ \cdots\ \ \underbrace{110\ \cdots\ 0}_{\text{unit N}}}^{\text{u}}$$

Fig. 3. representation genotypic of a solution u

$$\Downarrow$$

$$p_u^0 : \underbrace{0p^0_{1,2}0\cdots p^0_{1,T}}_{\text{unit 1}}\underbrace{0p^0_{2,2}0\cdots0}_{\text{unit 2}}\cdots\ \underbrace{p^0_{N,1}p^0_{1,2}0\cdots0}_{\text{unit N}}$$

$$p_u^{\frac{1}{3}} : \underbrace{0p^{\frac{1}{3}}_{1,2}0\cdots p^{\frac{1}{3}}_{1,T}}_{\text{unit 1}}\underbrace{0p^{\frac{1}{3}}_{2,2}0\cdots0}_{\text{unit 2}}\cdots\ \underbrace{p^{\frac{1}{3}}_{N,1}p^{\frac{1}{3}}_{1,2}0\cdots0}_{\text{unit N}}$$

$$p_u^{\frac{2}{3}} : \underbrace{0p^{\frac{2}{3}}_{1,2}0\cdots p^{\frac{2}{3}}_{1,T}}_{\text{unit 1}}\underbrace{0p^{\frac{2}{3}}_{2,2}0\cdots0}_{\text{unit 2}}\cdots\ \underbrace{p^{\frac{2}{3}}_{N,1}p^{\frac{2}{3}}_{1,2}0\cdots0}_{\text{unit N}}$$

$$p_u^1 : \underbrace{0p^1_{1,2}0\cdots p^1_{1,T}}_{\text{unit 1}}\underbrace{0p^1_{2,2}0\cdots0}_{\text{unit 2}}\cdots\ \underbrace{p^1_{N,1}p^1_{1,2}0\cdots0}_{\text{unit N}}$$

Fig. 4. representation phenotypic of a solution u for $n_\lambda = 3$

As many phenotypic solutions are attached to a single genotypic solution, the fitness assignment and diversity assignment methods of NSGA-II have to be adapted. This will be explained in detail in the following. The decoding process is represented in Fig. 3 and Fig. 4.

Adaptation of the fitness assignment process: The fitness value assigned to a solution u is the best fitness value among the fitness values of the phenotypic solutions $p_u \in P_u$:

$$fit(u) = \underset{p_u \in P_u}{\text{opt}} \left(fit(p_u) \right) \tag{6}$$

In NSGA-II the fitness is the rank of the solution, then *opt* is the minimization operator. This process ensures that the genotypic solution from which a pareto optimal solution can be generated is not be discarded.

Adaptation of the diversity assignment process: Let F_i be the set of the phenotypic solutions of rank i. In NSGA-II, the diversity measure used is the crowding distance between a solution x and the set of the other solutions having the same fitness, i.e. the same rank: $d_c(F_{fit(x)} - \{x\}, x)$. The adapted diversity assignment is the maximal diversity measurement among the ones of the individuals $p_u \in P_u \cap F_{fit(u)}$ computed without considering the elements of P_u:

$$d_c(F_{fit(u)}, u) = \max_{p_u \in P_u \cap F_{fit(u)}} d_c(F_{fit(u)} - \{P_u\}, p_u) \tag{7}$$

4 Experiments and Discussion

The aim of this section is to compare the three proposed methods.

4.1 Experimental Protocol

Instances Experiments will be realised on instances of 10, 40 and 100-unit data that are generated by duplicating the unit characteristics of the ten-unit system and the demand given in Tables 1 and 2. The load demands are adjusted in proportion to the size system. In all cases it is supposed that the reserve is 10% of the demand.

Table 2. Demand data with 24h time horizon.

hour	1	2	3	4	5	6	7	8	9	10	11	12
demand (MW)	700	750	850	950	1000	1100	1150	1200	1300	1400	1450	1500

hour	13	14	15	16	17	18	19	20	21	22	23	24
demand (MW)	1400	1300	1200	1050	1000	1100	1200	1400	1300	1100	900	800

Performance Assessment The different methods of performance assessment that can be chosen to compare multi-objectives algorithms are explained in details in [9]. In our case the ε-indicator and the hypervolume difference indicator are selected as they are complementary. Let Z^{all} be the set of objective vectors from all the Pareto set approximations we obtained during all our experiments. Then, a reference set R contains the non-dominated points of Z^{all}.

Table 1. Generating unit data for the ten-unit base system.

unit	unit1	unit2	unit3	unit4	unit5	unit6	unit7	unit8	unit9	unit10
P_{max}(MW)	455	455	130	130	162	80	85	55	55	55
P_{min}(MW)	150	150	20	20	25	20	25	10	10	10
a_1	1000	970	700	680	450	370	480	660	665	670
a_2	16.19	17.26	16.6	16.5	19.7	22.26	27.74	25.92	27.27	27.79
a_3 $(\times 10^{-5})$	48	31	200	211	398	712	79	413	222	173
b_1	712	570	700	860	350	370	480	660	665	670
b_2	12.9	10.26	10.60	15.50	7.70	9.26	3.74	5.92	7.27	7.79
$b_3(\times 10^{-4})$	4	3	22	11	10	22	30	40	13	23
T_{min}^{up}	8	8	5	5	6	3	3	1	1	1
T_{min}^{down}	8	8	5	5	6	3	3	1	1	1
CS_{hot}	4500	50000	550	560	900	170	260	30	30	30
CS_{cold}	9000	10000	1100	1120	1800	340	520	60	60	60
T_{CS}	5	5	4	4	4	2	2	0	0	0
initial status (h)	8	8	-5	-5	-6	-3	-3	-1	-1	-1

ε-indicator $I^1_{\varepsilon+}$ The unary version of this indicator is computed using the binary version given by (8) and the reference set R, with $I^1_{\varepsilon+}(A) = I_{\varepsilon+}(A, R)$.

$$I_{\varepsilon+}(A, B) = \inf_{\varepsilon \in \mathbb{R}} \{\forall z^1 \in B, \exists z^2 \in A, \forall i \in 1 \ldots n, z^1_i \leq \varepsilon + z^2_i\} \qquad (8)$$

Hypervolume difference indicator I^-_H The hypervolume indicator I_H is computed by the measure of the hypervolume between a set of solutions and the point $z = (z_1, \ldots, z_n)$ where z_k is the upper bound of the k^{th} objective regarding all the solutions of Z^{all}. The hypervolume difference indicator I^-_H is then computed with $I^-_H(A) = I_H(R) - I_H(A)$.

Experimental design All the implementations are realized under the ParadisEO 2.0 [11] software framework. A sensitivity analysis is carried out for each algorithm to determine the effect of the crossover rate and of the mutation rate. It is done thanks to the R statistical package *Irace* [12]. The population size is fixed to 100 individuals. A convergence criteria of 100 generations without improvement of the hypervolume is used as stopping criteria. For the multi decoding embedded approach the parameter n_λ is fixed to 10. For each case 20 runs are launched for each decoding system using the same seeds. Most of the performance assessment procedures are next achieved using PISA [1] platform and its performance assessment module. The existence of a significant difference between the result obtained by the different decoding systems is verified with the *Friedman* statistical test. Then a post-hoc test is carried out to compare the decoders by pairs. A p-value lower than 0.005 is used as a criterion for rejecting the null hypothesis.

Decoder	Naive	Scalarized	Multi
Naive	-	$=$	$<$
Scalarized	$=$	-	$<$
Multi	$>$	$>$	-

Fig. 5. Results optained from statistical comparisons with the ε-indicator and the hypervolume indicator.

Indicator	$I^1_{\varepsilon+}$		I_H^-	
Decoder	best	mean	best	mean
Naive	0.736	0.738	0.451	0.455
Scalarized	0.719	0.738	0.433	0.454
Multi	**0.709**	**0.712**	**0.422**	**0.425**

Fig. 6. Best value obtained for each indicator and decoder over the 20 runs.

4.2 Experimental Results

Results for the 10-units case: In Table 5 the results of the statistical tests of comparisons are summarized. The results do not differ from one indicator to the other one. In this table and on all the following the column "Naive" indicates results of the first approach, the column "Scalarized" those of the second one and the column "multi" those of the multi decoding embedded approach. From this table we can see that for the 10-unit based case the difference between the naive approach and the scalarized approach is statistically significant. However the multi decoding embedded approach gives results significantly better than those obtained by the other approaches. Table 6 gives the best and average values obtained for each indicator and decoder over the 20 runs. The decoding embedded approach improves the ε-indicator value of 1.36% in comparison to the other method. It also improves the hypervolume indicator of 2.54%.

Results for the 40-units case: In Table 7 the results of the statistical tests of comparisons are summarized. It can be observed that the multi decoding embedded approach gives significantly better results than the one obtained with the two other approaches. The scalarized decoding approach is better than the naive one. In Tab. 8 the best and average values obtained for each indicator and decoder over the 20 runs are shown. It can be seen that that for the ε-indicator the scalarizing approach improves the results of more than 7% in comparison with the first naive approach. Then the multi decoding embedded approach improves the result of the scalarizing approach by more than 99%. For the hypervolume indicator there is an improvement of 31% when adding the λ value in the representation. Then the multi decoding embedded approach improves the result of the scalarizing approach by more than 99.5%.

Decoder	Naive	Scalarized	Multi
Naive	-	$<$	$<$
Scalirized	$>$	-	$<$
Multi	$>$	$>$	-

Fig. 7. Results obtained from statistical comparisons with the ε-indicator and the hypervolume indicator.

Indicator	$I^1_{\varepsilon+}$		I^-_H	
Decoder	best	mean	best	mean
Naive	0.195	0.333	0.346	0.508
Scalarized	0.181	0.233	0.208	0.377
Multi	**0.00129**	**0.0840**	**0.000880**	**0.100**

Fig. 8. Best value obtained for each indicator and decoder over the 20 runs.

Decoder	Naive	Scalarized	Multi
Naive	-	$<$	$<$
Scalarized	$>$	-	$<$
Multi	$>$	$>$	-

Fig. 9. Results obtained from statistical comparisons with the ε-indicator and the hypervolume indicator.

Results for the 100-units case: In Table 9 the results of the statistical tests of comparisons are summarized. Again, the multi decoding embedded approach gives results significantly better than the ones obtained with the two other approaches. The scalarized decoding approach is better than the naive one.

Indicator	$I^1_{\varepsilon+}$		I^-_H	
Decoder	best	mean	best	mean
Naive	0.306	0.573	0.549	0.864
Scalarized	0.016	0.404	0.0169	0.636
Multi	**0.00389**	**0.150**	**0.000177**	**0.264**

Fig. 10. Best value obtained for each indicator and decoder over the 20 runs.

In Table 10 the best and average values obtained for each indicator and decoder over the 20 runs are shown. In comparison with the naive approach, the scalarazing approach improves the results by 94.7% for the ε-indicator and by 97% for the hypervolume. Then the multi decoding embedded approach improves the result of the scalarizing approach by 75.69% for the ε-indicator and by 99% for the hypervolume-indicator .

This results and the statistical tests lead to conclude that the choice of the decoder system has a significant impact on the result. Results obtained with the last decoder are drastically better than the ones obtained with the other decoders.

5 Conclusion

In this article a binary genetic algorithm has been proposed to solve a multi -objectives UCP. The main difficulty is that for one genotypic solution many phenotypic solutions could be attached. These phenotypic solutions are those of the pareto front solution of the dispatching problem. Three original decoding systems associating phenotypic solutions with the genotypic ones have been presented and compared.

The multi decoding embedded system is the main contribution of this paper. The efficiency of this method has been shown on three data set of different size. In each case the results obtained are significantly better than those obtained by the two other strategies. The bigger the data, the better the improvement. This decoding system is then the one selected. In a future work, the objective will be to compare the proposed binary GA using this decoder to the GA proposed in [15] and to a more classical multi-objective GA using a real vector to encode the solutions. First of all, it will be interesting to make an analyse study of the impact of the choice of the n_λ parameter. An important advantage of this method is that it could be reused to any multi-objective problem that can be written as follow:

$$\begin{cases} \underset{x,y}{\text{opt}}\, f_1(x,y) \\ \underset{x,y}{\text{opt}}\, f_2(x,y) \\ \ldots \\ \underset{x,y}{\text{opt}}\, f_n(x,y) \end{cases} \tag{9}$$

such that:

$$x \in X \tag{10}$$

$$y \in Y(x) \tag{11}$$

And such that for a fixed x the sub-problem $\mathscr{P}(x)$ finding the optimal pareto front of solutions with $y \in Y(x)$ is easy to solve. In this case the genetic algorithm will handle the x variables and the phenotypic solutions are found in solving $\mathscr{P}(x)$. The methodology chosen to solve $\mathscr{P}(x)$ does not matter. This is another advantage of this decoding system. Then, to test the multi decoding embedded approach on some other problem is one of our perspectives. We also believe that the multi decoding embedded approach can be generalized to any multi-objectives genetic algorithm. Then we plan to develop a generalized version of this approach that is suitable to any genetic algotrithm.

References

1. Bleuler, S., Laumanns, M., Thiele, L., Zitzler, E.: Pisa: A platform and programming language independent interface for search algorithms. In: EMO. pp. 494–508 (2003)
2. Catalao, J., Mariano, S., Mendes, V., Ferreira, L.: A practical approach for profit-based unit commitment with emission limitations. International journal of electrical power & energy systems 32(3), 218–224 (2010)
3. Damousis, I.G., Bakirtzis, A.G., Dokopoulos, P.S.: A solution to the unit-commitment problem using integer-coded genetic algorithm. Power Systems, IEEE Transactions on 19(2), 1165–1172 (2004)
4. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: Nsga-II. Evolutionary Computation, IEEE Transactions on 6(2), 182–197 (2002)
5. Ehrgott, M.: Multicriteria optimization, vol. 2. Springer (2005)
6. Guan, X., Zhai, Q., Papalexopoulos, A.: Optimization based methods for unit commitment: Lagrangian relaxation versus general mixed integer programming. In: Power Engineering Society General Meeting, 2003, IEEE. vol. 2. IEEE (2003)
7. Jeong, Y.W., Park, J.B., Shin, J.R., Lee, K.Y.: A thermal unit commitment approach using an improved quantum evolutionary algorithm. Electric Power Components and Systems 37(7), 770–786 (2009)
8. Kazarlis, S.A., Bakirtzis, A., Petridis, V.: A genetic algorithm solution to the unit commitment problem. Power Systems, IEEE Transactions on 11(1), 83–92 (1996)
9. Knowles, J., Thiele, L., Zitzler, E.: A Tutorial on the Performance Assessment of Stochastic Multiobjective Optimizers. TIK Report 214, Computer Engineering and Networks Laboratory (TIK), ETH Zurich (Feb 2006)
10. Lau, T., Chung, C., Wong, K., Chung, T., Ho, S.: Quantum-inspired evolutionary algorithm approach for unit commitment. Power Systems, IEEE Transactions on 24(3), 1503–1512 (2009)
11. Liefooghe, A., Jourdan, L., Talbi, E.G.: A software framework based on a conceptual unified model for evolutionary multiobjective optimization: Paradiseo-moeo. European Journal of Operational Research 209(2), 104–112 (2011)
12. López-Ibáñez, M., Dubois-Lacoste, J., Stützle, T., Birattari, M.: The irace package, iterated race for automatic algorithm configuration. Tech. rep., IRIDIA (2011)
13. de Moura Gomes Viana, A.M.M.: Metaheuristics for the Unit Commitment Problem The Constraint Oriented Neighbourhoods Search Strategy. Ph.D. thesis, Faculty of Engineering, University of Porto (2004)
14. Saramourtsis, A., Damousis, J., Bakirtzis, A., Dokopoulos, P.: Genetic algorithm solution to the economic dispatch problem—application to the electrical power grid of crete island. In: Proc. Workshop Machine Learning Applications to Power Systems (ACAI). pp. 308–317 (2001)
15. Srinivasan, D., Tettamanzi, A.G.: An evolutionary algorithm for evaluation of emission compliance options in view of the clean air act amendments. Power Systems, IEEE Transactions on 12(1), 336–341 (1997)
16. Swarup, K., Yamashiro, S.: Unit commitment solution methodology using genetic algorithm. Power Systems, IEEE Transactions on 17(1), 87–91 (2002)
17. Zhang, X.h., Zhao, J.q., Chen, X.y.: Multi-objective unit commitment fuzzy modeling and optimization for energy-saving and emission reduction [j]. Proceedings of the CSEE 22, 71–76 (2010)