International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:2, No:12, 2008

Multi-Case Multi-Objective Simulated Annealing (MC-MOSA): New Approach to Adapt Simulated Annealing to Multi-objective Optimization

Abdelfatteh Haidine and Ralf Lehnert

***Abstract*—**In this paper a new approach is proposed for the adaptation of the simulated annealing search in the field of the Multi-Objective Optimization (MOO). This new approach is called Multi-Case Multi-Objective Simulated Annealing (MC-MOSA). It uses some basics of a well-known recent Multi-Objective Simulated Annealing proposed by Ulungu *et al.*, which is referred in the literature as U-MOSA. However, some drawbacks of this algorithm have been found, and are substituted by other ones, especially in the acceptance decision criterion. The MC-MOSA has shown better performance than the U-MOSA in the numerical experiments. This performance is further improved by some other subvariants of the MC-MOSA, such as Fast-annealing MC-MOSA, Re-annealing MC-MOSA and the Two-Stage annealing MC-MOSA.

***Keywords*—**Simulated annealing, multi-objective optimization, acceptance decision criteria, re-annealing, two-stage annealing.

I. Introduction

THe Multi-Objective Optimizations (MOO) is the process of finding one or more vectors of decision variables that simultaneously satisfy all feasibility constraints and optimize a vector objective function that maps the decision variables to two or more performance criteria or objectives. Thus, in a Multi-objective Optimization Problem (MOP), we have a set of n parameters or variables (decision variables), a set of k objective functions, and a set of m constraints. Objective functions and constraints are functions of the decision variables. We assume that all the objectives have to be maximized, so that the MOP may be expressed as:

$$maximize \qquad \mathbf{z} = \mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_k(\mathbf{x})] \quad (1)$$

$$subject \quad to: \quad \mathbf{e}(\mathbf{x}) = [e_1(\mathbf{x}), e_2(\mathbf{x}), \ldots, e_m(\mathbf{x})] \leqslant 0 \quad (2)$$

$$where: \qquad \mathbf{x} = [x_1, x_2, \ldots, x_n] \in \mathbf{\Omega} \quad (3)$$

$$\mathbf{z} = [z_1, z_2, \ldots, z_k] \in \mathbf{Z} \quad (4)$$

The *decision vector* $\mathbf{x} = [x_1, x_2, \ldots, x_n]$ (also called solution) is a vector of decision variables (also parameters), representing the numerical qualities for which values must be found in an optimization problem. The variables may be integer, real, or a mixture. The set of all decision vectors for a given MOP is called the decision space (also parameter space) $\mathbf{\Omega}$. The sub-set of the decision space, where all decision vectors satisfy the vector of constraints $\mathbf{e}(\mathbf{x})$, is called "feasible set" $\mathbf{\Omega}_f$, that is described as:

Abdelfatteh Haidine and Ralf Lehnert are with the Dresden University of Technology, Dresden, Germany, email: haidine@ifn.et.tu-dresden.de

$$\mathbf{\Omega}_f = \{\mathbf{x} \in \mathbf{\Omega} \mid \mathbf{e}(\mathbf{x}) \leqslant 0\}$$

The vector objective function $\mathbf{f}(\mathbf{x})$ maps the decision vectors from the decision space into a *k*-dimensional objective space (also criterion space) $\mathbf{Z} \subset \mathbb{R}_k$, where each vector $\mathbf{z} \in \mathbf{Z}$ is denoted as *objective vector*, or *criterion vector*. The optimization definition given in this section has a maximization form. Generally, we may have to minimize all the objective functions, maximize them all, or minimize some functions and maximize others. However, any objective function can always be converted from the minimization form to the maximization form, and vice versa since:

$$max(f_i(\mathbf{x})) = -min(-f_i(\mathbf{x})) \qquad \text{and}$$
$$min(f_i(\mathbf{x})) = -max(-f_i(\mathbf{x}))$$

In the Single-Objective Optimization (SOO), the feasible set is completely ordered according to the objective function $f(.)$: for two feasible solutions $\mathbf{a}$ and $\mathbf{b}$, either $f(\mathbf{a}) \leq f(\mathbf{b})$ or $f(\mathbf{a}) \geq f(\mathbf{b})$. The goal of the optimization is to find the solution that gives the maximum of $f(.)$. In contrary to SOO, in the MOO the $\mathbf{\Omega}_f$ is not totally ordered, but only partially ordered. In the objective space, two objective vectors $\mathbf{u}$ and $\mathbf{v}$ can have two possible relations. In one case, one solution is dominating the other, e.g. $\mathbf{u}$ is dominating $\mathbf{v}$, if $f_i(\mathbf{u}) > f_i(\mathbf{v})\ \forall i$. In the second case, they are indifferent to each other, if $\exists i$ such that $f_i(\mathbf{u}) > f_i(\mathbf{v})$ and $\exists j$ for which $f_j(\mathbf{u}) < f_j(\mathbf{v})$. In the latter case, solutions $\mathbf{u}$ and $\mathbf{v}$ are called to be non-dominated. An algorithm that solves a MOP finds out the set of the non-dominated solutions found during the search process. This is called *Pareto front* if an exact algorithm is used, and *approximation set* if a metaheuristic is used. A simple example of an approximation set built by the solution $\widehat{S}_i$ and its similar solutions is given in Figure 1 for a two-dimensional objective space. Each solution from this set dominates at least one solution S_i, as shown in the figure. More details about characteristics and definitions of Pareto fronts (i.e. approximation sets) can be found in [1] [2].

For solving the MOPs, different multi-objective algorithms (MOAs) have been developed over the last decade. Generally, these MOAs can be categorized in some well-known classes, such as Multi-Objective Local Search (MOLS), the Multi-Objective Evolutionary Algorithms (MOEA), etc. One of the most widely used MOLS variant is that based on the simulated

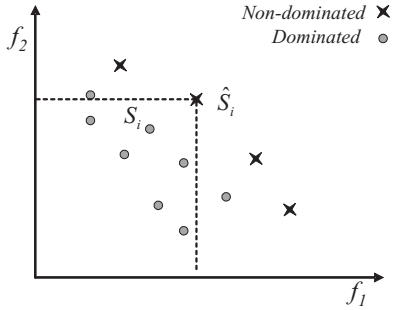

Fig. 1. Example of approximation set in a two-dimensional objective space

annealing approach, which is inherited from the SOO. This approach has been implemented in the MOO according to different schemes, as it is detailed in the next section. These schemes differ in the way they implement the decision criteria of acceptance/rejection of new bad solutions. Those schemes could be implemented in another possible way, in order to exploit the search space better. In this paper, we propose a new scheme that is called *Multi-Case Multi-Objective Simulated Annealing* (MC-MOSA). Extensions of this basic MC-MOSA form are also proposed and evaluated.

The rest of the paper is organized as follows: in the second section an overview on the MOA landscape is given, where also the usual method for the adaptation of the simulated annealing into the multi-objective optimization is described. The third section discusses some drawbacks of the standard MOSA and how they could be avoided by the means of a new proposed approach called the multi-case MOSA. Further extensions of the MC-MOSA that should improve its performance are described in the fourth section. The proposed MOSA variants are evaluated by simulations, which numerical results are presented in the fifth section. Their performance is evaluated and compared with those of the original MOSA version. Conclusions gained from the results are given in the last section.

II. Metaheuristics for Solving the MOPs

A. Overview on MOA Landscape

Similar to the single-objective optimization, there is a wide spectrum of metaheuristics that are proposed in the literature for solving the MOPs. The landscape of the multi-objective metaheuristics (MOMH) has the same structure to that one seen in the case of SOO. Thus, the algorithms could be classified into three different major classes. The class of Multi-Objective Local Search (MOLS) algorithms, which includes the Serafini's Multi-Objective Simulated Annealing (S-MOSA) [3], the Ulungu et al. MOSA (U-MOSA) [4], the Pareto Simulated Annealing (PSA) [5], the Multi-Objective Multi-Start Local Search (MOMSLS) [6], the Multi-Objective Tabu Search (MOTS) and its variants [7] [8], etc. The second class is the Multi-Objective Evolutionary Algorithms (MOEA). This class covers a large number of variants, like Vector Evaluated Genetic Algorithm (VEGA) [9], the Multi-Objective Genetic Algorithm (MOGA) [10], the Niche Pareto Genetic Algorithm (NPGA) [11], Non-dominated Sorting Genetic Algorithm (NSGA) [12] and its second variant (NSGA-II) [13], Strength Pareto Evolutionary Algorithm (SPEA) [14] and its extended variant SPEA-2 [15] [16], Pareto-Archived Evolutionary Strategy (PAES) [2], etc. Alike to the SOO field, the hybridation of MOLS and MOEA build the hybrid multi-objective algorithm (HMOA), called also Memetic Multi-Objective Algorithms (MMOA), whose principle elements are the Ishiburi's Multi-Objective Genetic Local Search (I-MOGLS) [17] and the Jaszkiewicz's MOGLS (J-MOGLS, it is called sometimes Random Directions MOGLS -RD-MOGLS) [18], the Pareto Ranking-based MOGLS algorithm that hybridize the Pareto MOEAs with a LS (PR-MOGLS) [6], the Memetic PAES (M-PAES) [2]. The SPEA and NSGA-II were hybridized with I-MOGLS to build the Hybrid SPEA and Hybrid NSGA-II; [19].

In this paper we focus only on the simulated annealing based MOAs. The mostly referenced and recent MOSA variant, namely the U-MOSA, is considered in this work. This choice has also been motivated by the relative better performance of this MOSA variant in comparison to other ones. An early comparison of the MOSA variants in [20] has stated that the Serafini's MOSA is the worst one, whilst the Ulungu-MOSA and PSA have better performance in solving the 0/1 multi-objective Knapsack problem. A recent performance evaluation in [21] has shown that the Ulungu MOSA is the best MOSA variant in solving the graph partitioning problem with load balancing in telecommunications networks. This is referred further in this work as Standard MOSA (S-MOSA). This is described in detail below. This discussion should allow to understand the functionality of the MOSA, and to find out some drawbacks of this technique. Those drawbacks will be avoided in a new proposed approach, which is called Multi-Case Multi-Objective Simulated Annealing (MC-MOSA).

B. Multi-Objective Simulated Annealing

The origins of Simulated Annealing (SA) lay in the analogy of optimization and a physical annealing process; [22]. In condensed matter physics, annealing is a thermal process for obtaining low-energy states of a solid in a heat bath. Roughly, the process can be described as follows. First, the temperature of the heat bath is increased to a maximum value at which the solid melts. Thus, all particles of the solid arrange themselves randomly. Afterwards, the temperature is carefully decreased until the particles of the melted solid reach in the ground state of the solid in which the particles are arranged in a highly structured lattice with minimum energy. The physical annealing process can be simulated by computer programs using Monte Carlo techniques proposed in [23]. Given an actual state S_{act} of the solid with energy E_{act}, a subsequent new state S_{new} is generated by applying a perturbation mechanism, which transforms the current state into the next state by a small distortion, for instance by displacement of a single particle. If the energy difference $\Delta E = E_{act} - E_{new}$ is less or equal to zero, the state S_{new} is accepted as the current state. If the energy difference is greater than zero, the state S_{new} is accepted with probability $exp(\Delta E/(\alpha.T))$, where T denotes

the temperature of the heat bath and α the Boltzmann constant. This acceptance rule is known as the Metropolis criterion. The flowchart of a simple SA in SOO is given by Algorithm 1.

Algorithm 1 Flowchart of SA in SOO

begin
$t := T(0)$, $n := 1$;
$S_{best} \leftarrow S_{act}$;

repeat
Generate neighboring solution $S_{new} \in \mathcal{N}(S_{act})$
$\Delta f := f(S_{act}) - f(S_{new})$
if($\Delta f \leqslant 0$)
$S_{act} \leftarrow S_{new}$
else
$S_{act} \leftarrow S_{new}$ with $p_{accept} = exp(-\Delta f/t)$
if($f(S_{act} > f(S_{best})$
$S_{best} \leftarrow S_{act}$
$t := T(n)$
$n := n + 1$
until termination criterion fulfilled
return S_{best}
end

Different schemes have been proposed in order to adapt the principles of the simulated annealing in the field of the MOO, as cited above. This results in different variants of the Multi-Objective Simulated Annealing (MOSA). The MOSA variant proposed by Ulungu et al. in [4] is the recent and mostly referenced one, because of that it is considered in this work and abbreviated by Standard MOSA (S-MOSA). The SA in MOO has the same structure as in SOO; however, the acceptance decision of new solution has to take into consideration the improvement (or deterioration) of k objectives simultaneously. In case of two optimization objectives (i.e. $k = 2$), the comparison of the actual and the new solution (i.e. decision vector) results in three cases, as depicted in Figure 2:

- *Case (a)*: The move from $\mathbf{x}_{act}$ to $\mathbf{x}_{new}$ is improving with respect to all k objectives. This means that $\Delta f_{k'} = f_{k'}(\mathbf{x}_{new}) - f_{k'}(\mathbf{x}_{act}) \leq 0$ (supposing a minimization MOP) for $k' \in \{1, ..., k\}$.
- *Case (b)*: An improvement and a deterioration can be simultaneously observed on different objectives. This means, there exist a k' and a k'' with $\Delta f_{k'} < 0$ and $\Delta f_{k''} > 0$. This is the case where the new solution is indifferent to the actual one. Thus, a strategy has to be defined to decide if the new solution should be accepted as current solution for the next iteration.
- *Case (c)*: All objectives are deteriorated, with $\Delta f_{k'} \geq 0$ for all k' and $\exists k'' \in \{1, ..., k\}$ such that $\Delta f_{k''} > 0$. In this case, an acceptance probability to accept $\mathbf{x}_{new}$ has to be calculated.

Different procedures have been discussed to calculate the acceptance probability in cases (b) and (c). The *criterion scalarizing approach* was found to be the best one. Its idea is to project the multidimensional criteria space $\Delta_{\mathbf{f}} = \{\Delta f_{k'} | k' = 1, ..., k\}$ into a monodimensional one (ΔS). On this monodimensional space, the classical acceptance decision rule can be applied, exactly like in SOO. To perform this projection, a scalarizing function $S(.) = S(\mathbf{f}(), \lambda)$ is used, which aggregates the multi-criteria information into a unicriterion one $S(.)$ using a weight vector λ. By taking $\Delta S = S(\mathbf{f}(\mathbf{x}_{new}), \lambda) - S(\mathbf{f}(\mathbf{x}_{act}), \lambda)$, the acceptance decision rule takes the following form for a minimization problem:

$$p_{accpet} = \begin{cases} 1 & \text{if } \Delta S \leq 0 \\ \exp(-\frac{\Delta S}{T}) & \text{if } \Delta S > 0 \end{cases} \quad (5)$$

Several scalarizing functions can be applied to compute this probability. The most known one is the *weighted sum*, beside the *Tchebychev norm*. The weighted sum has the following form:

$$S(\mathbf{f}(), \lambda) = \sum_{1 \leq k' \leq k} \lambda_{k'} f_{k'} \quad (6)$$

with

$$\sum_{1 \leq k' \leq k} \lambda_{k'} = 1 \quad \text{and} \quad \lambda_{k'} > 0, \forall k' \quad (7)$$

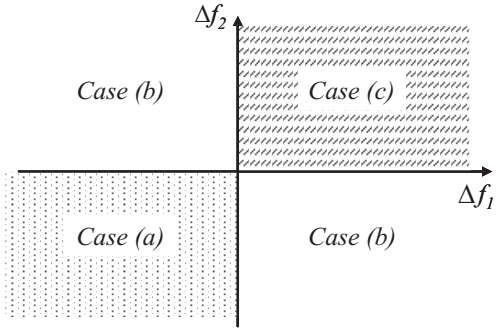

Fig. 2. Three possible cases in comparing two decision vectors in two-dimensional decision space

C. MOA Performance Metrics

Basically, there are two ways to assess the performance of a multi-objective algorithm, either theoretically by analysis or empirically by simulation. For the theoretical analysis, two approaches were proposed in the literature. On one hand, there is the "limit behavior" that tries to find what the algorithm is able to achieve over time, when unlimited time resources are available. On the other hand, the "run-time analysis" analyzes the expected running time for a given class of problems and the success (i.e. finding the Pareto set) probability for a given optimization time; [16]. These methods are applicable only on the standard and very simple MOO problems, such as the Leading Ones Trailing Zeros (LOTZ) function that is considered in [16], while their applicability to the real-life problems is neither checked nor justified. Because of that only the performance evaluation approaches that are based on the simulative investigations are considered in this paper.

The notion of performance covers both the quality of the outcome as well as the computational resources needed to generate this outcome. For the latter aspect, it is common practice to monitor either the number of the fitness evaluations (or the number of algorithm iterations) or the overall run time on a particular computer, exactly like in the SOO. However, concerning the outcome quality aspect, there is a difference. The evaluation and the comparison of two SOO algorithms is relatively simple, because a SOO algorithm produces at the end of its execution exactly one solution as optimal (or near-to-optimal) solution for the considered problem. The comparison of the fitness of the solutions delivered by different algorithms tells us which algorithm is the best one for the treated problem. In the MOO, the algorithm provides at the end not a unique solution but a set of solutions that realize the optimal trade-offs between the considered optimization objectives. This set is the approximation set. In case of a two-dimensional MOO, the optimal solutions build a Pareto front (or curve). In general, for a MOO model with k objectives, the near-Pareto solutions front takes the shape of a k-dimensional geometrical "surface" sitting above k-dimensional hyperplane spanned by the vector $\mathbf{S} = [f_1(\mathbf{S}), f_2(\mathbf{S}), \cdots, f_k(\mathbf{S})]$. Therefore, any evaluation of a MOO algorithm or the comparison of the performance of different algorithms has to be based on their supplied approximation sets. In the MOO literature devoted to the performance evaluation, a large number of methods is proposed. For example, the set cardinality, set coverage, spread and distribution are proposed in [1]; while *Epsilon* quality measure and other ones are in [2]. In this work, set cardinality and coverage of two sets are taken as metrics, which are defined as:

- **Approximation set cardinality**: This is the number of elements (i.e. solutions) constituting the front. It is the number of the optimal trade-offs between the optimization objectives found by the considered algorithm. It reflects how good is the algorithm in exploring the search space.
- **Coverage of two sets** ($\mathcal{C}$): let $\mathbb{A}, \mathbb{B} \subseteq \mathbf{\Omega}_f$ be two approximation sets built by set of decision vectors. The function $\mathcal{C}()$maps the ordered pair $(\mathbb{A}, \mathbb{B})$ to the interval [0,1] as follows:

$$\mathcal{C}(\mathbb{A}, \mathbb{B}) := \frac{|\{\mathbf{b} \in \mathbb{B} | \exists \mathbf{a} \in \mathbb{A} : \mathbf{a} \succcurlyeq \mathbf{b}\}|}{|\mathbb{B}|} \qquad (8)$$

  The value $\mathcal{C}(\mathbb{A}, \mathbb{B}) = 1$ means that all solutions (i.e. decision vectors) in $\mathbb{B}$ are weakly dominated by $\mathbb{A}$. If $\mathcal{C}(\mathbb{A}, \mathbb{B}) = 0$, then no one of the points of $\mathbb{B}$ is dominated by any point of $\mathbb{A}$. During the comparison always both directions have to be considered, since $\mathcal{C}(\mathbb{A}, \mathbb{B})$ is not necessary equal to $1 - \mathcal{C}(\mathbb{B}, \mathbb{A})$.

Generally, the set cardinality evaluates the algorithm performance in keeping a good diversity, which allow a good exploration of the search space. The coverage set reflects the algorithm's ability in the convergence toward the global optimum approximation, simply called Pareto front. If the performance of two different algorithms has to be compared, then their reached fronts are compared with each other. Comparison examples of two fronts $\mathbb{A}$ and $\mathbb{B}$ are given in Figure 3. In case 1, front $\mathbb{A}$ is better than front $\mathbb{B}$ in the convergence, but it has a very low cardinality. In case 2, the fronts (i.e. the algorithms) have a similar performance level in both metrics. The third case shows a clear and absolute advantage in the performance of the algorithm front $\mathbb{A}$. It has a large number of solutions, which clearly dominate those of front $\mathbb{B}$.

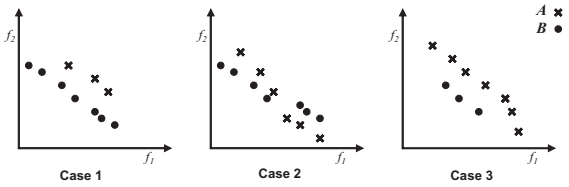

Fig. 3. Different states for comparison of two fronts

III. Multi-Case Multi-Objective Simulated Annealing (MC-MOSA)

The main negative point in the functions of the S-MOSA is its acceptance decision rule, which is based only on a direct comparison between the actual and the new solutions. The drawback of such comparison is that some new solution that are good for the search could be rejected, especially in cases (b) of Figure 2. For example, the case (b.1) in Figure 4 illustrates a subcases of case (b), where the $\mathbf{x}_{new}$ is not better than $\mathbf{x}_{act}$. In this subcase, it is probable that the new solution will be rejected. However, if we compare the new solution with the solution of the approximation sets, we remark that it dominates three of those solutions and indifferent to the rest of solutions. The new solution has reached a better region of the objective space, which has not yet been reached by any other solution. Because of that, this solution must not be rejected; instead it must be accepted, since it will lead the search to a better search region. The new multi-objective local search approach that is proposed in this section is called Pareto Multi-Case MOSA (simply MC-MOSA). The MC-MOSA proposes an acceptance decision rule that is based on the comparison of both solution with each other, and by taking also into the consideration the Pareto solution in case (b). This is done by subdividing the MOSA case (b) in three subcases (case (b.1), case (b.2) and case (b.3), which are shown in Figure 4. In the first subcase, the algorithm checks the number of the Pareto solutions that are dominated by the new solution ($R_{\succ}(\mathbf{x}_{new})$), which is called as *dominance counter* of the new solution. if $R_{\succ}(\mathbf{x}_{new}) > 0$, then $p_{accept} = 1$. If the new solution does not dominate any solution, then we check if it can be a new solution in the Pareto set. This is done by calculating the number of the solutions in the Pareto set that dominates this solution ($R_{\prec}(\mathbf{x}_{new})$), called also the *dominance rank*. If $R_{\prec}(\mathbf{x}_{new}) = 0$, then $p_{accept} = 1$, as illustrated by case(b.2). If the new solution can not build a solution of the Pareto set (i.e. $R_{\prec}(\mathbf{x}_{new}) > 0$), then the acceptance probability is based on the difference between $R_{\prec}(\mathbf{x}_{new})$ and $R_{\prec}(\mathbf{x}_{act})$. This difference should allow the search process to accept the solution that is not too far from the actual approximation

set and to avoid the regions that are dominated by a dense population of the Pareto solutions, as depicted in case (b.3). This subdivision to decide which probability should be used to compute acceptance probability of new solution builds the Pareto MC-MOSA, which flowchart is given by Algorithm 2.

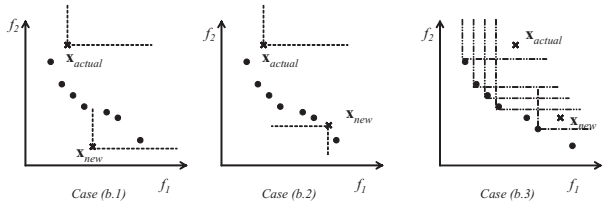

Fig. 4. Subdivision of the MOSA case (b) into different sub-cases

Algorithm 2 Flowchart of Multi-Case Multi-Objective Simulated Annealing (MC-MOSA)

begin
$t := T(0)$, $n := 1$;
$\mathbb{A}_n \leftarrow \mathbf{x}_{act}$;

repeat
Generate neighboring solution $\mathbf{x}_{new} \in \mathcal{N}(\mathbf{x}_{act})$
if($\mathbf{x}_{new} \succ \mathbf{x}_{act}$) *//Case (c)*
$\mathbf{x}_{act} \leftarrow \mathbf{x}_{new}$ *//*$p_{accept} = 1$
else if($\mathbf{x}_{new} \not\succ \mathbf{x}_{act}$) *//Case (b)*
if($R_{\succ}(\mathbf{x}_{new}) > 0$) *//Case (b.1)*
$p_{accept} = 1$
else if($R_{\prec}(\mathbf{x}_{new}) = 0$) *//Case (b.2)*
$p_{accept} = 1$
else if($R_{\prec}(\mathbf{x}_{new}) < R_{\prec}(\mathbf{x}_{act})$) *//Case (b.3)*
$p_{accept} = 1$
else if($R_{\prec}(\mathbf{x}_{new}) = R_{\prec}(\mathbf{x}_{act})$) *//Case (b.3)*

$$p_{accept} = \exp\left(-\frac{1-\left(\sum_{k'} \lambda_{k'} \frac{f_{k'}(\mathbf{x}_{act}) - f_{k'}(\mathbf{x}_{new})}{(f_{k'}(\mathbf{x}_{act}) + f_{k'}(\mathbf{x}_{new}))/2}\right)}{T}\right)$$

else *//*$(R_{\prec}(\mathbf{x}_{new}) > R_{\prec}(\mathbf{x}_{act}))$

$$p_{accept} = \exp\left(-\frac{\left(R_{\prec}(\mathbf{x}_{new})/R_{\prec}(\mathbf{x}_{act}) + 1\right)}{T}\right)$$

else//case (c)

$$p_{accept} = \exp\left(-\frac{1-\left(\sum_{k'} \lambda_{k'} \frac{f_{k'}(\mathbf{x}_{act}) - f_{k'}(\mathbf{x}_{new})}{(f_{k'}(\mathbf{x}_{act}) + f_{k'}(\mathbf{x}_{new}))/2}\right)}{T}\right)$$

update $\mathbb{A}_n$
$t := T(n)$
$n := n + 1$
until termination criterion fulfilled
return $\mathbb{A}_n$
end

Different methods could be used to extend standard form of the MC-MOSA, such as the utilization of some approaches that have been developed and tested for the application of the simulated annealing in the SOO. Most of those approaches have shown considerable improvements of the performance of the simulated annealing search process in SOO; however, they are still not investigated for the case of the MOO. Therefore, one of the goals of the MC-MOSA extensions in this work is to check if those approaches are also successful in improving the SA-based search in the MOO, as it was the case in SOO. Several SA extension approaches can be found in the literature; however, only some of them (the most successful) are considered.

IV. Extensions of the MC-MOSA

A. Re-annealing MC-MOSA (R-MC-MOSA)

This variant adapts the approach used in the SOO SA, which uses *Non-monotonic cooling schedules*; [24]. In this MOSA variant, phases of cooling and reheating (increase of temperature) are alternated. This approach should help the MOO search to escape the local optima, as it was realized in the SOO search. The reheating is a periodic increase of the temperature. This temperature increase results in increasing the acceptance probability of new solutions, which are worse than the actual one. The reheating is applied in an advanced phase of the search, where the search algorithm is nearing to the convergence. Because the early convergence of the local search-based algorithms tends toward local minima, the reheating should allow the search algorithm to escape them with higher probability and increase the chance to reach the global optimum (or at least to reach better local minimum). The reheating can have different forms. In this work, a linear reheating is considered. The temperature at the cooling iteration c of R-MOSA can be formulated as follows:

$$T_c^{(R)} = f_{cool}(T_{c-1}^{(R)}) + \alpha_{Reh}^{(T)}.R_A \tag{9}$$

with

$$\alpha_{Reh}^{(T)} = \begin{cases} 1 & \text{if } \dfrac{c}{R_{period}^{(T)}} \in \mathbb{N} \\ 0 & \text{otherwise.} \end{cases} \tag{10}$$

where $f_{cool}(T_{c-1})$ is the cooling function that decreases the temperature resulting form the previous cooling (i.e. $c-1$), $\alpha_{Reh}^{(T)}$ is the reheating factor that indicates if a reheating with a *reheating amplitude* R_A should occur or not. This is a function of the reheating period $R_{period}^{(T)}$, which is expressed in number of cooling iterations. In Figure 5 the behavior of the temperature over the cooling iterations in R-MC-MOSA is qualitatively compared to that of simple MC-MOSA.

B. Fast Annealing MC-MOSA (F-MC-MOSA)

The standard MOSA uses, like standard SA in SOO, a cooling schedule that is a geometric function of the number of the cooling iterations (or steps). Such cooling is written as $T_c = \alpha.T_{c-1} = \alpha^c.T_0$, where $\alpha \in [0, 1[$ is a scaling

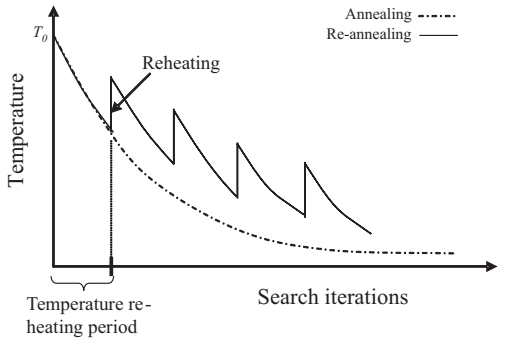

Fig. 5. Comparison of a cooling schedule with and without re-annealing

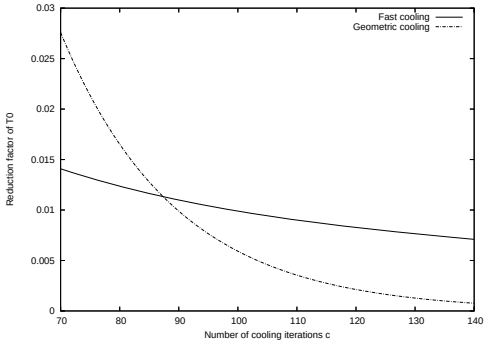

Fig. 7. Difference between geometric and fast cooling at high iteration numbers

constant and c is the cooling iteration counter. According to the literature, useful values for α have been claimed to be $0.8 < \alpha < 0.99$, and mostly this is taken $\alpha = 0.95$. A sub-variant of the SOO SA has been proposed and called Fast Simulated Annealing (FSA), which uses a faster cooling schedule. The fast cooling schedule decreases the temperature according to $T_c = T_0/(1 + c)$. This cooling behavior is compared to that of standard geometric cooling in Figure 6. In this section, the fast cooling is studied for the case of MOO, which builds the Fast MOSA (F-MOSA). The expected advantage that the fast cooling could show lies in the fact that the bad moves will be more probably rejected with increasing number of search iterations, in the contrary to the S-MOSA. This should realize a faster convergence to the optimum. But this could become inconvenient by leading the search to a local optimum because of the premature convergence. However, this drawback should be avoided because the temperature at the late iterations remain higher than in case of geometric cooling; as illustrated in Figure 7. The higher temperatures result in a higher acceptance probability of relatively bad solutions, which generally allows the algorithm to escape from local optima.

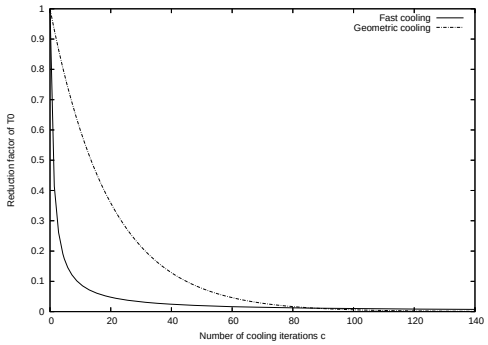

Fig. 6. General comparison of geometric and fast cooling schedules

C. Two-Stage Annealing MC-MOSA (TS-MC-MOSA)

This variant consists in using two different cooling schedules in the different phases of the search. Different mixtures of the cooling schemes are possible. However, in this work we propose to test only one variant, in order to check if generally the variation of the cooling process according to the search phase does bring any advantage to the search or not. Therefore, a mixed cooling scheme that is proposed, where a fast annealing is used at the higher temperature, while a slower one is used at lower temperatures. The geometric cooling of the S-MOSA is used at the early phase of the search, i.e. in the first half of the search process duration. The effect of the two-stage cooling resides in the fact that it realizes a slower cooling over all the process duration than the geometric and the fast annealing schemes. In fact, in the early iterations of the geometric cooling the temperature decrease slowly than the fast annealing, while at the last iterations the temperature decreases in fast annealing slower than with geometric one. The low decrease of the temperature in the later iterations means that the bad moves acceptance probability is higher. The acceptance of bad moves in this phases where the algorithm is in the convergence phase allows the search process to escape from the probable local optima, where the algorithm could be trapped.

V. Numerical Results and Evaluations

A. Numerical Experiments

The performance investigations of the proposed MOO algorithm variants are based on simulation, where the considered variants are applied to solve a well-known multi-objective combinatorial optimization problem that is called Multi-Objective Knapsack Problem (MOKP). After that, the performance of each variant will be compared with those of the standard form. The 0/1 Knapsack problem was first considered in the SOO; [25]. It consists of a set of items, weight and profit associated with each item, and an upper bound for the capacity of the Knapsack. The task is to find a subset of items which maximizes the total profits in the subset, yet all selected items fit into the Knapsack, which means the total weight does not exceed the given capacity. This SOO problem has been extended to a MOO problem by using a given number of Knapsacks; [1]. Formally, the 0/1 MOKP considered is defined as follows:

Given: A set of n items and a set of k knapsacks, with $p_{i,j}$: profit of item j according to knapsack i; $w_{i,j}$: weight of item j according to knapsack i, and C_i: capacity of knapsack i;

Task: Find a vector a vector $\mathbf{x} = (x_1, x_2, \cdots, x_n) \in \{0,1\}^n$;

Objective: *maximize* $\mathbf{f(x)} = (f_1(\mathbf{x}), f_2(\mathbf{x}), \cdots, f_k(\mathbf{x}))$, where $f_{k'} = \sum_{j=1}^{n} p_{i,j}.x_j$ and $x_j = 1$ if and only if item j is selected;

Constrained by: Capacity constraint of each knapsack k', with $e_{k'} = \sum_{j=1}^{n} w_{i,j}.x_j$ and $1 \leq k' \leq k$.

The MOKP is solved by the new variants and their performance is compared to those of the original standard MOSA version. The experiments are done with three different instances with different sizes and in a two-dimensional search space (i.e. with two knapsacks). In the first instance 250 items are used, while 500 and 750 items are used in the second and third instances; respectively. Firstly, the standard MOSA is compared with the simple variants of the Multi-Case MOSA. After that, the S-MOSA is compared with the extended MC-MOSA subvariants. The extensions implemented in this work have been realized inside the platform of MOO libraries called MOMHALib++; [26]. All the given computational results in this section are achieved by a PC with 1200MHz and 512MB, under Linux Suse 8.2. The presented statistical numerical results given in this section are the average (with confidence interval) of the values achieved by 30 runs for each experiment.

B. S-MOSA vs MC-MOSA

The application of MC-MOSA to solve the MOKP has shown improvements in the performance of the local search in MOO. The very important advantage that MC-MOSA brought is the size of the approximation sets, which is widely superior to that found by S-MOSA. The cardinality of the fronts of both algorithms is compared in Table I. This improvement is realized by the fact that the introduced subcases in MC-MOSA allows the detection of any small improvement that leads to finding new solution of that could lead to larger fronts. Concerning the MC-MOSA convergence, it is evaluated by reference to the S-MOSA by the means of the coverage of two sets, which statistics are given in Table II. The MC-MOSA convergence is lightly better than that of S-MOSA. This light improvement is caused by the large improvement of the cardinality, which are paradoxical metrics. The fronts of S-MOSA and MC-MOSA are given in Figure 8, where MC-MOSA front shows a large number of solutions that cover the whole front (almost equally distributed over the front). Generally, the fact that MC-MOSA found a higher number of optimal trade-offs offers to the decision maker more chance to meet a optimal decision. Furthermore, the MC-MOSA front shows widely better spread and distribution of the found solutions along the Pareto fronts, which are also mostly used as performance metrics for comparing MOO algorithms, see [1].

TABLE I
COMPARISON OF FRONTS CARDINALITY FOUND BY S-MOSA AND MC-MOSA.

Instance	S-MOSA	MC-MOSA
250 items	49.17 ± 2.54	$\mathbf{99 \pm 3.28}$
500 items	53.03 ± 4.42	$\mathbf{129.73 \pm 4}$
750 items	61.3 ± 4.83	$\mathbf{185.3 \pm 6.78}$

TABLE II
TWO-SETS COVERAGE BETWEEN S-MOSA AND MC-MOSA FOR TWO PROBLEM INSTANCES.

	250 items	
	S-MOSA	**MC-MOSA**
S-MOSA	#	0.34 ± 0.04
MC-MOSA	0.48 ± 0.06	#
	500 items	
	S-MOSA	**MC-MOSA**
S-MOSA	#	0.35 ± 0.04
MC-MOSA	0.38 ± 0.05	#
	750 items	
	S-MOSA	**MC-MOSA**
S-MOSA	#	0.32 ± 0.03
MC-MOSA	0.38 ± 0.04	#

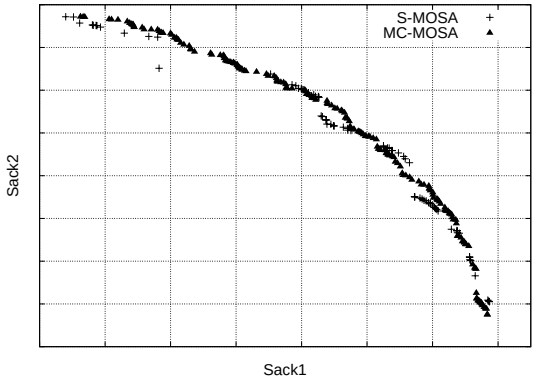

Fig. 8. A sample of S-MOSA and MC-MOSA fronts (MOKP-750)

C. S-MOSA and the Extended MC-MOSA Variants

The results obtained from the comparison of the S-MOSA and the sub-variants of the MC-MOSA concerning the sets cardinality and the coverage are presented in Tables III and IV, respectively. From the first view, one can see that the extensions of the MC-MOSA improve further the performance of the local search in the MOO. This is true for both, the sets size and the convergence. Especially, the re-annealing in the MC-MOSA allows the largest front size, which is almost three times larger than the fronts of the standard form. Furthermore, the improvement of the search space exploration did not penalize the convergence properties of the modified algorithm. The

front samples depicted in Figure 10 and Figure 9 show a clear advantage of the R-MC-MOSA, because its front is clearly clearly superior to the front of the standard form. This behavior is made possible by the allowed large moves in the search space, when the temperature is increased when the algorithm near the local optima. The advantage of this moves becomes important, when the search space becomes larger. However, the re-heating frequency and intensity (i.e. the level of re-heating) are two parameters of R-MC-MOSA that could affect the optimization performance, which should be investigated in future work. When the F-MC-MOSA and TS-MC-MOSA are compared, a strong similarity in their performance can be seen. This lets conclude that the introduced geometric cooling in the first half of the search did not bring any clear improvement. This means that an accelerated convergence in the early search phases does not damage the convergence characteristics of the algorithm over the entire optimization duration. This is guaranteed by an always higher acceptance probability of the relative bad solutions (or springs). However, the introduction of the multi-case in case (b) of S-MOSA has allowed a detailed classification of the solution that is seen as bad solution by the S-MOSA. A solution that does not allow a clear convergence toward the global optimum Pareto front, could allow an enlargement of the algorithm front size. In this case, any simple improvement is exploited by the MC-MOSA and theretofore allows it to explore further the search space by always escaping the local optimum solutions.

TABLE III
COMPARISON OF FRONTS CARDINALITY BETWEEN S-MOSA AND MC-MOSA SUB-VARIANTS.

Algorithm	**250 items**	**500 items**	**750 items**
S-MOSA	49.17 ± 2.54	53.03 ± 4.42	61.3 ± 4.83
F-MC-MOSA	124.67 ± 3.01	173.7 ± 5.38	123.23 ± 6.86
R-MC-MOSA	$\mathbf{140.47 \pm 3}$	$\mathbf{199.37 \pm 4.7}$	$\mathbf{293.83 \pm 7.5}$
TS-MC-MOSA	124.83 ± 3.18	173.93 ± 5.44	249.17 ± 6.06

TABLE IV
COMPARISON OF COVERAGE OF TWO SETS BETWEEN S-MOSA AND MC-MOSA SUB-VARIANTS.

	S-MOSA	F-MC	R-MC	TS-MC
		250 items		
S-MOSA	#	0.11 ± 0.02	0.03 ± 0.01	0.11 ± 0.02
F-MC	0.76 ± 0.04	#	0.2 ± 0.04	0.06 ± 0.06
R-MC	0.9 ± 0.03	0.7 ± 0.03	#	0.7 ± 0.03
TS-MC	0.76 ± 0.04	0.06 ± 0.06	0.2 ± 0.04	#
		500 items		
S-MOSA	#	0.04 ± 0.01	0 ± 0	0.04 ± 0.01
F-MC	0.9 ± 0.03	#	0.13 ± 0.03	0.04 ± 0.05
R-MC	0.98 ± 0.01	0.83 ± 0.04	#	0.82 ± 0.03
TS-MC	0.9 ± 0.03	0.05 ± 0.06	0.14 ± 0.03	#
		750 items		
S-MOSA	#	0 ± 0	0 ± 0	0.01 ± 0.01
F-MC	0.95 ± 0.02	#	0.38 ± 0.03	0.7 ± 0.03
R-MC	1 ± 0	0.35 ± 0.04	#	0.89 ± 0.02
TS-MC	0.96 ± 0.02	0.09 ± 0.02	0.09 ± 0.02	#

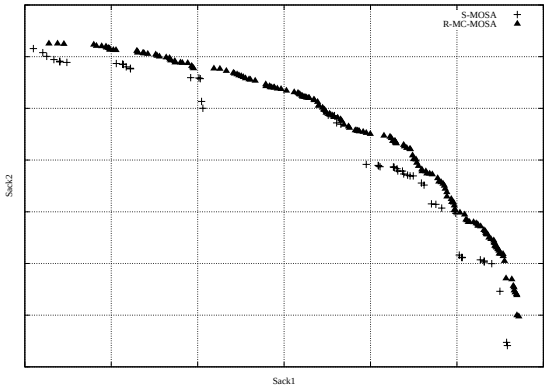

Fig. 9. Comparison of S-MOSA and R-MC-MOSA fronts for a 500 items MOKP

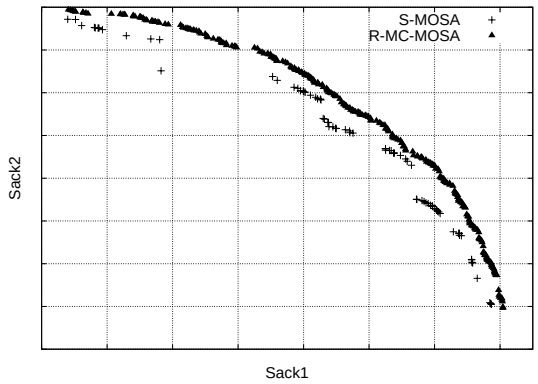

Fig. 10. Comparison of S-MOSA and R-MC-MOSA fronts for MOKP-750

VI. CONCLUSIONS

Most of the real-life problem are multi-objective problems. Because of that, different multi-objective algorithm have been developed over last decade. These algorithms have been applied in different engineering fields. One of the well-known category of these algorithms is that called multi-objective local search. This adapts the simulated annealing approach to the multi-objective optimization in different ways. In this work, we focused on the most referenced and recent variants, which is called Ulungu Multi-Objective Simulated Annealing, and referred in this work as Standard MOSA (S-MOSA). In this paper we have improved this variant by changing the algorithm of its acceptance decision, in order to allow a better exploration of the search space. This improvement results in a new approach called Multi-Case Multi-Objective Simulated Annealing (MC-MOSA). Indeed, the MC-MOSA has allowed to reach fronts that are larger than those reached by S-MOSA; however, with similar convergence properties (measured by the coverage of two sets). Some sub-variants of the MC-MOSA are also proposed in this paper, such as the Re-annealing MC-MOSA (R-MC-MOSA), the Fast MC-MOSA (F-MC-MOSA) and the Two-Stage MC-MOSA (TS-MC-MOSA). These subvariants did not only allow to improve

the size of the achieved fronts, but also to improve drastically the coverage properties of the original algorithm (i.e. MC-MOSA). In this paper, only the statistics of the performance metrics set cardinality and coverage of two sets are shown, but the new variants show the same high improvement according to all other metrics, such as the generational distance, the spread, etc.

In future work, these improved variants should be compared to other MOAs classes, such as the multi-objective evolutionary algorithms (MOEA), or Pareto-Archived Evolutionary Strategy (PAES), etc. Furthermore, these variants may be also used to hybridize these MOEA, in order to build new hybrid variants of the hybrid MOEAs. It is expected that the good MC-MOSA performance will be always achievable, even if they are mixed with other MOAs. However, this must be checked by numerical experiments.

REFERENCES

[1] E. Zitzler, "Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications," PhD Thesis at the Swiss Federal Institute of Technology Zurich, Zurich, Switzerland, November 1999

[2] J. D. Knowles, "Local-Search and Hybrid Evolutionary Algorithms for Pareto Optimization," PhD Thesis at the Department of Computer Science, The University of Reading, Reading, UK, January, 2002

[3] P. Serafini, "Simulated Annealing for Multiobjective Optimization Problems," in *Proc. of the 10th International Conference on Multiple Criteria Decision Making*, Taipei Taiwan, pp. 87-96, 1992.

[4] E. L. Ulungu, J. Teghem, P. H. Fortemps and D. Tuyttens, "MOSA Method: A Tool for Solving Multiobjective Combinatorial Optimization Problems," *Journal of Multicriteria Decision Analysis*, Vol. 8, pp. 221-236, 1999

[5] P. Czyzak and A. Jaszkiewicz, "Pareto Simulated Annealing - a Metaheuristic for Multiple-Objective Combinatorial Optimization," *Journal of Multi-Criteria Decision Analysis*, Vol. 7, No. 1, pp. 34-47, 1998

[6] A. Jaszkiewicz, M. Hapke and P. Kominek, "Performance of Multiple Objective Evolutionary Algorithms on a Distribution System Design Problem - Computational Experiment," in *Proc. of the First International Conference Evolutionary Multi-Criterion Optimization (EMO 2001)*, Zurich, Switzerland, March 7-9, 2001,

[7] M. P. Hansen, "Tabu Search for Multiobjective Optimization: MOTS," Technical Report Presented in *Proc. of 13th International Conference on MCDM*, Technical University of Denmark, 1997.

[8] V. Barichard, "Approches Hybrides pour les Problmes Multiobjectifs," PhD Thesis at the Ecole Doctorale d'Angers, University of Angers, Angers, France, November, 2003

[9] J. D. Schaffer, "Multiple Objective Optimization with Vector Evaluated Genetic Algorithms, Genetic Algorithms and Their Applications," in *Proc. of the First International Conference on Genetic Algorithms*, pp. 93-100, 1985.

[10] C. M. Fonseca and P. J. Fleming, "Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization," in *Proc. of the Fifth International Conference on Genetic Algorithms*, pp. 416-423, San Mateo, USA, 1993.

[11] J. Horn, N. Nafpliotis and D. E. Goldberg, "A Niched Pareto Genetic Algorithm for Multiobjective Optimization," in *Proc. of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence*, Piscataway USA, pp. 82-87, 1994

[12] N. Srivivas and K. Deb, "Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms," *Evolutionary Computation*, Vol. 2, No. 3, pp. 221-248, 1995

[13] K. Deb, A. Pratap, S. Agrawal and T. Meyarivan, "A Fast and Elitist Multiobjective Genetic Algorithm for Multi-Objective Optimization: NSGA-II," *IEEE Transactions on Evolutionary Computation*, Vol. 6, No. 2, April 2002.

[14] E. Zitzler and L. Thiele, "Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach," *IEEE Transactions on Evolutionary Computation*, Vol. 3, No. 4, pp. 257-271, 1999

[15] E. Zitzler, N. Laumanns and L. Thiele, "SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization," in *Proc. of Conf. on Evolutionary Methods for Design, Optimization, and Control*, Barcelona, Spain, pages 95-100, 2002

[16] E. Zitzler, M. Laumanns and S. Bleuler, "A Tutorial on Evolutionary Multiobjective Optimization," in *Metaheuristics for Multiobjective Optimization*, pages 3-38, Springer-Verlag, Berlin, Germany, 2004

[17] H. Ishibuchi and T. Murata, "Multi-objective genetic local search algorithm," in *Proc. of 3rd IEEE International Conference on Evolutionary Computation (ICEC'96)*, pp. 119-124, IEEE, 1996.

[18] A. Jaszkiewicz, "Genetic Local Search for Multiple Objective Combinatorial Optimization," *European Journal of Operational Research*, 137(1), pp. 50-71, 2002.

[19] H. Ishibuchi, T. Yoshida and T. Murata, "Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling," *IEEE Transactions on Evolutionary Computation*, Vol. 7, Issue 2, pp. 204- 223, April 2003

[20] A. Jaszkiewicz, "Multiple Objective Metaheuristic Algorithms for Combinatorial Optimization," Habilitation Thesis 360, Poznan University of Technology, Poznan, Poland, 2001

[21] C. Gil, R. Banos, M. G. Montoya and J. Gomez, "Performance of Simulated Annealing, Tabu Search, and Evolutionary Algorithms for Multi-objective Network Partitioning," *Algorithmic Operations Research*, Vol.1 (2006) pp.55-64.

[22] S. Kirkpatrick, C. D. Gelatt and M. P. Vecchi, "Optimization by Simulated Annealing," *Science*, vol. 220, pp. 671-680, 1983

[23] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller and E. Teller, "Equation of State Calculation by Fast Computing Machines," *Journal of Chemical Physics*, vol. 21, pp. 1087-1092, 1953

[24] I. H. Osman, "Heuristics for the Generalized Assignment Problem: Simulated Annealing and Tabu Search Approaches," *OR Spektrum*, Vol 17, pp. 211-225, 1995

[25] S. Martello and P. Toth, "*Knapsack Problem: Algorithms and Computer Implementations,* John Wiley & Sons, 1990

[26] A. Jaszkiewicz, "Multiple Objective MetaHeuristics Library in C++ MOMHLib++," Free Source Code Library, Institute of Computing Science, Poznan University of Technology, Poznan, Poland, 2001.

Abdelfatteh Haidine received the B.S. in Electronics and Telecommunications from the University Cadi Ayyad in Marrakech and the MSc in telecommunications from University Chouaib Doukkali El Jadida in 1999, in Morocco. Since 2000, he joined the University of Technology Dresden in Germany as research assistant. He worked in different European and German projects about power line communication networks. His actual research domain is the use of different operations research approaches in the planning and optimization of the access as well as the backbone network.

Ralf Lehnert received both his 1972 diploma degree and the 1979 Ph.D. degree in electrical engineering from Aachen University, Germany. Since 1980 he has been with the basic development department at Philips Communication Systems in Nuremberg, Germany, as the head of a group on applied research in performance evaluation of communication networks, traffic engineering and network planning. In July 1994 he took over the chair for telecommunications, as a full professor in the department of electrical engineering at Dresden University, Germany. His current research interests are in the field of performance evaluation of telecommunication systems, and the development of optimization approaches for the network planning and optimization.