Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 60 (2015) 448 – 457

Procedia Computer Science

19th International Conference on Knowledge Based and Intelligent Information and Engineering Systems

Indicator Based Ant Colony Optimization for Multi-Objective Knapsack Problem

Imen Ben Mansour[a], Ines Alaya[a]*

[a] *Cosmos-ENSI, University of Manouba, Manouba 2010, Tunisia*

Abstract

The use of metaheuristics to solve multi-objective optimization problems (MOP) is a very active research topic. Ant Colony Optimization (ACO) has received a growing interest in the last years for such problems. Many algorithms have been proposed in the literature to solve different MOP. This paper presents an indicator-based ant colony optimization algorithm called IBACO for the multi-objective knapsack problem (MOKP). The IBACO algorithm proposes a new idea that uses binary quality indicators to guide the search of artificial ants. These indicators were initially used by Zitzler and Künzli in the selection process of their evolutionary algorithm IBEA. In this paper, we use the indicator optimization principle to reinforce the best solutions by rewarding pheromone trails. We carry out a set of experiments on MOKP benchmark instances by applying the two binary indicators: epsilon indicator and hypervolume indicator. The comparison of the proposed algorithm with IBEA, ACO and other state-of-the-art evolutionary algorithms shows that IBACO is significantly better on most instances.

Peer-review under responsibility of KES International

Keywords: Multi-objective Ant colony optimization; binary-indicator optimization; multi-objective knapsack problem

1. Introduction

In real world applications, many problems involve optimizing multiple objectives and that are usually conflicting. Thus, there is not, usually, a single best solution but a set of solutions that are superior to others when considering all objectives. This set is called the Pareto set or non-dominated solutions.

* Corresponding author
E-mail address: imen.benmansour@ensi-uma.tn (I. Ben Mansour); ines.alaya@ensi-uma.tn (I. Alaya)

Peer-review under responsibility of KES International
doi:10.1016/j.procs.2015.08.165

These problems appear in many areas such as computer science, engineering, ecology, physics and chemistry. One of the widely studied problems in the literature is the multi-objective knapsack problem; the main goal of this problem consists in selecting a subset of items in order to maximize a multi-objective function while satisfying a set of knapsack constraints.

Several approaches have been proposed in the literature to solve MOP. Evolutionary algorithms (EAs) were intensively studied[26]. The Pareto-based approaches were very successful and presented an alternative to aggregation-based methods, which represent a simple way to transform a multi-objective problem into a single objective one. Recently, most successful EAs use indicator-based approaches and in particular the hypervolume indicator[5,28,29]. In[28] Zitzler and Künzli have demonstrated that indicator-specific search can yield results which are superior to well-known EAs such as SPEA2[30] and NSGA-II[10] with respect to the indicator under consideration.

The success of indicator based optimizers in the selection process of EAs motivates us to investigate the capability of this approach when used by artificial ants to find the best solutions in their optimization search. The main advantage of the indicator principle is that it could be easily adapted to other types of approaches as the proposed algorithms: indicator based multi-objective local search[7], the indicator-based search proposed in[16], and optimization with uncertainty[8].

In this paper, we propose an ACO algorithm that employs binary indicators that were used in the evolutionary algorithm IBEA[28]. The idea is to use these indicators to guide artificial ants to find the best solutions by laying pheromone relatively to the indicator values. This approach is different from the classical approaches in the literature, where the indicator optimizers are used to eliminate the worst solutions. In the proposed algorithm IBACO, these indicators are used to reinforce the best ones to guide the search of ants.

ACO is a metaheuristic inspired by the behaviour of real ants. This metaheuristic, described in[14,15], is a generalization of the first ant based algorithms: Ant System. This algorithm was proposed by Dorigo in[13] for the Traveling Salesman Problem. The basic idea is to use artificial ants to find the minimum cost path in a graph. ACO is inspired from the behavior of real ants; they use a chemical substance called pheromone to communicate their experience with the rest of the colony. An ant starts with empty solution and then adds iteratively solution components with respect to probabilities that depend on pheromone trails, previously accumulated by the colony, and a heuristic information that depends on the problem to solve. After the solutions construction, a pheromone update is launched to decide which ants modify the pheromone trails and how. To prevent unlimited accumulation, the pheromone trails progressively decrease by evaporation. This pheromone information gives idea about the quality of the path in order to attract ants towards the corresponding areas of the search space in the following iterations.

The ACO metheuristic has been successfully applied to different combinational optimization problems such as quadratic assignment problems[25], vehicle routing problems[9,17] and multidimensional knapsack problem[1]. Due to the good results found by ACO for mono-objective problems, many papers investigated the capabilities of ACO to solve multi-objective problems. In the literature, many taxonomies of multi-objective ACO algorithms (MOACO) was proposed[18,4,21]. These taxonomies categorize the MOACO algorithms relatively to different features. In[18], the proposed taxonomy categorizes MOACO algorithms by the number of pheromone matrices and the number of heuristic matrices. In[2], a generic ACO algorithm called m-ACO, is proposed and instantiated with four variants. This algorithm is parameterized by the number of ant colonies and the number of considered pheromone structures. For the four variants, pheromone trails are updated relatively to the objective functions values or relatively to constant values.

To apply ACO for multi-objective problems, there are many points to define. However, we consider that the key point is how to define the pheromone trails and how to use this pheromone information in the probability transition.

The quantity of pheromone laying on a component represents the past experience of the colony with respect to choosing this component. When there is only one objective function, this past experience is defined with respect to this objective. However, when there are several different objectives, one may consider two different strategies. A first strategy is to associate a single pheromone trail to each component, as proposed in[22,19,23,6,3]. In this case, the quantity of pheromone laid by ants is defined with respect to an aggregation of the different objectives. To avoid this function aggregation, a second strategy is to associate several pheromone trails to each component, as proposed in[20,11,17,9,12]. In this case, one usually associates a different colony of ants with each different objective, each colony having its own pheromone information. However, most of these approaches are obliged to aggregate these

pheromone structures to define the pheromone factor used in the probability transition. This probability is used when constructing solutions, at each step, to choose a candidate. It depends on two factors: the pheromone factor and a heuristic factor that is problem specific.

In this paper, the proposed algorithm IBACO uses a single pheromone structure employing a new idea. The pheromone information is defined relatively to the quality indicators values. These indicators, as defined in[28], assign each Pareto set approximation a real value reflecting its quality. They are considered as a natural extension of the Pareto dominance relation. Therefore, in IBACO ants use directly these values to deposit pheromone trails on the solutions they constructed relatively to their quality.

The paper is organized as follows. We recall in the next section definitions and notations of the multi-objcetive optimization problems. In section 3 we will present the indicator-based optimization principle, and we define dominance ranking techniques in terms of binary indicators. Then in section 4, we describe the IBACO algorithm. In section5, we present the experimental results, obtained by the application of IBACO to the multi-objective knapsack problem and compared to state-of-the-art EAs algorithms. Finally, conclusion and perspectives are discussed in section 6.

2. Multi-Objective Optimization Problems

A MOP is defined by a tuple (X,Z,C,F) such that X is a vector of n decision variables, i.e., $X=(x_1,...,x_n)$; Z is a vector of n value sets defining the domains of the decision variables, i.e., $Z=(z_1,...,z_n)$, such that z_i is the set of values that may be assigned to x_i; C is a set of constraints on X, i.e., a set of relations restricting the values that may be simultaneously assigned to the decision variables; and F is a vector of $m\geq 2$ objective functions $F(X)=(f_1(X),...,f_m(X))$; without loss of generality, we assume that these different objective functions have to be minimized.

The space of candidate solutions, noted $E=(X,Z,C)$, is the set of all value vectors $z\in Z$ satisfying all the constraints of C. We define a partial order relation on this set as follows: a solution $z\in E(X,Z,C)$ dominates a solution $z'\in E(X,Z,C)$, noted $z\prec z'$, iff z is at least as good as z' for each of the m criteria to optimize, and strictly better than z' for at least one of these criteria, i.e., iff $\forall i\in\{1,...,m\}, f_i(z)\leq f_i(z')$ and $\exists i\in\{1,...,m\}, f_i(z)<f_i(z')$.

We note that for each pair of solutions z and z' one and only one of these cases can occur:

- z dominates z', i.e., $z\prec z'$,
- z is dominated by z', i.e., $z'\prec z$,
- z and z' are equivalent in sense of dominance, i.e., $\neg(z\prec z')\wedge\neg(z'\prec z)$.

Solutions equivalent in sense of dominance are called Pareto-optimal, non-dominated solutions. The goal of a MOP is to find the Pareto set of all non-dominated solutions, i.e., $\{z\in E(X,Z,C) \mid \forall z'\in E(X,D,C), \neg(z'\prec z)\}$.

When using a metahuristic approach, the goal is to find Pareto set approximation that we note A. The set of all approximation sets is noted Ω.

3. Indicator Based Optimization

In this section we introduce the binary indicator optimization principle as defined and used in[28,32]:

Definition 1: Binary quality indicator *I*:

The function $I:\Omega\times\Omega\rightarrow\Re$, which assigns a real value to any pair of approximation sets $(A_1,A_2)\in\Omega\times\Omega$, is called binary quality indicator.

A binary quality indicator is a natural extension of the concept of Pareto-dominance on sets of objective vectors and can be directly used to calculate fitness. It can be used to compare two solutions or a single solution against a whole population. As defined in[28], the quality indicator *I* is compliant with the Pareto dominance relation:

Definition 2: A binary indicator *I* is denoted as dominance preserving if for all $x_1,x_2,x_3\in X$

a) $x_1\succ x_2\Rightarrow I(\{x_1\},\{x_2\})<I(\{x_2\},\{x_1\})$ and

b) $x_1\succ x_2\Rightarrow I(\{x_3\},\{x_1\})<I(\{x_3\},\{x_2\})$

In their study, Zitzler and Küenzli proposed the Indicator-Based Evolutionary Algorithm (IBEA), which use the principle of indicator to establish the selection process. The individual with the worst fitness value, in terms of the

quality indicator used, is deleted from the population and the fitness of the remaining individual is updated. In IBEA, two indicators are tested: the epsilon indicator I_ε (equation1) and the hypervolume indicator I_{HD} (equation 2).

$$I_{\varepsilon^+}(x_1,x_2)=min_\varepsilon\left(f_i(x_1)-\varepsilon\leq f_i(x_2)\right) \text{ for } i\in\{1,\ldots m\} \tag{1}$$

$I_\varepsilon(x_1,x_2)$, $x_1,x_2\in X$, is the minimum value by which x_1 must be moved to weakly dominate x_2.

$$I_{HD}(x_1,x_2)=\begin{cases} H(x_2)-H(x_1) & if \quad x_2\succ x_1 \\ H(x_1+x_2)-H(x_1) & else \end{cases} \tag{2}$$

$H(x_1)$ gives the hypervolume of the objective space dominated by x_1 and $I_{HD}(x_1,x_2)$ is the volume of the space that is dominated by x_2 but not by x_1.

To evaluate the quality a solution x_1 according to the whole population, different fitness assignment scheme are proposed in the literature. One of them is to sum up the indicator values with the respect to the rest of the population (equation 3).

$$Fit(x_1)=\sum_{x_2\in P\setminus\{x_1\}} I(\{x_2\},\{x_1\}) \tag{3}$$

In order to emphasize the influence of dominating solution over dominated ones an alternative approach can be used (see equation 4). In our experiments, we will use this formulation for the I_ε and I_{HD} indicators.

$$Fit(x_1)=\sum_{x_2\in P\setminus\{x_1\}} e^{-I(\{x_2\},\{x_1\})/\kappa} \tag{4}$$

κ is a scaling factor, when it's equal to 0, the same order relation between solution is obtained. Then, values near to 0 are preferred.

As shown previously in the paper, binary quality indicators are considered as a natural extension of the Pareto dominance relation, as defined in[28] the fitness scheme is also Pareto dominance compliant.

Theorem 1: let *I* be a binary quality indicator. If *I* is dominance preserving, then it holds that : $x_1\succ x_2\Rightarrow Fit(x_1)>Fit(x_2)$.

4. Indicator Based Ant Colony Optimization

This section presents the main contribution of the paper. Before defining our proposed algorithm IBACO, let us describe the problem to solve: the multi-objective knapsack problem (MOKP).

4.1. Problem Description

The MOKP consists in selecting a subset of items in order to maximize several utilities while satisfying a set of knapsack constraints. More formally, the problem is defined as follows:

$$\text{Maximize} \quad \sum_{j=1}^{n} p_j^k x_j \qquad k=1,\ldots,m$$

$$\text{Subject to} \quad \sum_{j=1}^{n} w_j^i x_j \leq b_i \qquad i=1,\ldots,q$$
$$x_j\in\{0,1\} \qquad j=1,\ldots,n$$

m denotes the number of objectives functions, *n* denotes the number of items, x_j the decision variable for the item o_j, *q* is the number of resource constraints, w_j^i the quantity of the resource *i* consumed by the item o_j, b_i the total quantity available for the resource *i*, p_j^k is the profit of the item o_j relatively to the objective *k*.

Algorithm IBACO:
Initialize the pheromone trails to τ_{init}
do
for each ant *ant* =1 to *NbAnts*
construct a solution S_{ant}
end for;
Calculate fitness values of solutions S_{ant}:

$$Fit(S_{ant}) = \sum_{S_{ant'} \in A \setminus \{S_{ant}\}} e^{-I(\{S_{ant'}\},\{S_{ant}\})/\kappa}$$

Update Pareto set P with new non dominated solutions
Update the pheromone trails
while maximum number of cycles reached

Fig. 1. Baseline of the IBACO algorithm.

Algorithm construct a solution:
choose a first item randomly $o_1 \in 1..n$

$S_{ant} \leftarrow o_1$

$Cand \leftarrow$ {objects o_j without violating any constraint}

While $Cand \neq 0$ **do**

Choose an item $o_j \in Cand$ with probability $P_s^{ant}(o_j)$

$$S_{ant} \leftarrow S_{ant} \cup o_j$$

Remove from *Cand* items that violates resource constraints

End while

Fig. 2. Solution Construction

4.2. Algorithm Description

The basic idea of IBACO is to apply the indicator-based approach as a search guide for the artificial ants to find the Pareto set approximation. Ants build solutions within a construction graph $G=(V,E)$, a complete graph where V is the set of items of the MOKP to solve. The pheromone trails are associated with vertices of this graph. IBACO follow an elitist version of Ant System scheme[13] in which only solutions of the Pareto set are authorized to lay pheromone. Pheromone trails are initialized to an initial amount. At each cycle of the algorithm, every ant constructs a solution. Once the construction phase is completed, the fitness assignment is carried out and the Pareto set is updated using the binary indicator. Then, the pheromone trails are updated according to the fitness values. The algorithm stops iterating when a maximum number of cycles is reached. The IBACO algorithm is outlined in Fig 1.

4.3. Solution Construction

In order to construct its solution, at each construction step, each ant *ant* choose an item o_j to add to the solution S_{ant} among the set of candidate items *Cand* with the probability $P_s(o_j)$ defined in equation 5. Then *Cand* is updated by removing the items that violate constraints. The solution construction algorithm is described in Fig 2.

$$p_S^{ant}(o_j) = \frac{[\tau_S(o_j)]^{\alpha} \cdot [\eta_S(o_j)]^{\beta}}{\sum_{o_j \in Cand} [\tau_S(o_j)]^{\alpha} \cdot [\eta_S(o_j)]^{\beta}} \tag{5}$$

where α and β are two parameters which determine the relative importance of the pheromone factor $\tau_S(o_j)$ and the heuristic factor $\eta_S(o_j)$. The pheromone factor is the pheromone trails laid on the candidate item. The heuristic factor is a specific problem heuristic information. We define this heuristic factor for the MOKP as an aggregation of the heuristic factor defined in [1] for the uniobjective multidimensional knapsack problem. Thus, the heuristic factor $\eta_S(o_j)$ used in the transition probability of the choice of a candidate item o_j is defined as follows: let $d_S(i)=b_i-\sum_{g \in S} r_{ig}$ be the remaining quality of the source i when the ant has constructed the solution S; we define the ratio:

$$h_S(o_j) = \sum_{i=1}^{q} \frac{w_j^i}{d_S(i)} \tag{6}$$

Which measures the tightness of the item j on all the constraints $i \in I..q$ relatively to the constructed solution S. Thus, the lower this ratio is, the more the item is profitable. We integrate in this ratio the profit p_j^k of the item o_j relatively to the objective k to obtain a pseudo-utility factor corresponding to the heuristic information of the objective k.

$$\eta_S^k(o_j) = \frac{p_j^k}{h_S(o_j)} \tag{7}$$

We can now define the heuristic factor formula as an aggregation of heuristic information of all objectives,

$$\eta_S(o_j) = \sum_{k=1}^{m} \eta_S^k(o_j) \tag{8}$$

4.4. Pareto Update

The Pareto set is updated by using binary indicator values. As shown earlier in the paper, the binary quality indicator is a natural extension of the Pareto dominance relation. I_ε and I_{HD} verify the dominance preserving relation; for instance, the I_ε values become negative as soon as S_{ant} dominates $S_{ant'}$[32].

4.5. Fitness Assignment

The fitness assignment ranks the solutions according to their quality, in terms of the quality indicator used. In other words, the best solution is the solution with the biggest fitness value. In IBACO algorithm, the fitness assignment is only carried out for the members of the Pareto set with respect to the rest of solutions. Fitness assignment formulation is defined in equation 4.

4.6. Pheromone Trails Update

This function updates the amount of pheromone laying on each item o_j of solution S_{ant} belonging to the Pareto set. First the pheromone trails are decreased, in order to simulate some kind of evaporation, and then the amount of pheromone is added on each item o_j of non-dominated solutions. Thus, the pheromone trails are updated according to equation 9 and 10:

$$\tau(o_j) \leftarrow (1-\rho) \times \tau(o_j) + \Delta\tau(o_j) \tag{9}$$

Where ρ is the evaporation factor, such that $0 < \rho < 1$ and

$$\Delta\tau(o_j) = \begin{cases} Fit\,(S_{ant}) & \text{if } o_j \in S_{ant} \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

If a solution S_{ant} dominates another solution $S_{ant'}$, the indicator value is negative and it contributes much more than the positive ones to the overall fitness. The fitness value is greater when a negative indicator value is assigned. Thereby, we have large fitness values for the dominating solutions and smaller values for dominated ones. In order to further attract ants towards the most promising search areas, in the IBACO algorithm, only the solutions of the Pareto set will be rewarded by the pheromone. During the pheromone update, the components of the best solutions, in terms of fitness value, receive the highest amount of pheromone according to their quality, following which they will be selected more often in the future cycles of the algorithm.

5. Experimental Results

In this section, we present the results of our experiments realized on multi-dimensional multi-objective knapsack problem (MOKP). The benchmark instances used for these experiments defined in[27]. The experiments are based on nine instances with 2, 3 and 4 objectives, in combination with 250, 500 and 750 items.

In our experimentation, a comparative study has been carried out on the two binary indicators I_ε and I_{HD}. We have chosen to compare our results with the indicator based algorithms IBEA$_\varepsilon$, IBEA$_{HD}$[28], the best variant of the generic ACO algorithm m-ACO$_4$ proposed in [2], HypE[5] and also the popular SPEA2[30]. HypE is a hypervolume-based search algorithm using Monte Carlo simulation. In the following, we discuss about the parameters setting and performance analysis protocol before detailing results.

5.1. Parameters Setting

We have done experimentations on some MOKP instances to choose the parameters values. We have set α, the weight of pheromone factor, to 1, and β, the weight of heuristic factor, to 5. The evaporation ratio ρ set to 0.01, the number of cycles to 100, *NbAnts,* the number of ants, to 20 and τ_{init} to 1. For the I_ε and I_{HD} indicators, κ has been set to 0.05. Moreover, I_{HD} needs a reference point, which has been set to [2,2], as suggested in[28].

IBEA, m-ACO, HypE and SPEA2 were configured as described in[28,2,5,30] respectively. In order to get fair comparison all algorithms have been implemented in C, run and tested on the same machine.

5.2. Performance Metrics

The performance assessment of multi-objective optimizers should consider at least three aspects: minimal distance to the Pareto-optimal front, adequate (good) distribution and maximum spread.

Various performance metrics to measure these three aspects have been introduced in the literature. We choose two measures: the size of the dominated space and coverage of two Pareto fronts[31]. Zitzler has shown that the two metrics are sufficient to measure the difference in performance between algorithms[31].

- The size of the dominated space (S) indicates how good the Pareto-front set is approximated by the non-dominated solutions of a given algorithm. The greater the size of the space dominated by the non-dominated solutions is, the more the solutions are close to Pareto-front set.
- The Coverage of two Pareto fronts compares two Pareto optimal sets A and B to each other. The coverage C (A,B) of the two Pareto fronts maps the ordered pair (A,B) to the interval [0, 1]:

$$C(A,B)=\frac{\left|\left\{b\in B \mid \exists a\in A: a\succ b\right\}\right|}{|B|} \tag{11}$$

Therefore, C(A,B) gives the fraction of B dominated by A. When C(A, B) = 1, all points in B are dominated by or equal to points in A, whereas when C(A, B) = 0, none of the points in B are covered by the set A. Note that C(A,B) is not necessarily equal to 1- C(B,A).

5.3. Comparative Results

In this section, we show the results of the C and S metrics calculated over 30 runs for all the compared algorithms on the benchmark instances. The results of both metrics are analysed through the non-parametric Wilcoxon rank-signed test[24] (*W*-test) in order to verify if the difference between the tested algorithms is statistically significant.

Table 1 and 2 compare, respectively, IBACO$_{HD}$ and IBACO$_\varepsilon$, with IBEA$_\varepsilon$, IBEA$_{HD}$, SPEA2, m-ACO and HypE algorithms relatively to the C average values calculated over 30 runs. These tables show that the solutions returned by IBACO$_{HD/\varepsilon}$ dominate always the ones returned by IBEA$_\varepsilon$, IBEA$_{HD}$, m-ACO, SPEA2 and HypE. Moreover, there

are no solutions returned by these compared algorithms that dominate any one returned by $IBACO_{HD}$ and $IBACO_{\varepsilon}$ since the values of the C measure are, for all these algorithms and for all the instances, equal to 0. The *W*-test shows that the results of $IBACO_{HD}$ are significantly larger than the tested algorithms. When comparing $IBACO_{\varepsilon}$ with $IBACO_{HD}$ relatively to the C measure, we have found very close results and no significant difference.

Table 1. The C measure statistics returned by $IBACO_{HD}$ compared to IBEA, SPEA2, m-ACO and HypE over 30 runs: Average values (*W*-Test). For *W*-Test, the level of significance considered is 0.05, (+) and (-) denote that $IBACO_{HD}$ C value is, respectively, significantly or not significantly larger than that of the compared algorithm.

	2.250	2.500	2.750	3.250	3.500	3.750	4.250	4.500	4.750
$C(IBACO_{HD}, IBEA_{\varepsilon})$	0.954 (+)	1 (+)	1 (+)	0.832 (+)	1 (+)	1 (+)	0.575 (+)	1 (+)	1 (+)
$C(IBEA_{\varepsilon}, IBACO_{HD})$	0	0	0	0	0	0	0	0	0
$C(IBACO_{HD}, IBEA_{HD})$	0.976 (+)	1 (+)	1 (+)	0.992 (+)	1 (+)	1 (+)	0.548 (+)	1 (+)	1 (+)
$C(IBEA_{HD}, IBACO_{HD})$	0	0	0	0	0	0	0	0	0
$C(IBACO_{HD}, SPEA2)$	0.997 (+)	1 (+)	1 (+)	0.950 (+)	1 (+)	1 (+)	0.896 (+)	1 (+)	1 (+)
C(SPEA2 , $IBACO_{HD}$)	0	0	0	0	0	0	0	0	0
$C(IBACO_{HD}, m\text{-}ACO)$	0.362 (+)	0.285 (+)	0.678 (+)	0.098 (+)	0.499 (+)	0.924 (+)	0.003 (+)	0.337 (+)	0.926 (+)
C(m-ACO, $IBACO_{HD}$)	0	0	0	0	0	0	0	0	0
$C(IBACO_{HD}, HypE)$	0.838 (+)	1 (+)	1 (+)	0.717 (+)	1 (+)	1 (+)	0.790 (+)	1 (+)	1 (+)
C(HypE , $IBACO_{HD}$)	0	0	0	0	0	0	0	0	0

Table 3 shows the results found by $IBACO_{HD/\varepsilon}$, and the other tested algorithms for the S average values and the statistical results of the *W*-test. Due to space limitations, only the $IBACO_{HD}$ is compared with the other algorithms.

From the table, we observe, first, that SPEA2 gives the worst results of S average values comparing to the indicator-based algorithms for all the instances. Therefore, one could conclude that indicator-based algorithms perform better than the classical Pareto based approach. When comparing $IBACO_{HD}$ and $IBACO_{\varepsilon}$ with the S measure, we remark that for most of instances there is no significant difference.

The comparison of our algorithm with IBEA for the two indicators shows that $IBACO_{HD/\varepsilon}$ outperforms widely and significantly $IBEA_{HD/\varepsilon}$. When comparing $IBACO_{HD/\varepsilon}$ with m-ACO and HypE relatively the S average value, we note that these latters slightly outperform our new algorithm for some small instances. However, IBACO clearly outperforms m-ACO and HypE for all the largest and hardest instances, and statistically, IBACO is always significantly larger.

The table 4 shows the processing times of all the tested algorithms with the parameters values recommended in[28,2,5,30]. We note that for both, $IBACO_{HD/\varepsilon}$, the execution times are clearly shorter than the other algorithms.

To summarize, the two indicators applied to IBACO obtain nearly the same results. IBACO significantly outperforms the other compared algorithms regardless of the indicator chosen. Moreover, IBACO needs much less processing time to find these results.

Table 2. The C measure statistics returned by $IBACO_{\varepsilon}$ compared to IBEA, SPEA2, m-ACO and HypE over 30 runs: Average values (*W*-Test). For *W*-Test, the level of significance considered is 0.05, (+) and (-) denote that $IBACO_{\varepsilon}$ C value is, respectively, significantly or not significantly larger than that of the compared algorithm.

	2.250	2.500	2.750	3.250	3.500	3.750	4.250	4.500	4.750
$C(IBACO_{\varepsilon}, IBEA_{\varepsilon})$	0.986 (+)	1 (+)	1 (+)	0.838 (+)	1 (+)	1 (+)	0.524 (+)	1 (+)	1 (+)
$C(IBEA_{\varepsilon}, IBACO_{\varepsilon})$	0	0	0	0	0	0	0	0	0
$C(IBACO_{\varepsilon}, IBEA_{HD})$	0.972 (+)	1 (+)	1 (+)	0.774 (+)	1 (+)	1 (+)	0.500 (+)	1 (+)	1 (+)
$C(IBEA_{HD}, IBACO_{\varepsilon})$	0	0	0	0	0	0	0	0	0
$C(IBACO_{\varepsilon}, SPEA2)$	0.999 (+)	1 (+)	1 (+)	0.952 (+)	1 (+)	1 (+)	0.843 (+)	1 (+)	1 (+)
$C(SPEA2, IBACO_{\varepsilon})$	0	0	0	0	0	0	0	0	0
$C(IBACO_{\varepsilon}, mACO)$	0.239 (+)	0.489 (+)	0.647 (+)	0.114 (+)	0.483 (+)	0.879 (+)	0.187 (+)	0.401 (+)	0.650 (+)
$C(mACO, IBACO_{\varepsilon})$	0	0	0	0	0	0	0	0	0
$C(IBACO_{\varepsilon}, HypE)$	0.841 (+)	1 (+)	1 (+)	0.734 (+)	1 (+)	1 (+)	0.492 (+)	0.999 (+)	1 (+)
$C(HypE, IBACO_{\varepsilon})$	0	0	0	0	0	0	0	0	0

Table 3. The S measure statistics returned by $IBACO_{HD/\varepsilon}$, IBEA, SPEA2, m-ACO and HypE over 30 runs: Average values (*W*-Test). For *W*-Test, the level of significance considered is 0.05, (+) and (-) denote that $IBACO_{HD}$ S value is, respectively, significantly or not significantly larger than that of the compared algorithm.

Instance	$IBACO_{HD}$	$IBACO_{\varepsilon}$	$IBEA_{HD}$	$IBEA_{\varepsilon}$	SPEA2	m-ACO	HypE
2.250	8.84e+7	8.86e+7 (-)	8.78e+7 (+)	8.65e+7 (+)	8.28e+7 (+)	8.83e+7 (+)	9.18e+7 (+)
2.500	3.67e+8	3.68e+8 (-)	3.15e+8 (+)	3.11e+8 (+)	2.98e+8 (+)	3.67e+8 (+)	3.20e+8 (+)
2.750	7.90e+8	7.95e+8 (-)	6.61e+8 (+)	6.60e+8 (+)	6.30e+8 (+)	7.79e+8 (+)	6.94e+8 (+)
3.250	7.36e+11	7.32e+11 (-)	7.23e+11 (+)	7.30e+11 (+)	6.65e+11 (+)	7.43e+11 (+)	7.71e+11 (+)
3.500	6.01e+12	6.01e+12 (+)	5.06e+12 (+)	4.97e+12 (+)	4.58e+12 (+)	5.88e+12 (+)	5.08e+12 (+)
3.750	2.13e+13	2.12e+13 (-)	1.54e+13 (+)	1.53e+13 (+)	1.44e+13 (+)	2.05e+13 (+)	1.56e+13 (+)
4.250	5.67e+15	5.68e+15 (-)	5.70e+15 (+)	5.61e+15 (+)	5.17e+15 (+)	5.69e+15 (+)	5.95e+15 (+)
4.500	9.35e+17	9.20e+17 (-)	7.57e+17 (+)	7.76e+17 (+)	6.85e+17 (+)	8.90e+17 (+)	7.93e+17 (+)
4.750	4.86e+17	4.85e+17 (-)	3.50e+17 (+)	3.49e+17 (+)	3.21e+17 (+)	4.62e+17 (+)	3.49e+17 (+)

6. Conclusion

In this paper, we have proposed a new indicator-based ACO algorithm for the multi-objective knapsack problem. This algorithm, called IBACO, uses the principle of binary indicator optimization proposed in IBEA algorithm[28].

The indicators are employed in order to guide the search of ants by laying pheromone trails relatively to a fitness assignment function. The experimental results on MOKP instances show that IBACO significantly outperforms the compared algorithms. In fact, the solutions returned by IBACO dominate always the ones returned by the other algorithms, and there are no solutions returned by these compared algorithms that dominate any one returned by IBACO.

The binary indicator search principle used in this paper by IBACO is different from that used in[28] since it is used to reinforce the best solutions and not to delete the worst ones as done in the selection phase of IBEA. The proposed algorithm shows its effectiveness compared to the tested algorithms for the MOKP. The efficiency of IBACO could vary according to the problem to solve and the choice of the parameter values. In fact, it would be interesting to apply IBACO on other multi-objective problems to test its effectiveness and scalability.

Table 4. Processing time for all algorithms over all instances (average values 'Avg' in seconds and standard deviation 'SD').

Instances		$IBACO_{HD}$		$IBACO_{\varepsilon}$		$IBEA_{HD}$		$IBEA_{\varepsilon}$		SPEA2		m-ACO		HypE	
		Avg	SD	Avg	SD	Avg	SD	Avg	SD	Avg	SD	Avg	SD	Avg	SD
2	250	3	0.10	4	0.23	119	6.15	218	8.26	211	3.58	408	26.47	219	1.78
	500	12	0.17	13	0.69	222	12.42	212	7.81	212	4.60	1450	146.41	218	1.86
	750	28	0.12	29	0.85	225	12.49	215	7.83	210	2.70	3114	232.32	218	1.65
3	250	4	0.05	4	0.05	210	3.78	215	9.99	212	5.93	441	13.49	224	1.39
	500	16	0.26	16	0.31	215	8.77	214	8.45	213	5.04	1511	40.77	223	2.56
	750	36	0.47	36	0.73	214	6.76	212	6.34	211	3.01	3695	483.65	222	1.53
4	250	5	0.15	5	0.18	209	2.74	216	10.35	212	1.69	1300	604.4	226	1.55
	500	20	0.24	20	0.44	210	3.60	213	7.82	218	12.07	1941	252.64	227	1.85
	750	47	1.18	49	3.71	211	3.70	215	11.31	212	1.94	4389	439.03	226	1.44

References

1. Alaya I, Solnon C, Ghedira K. Ant algorithm for the multi-dimensional knapsack problem. *Proceedings of International Conference on Bioinspired Optimization Methods and their Applications* (BIOMA 2004); 2004. p. 63–72.

2. Alaya I, Solnon C, Ghédira K. Ant Colony Optimization for Multi-objective Optimization Problems. *19th IEEE International Conference on Tools with Arti_cial Intelligence* (ICTAI'07); 2007. p. 450-457.
3. Angus D. Crowding population-based ant colony optimisation for the multi-objective travelling salesman problem. *on IEEE Symposium on Computational Intelligence in Multi-Criteria Decision- Making* (MCDM 2007); 2007. p. 333–340.
4. Angus D, Woodward C. Multiple objective ant colony optimisation, Swarm Intelligence 2009. 3(1): 69–85.
5. J. Bader, and E. Zitzler. Hype: An algorithm for fast hypervolume-based many-objective optimization. *IEEE Transactions on Evolutionary Computation* 2011; 19(1): 45-76.
6. Barán, B, Schaerer M. A Multiobjective Ant Colony System for Vehicle Routing Problem with Time Windows, Proc. *Twenty first IASTED International Conference on Applied Informatics*, Insbruck, Austria; 2003. p. 97-102.
7. Basseur M, Burke E.K. Indicator-based multiobjective local search. *Proceedings of the IEEE congress on evolutionary computation*; 2007. p.3100–3107 Singapore.
8. Basseur M, Zitzler E. Handling uncertainty in indicator-based multiobjective optimization. *International Journal of Computational Intelligence Research (IJCIR)*, 2006; 2(3):255-272.
9. Bullnheimer B, Hartl R.F., Strauss C. An Improved Ant system Algorithm for the Vehicule Routing Problem, Annals of Operations Research, 1999; 89: 319-328.
10. Deb K., Pratap A, Agarwal S, Meyarivan T..A fast and elitist multi- objective genetic algorithm: NSGA-II. *IEEE Transaction on Evolutionary Computation*, 2002; 6(2):181–197.
11. Dörner K, Gutjahr WJ, Hartl RF, Strauss C, Stummer C. Pareto Ant Colony Optimization: A metaheuristic approach to multiobjective portfolio selection, *Annals of Operations Research*, 2004; 131(1-4): 79-99.
12. Doerner K, Hartl RF, Teimann M.Are COMPETants More Competent for Problem Solving? The Case of Full Truckload Transportation, *Central European Journal of Operations Research*, 2003; 11(2): 115-141.
13. Dorigo M. Optimization, Learning, Natural Algorithms (in Italian), PhD thesis, Dipartimento di Elettronica, Politecnico di Milano, Italy. 1992.
14. Dorigo M, Stützle T. Ant Colony Optimization, MIT Press. 2004.
15. Dorigo M, Di Caro G. The Ant Colony Optimization Meta-Heuristic. in New Ideas in Optimization, D. Corne and M. Dorigo and F. Glover editors; 1999. p.11–32.
16. Emmerich M, Beume N, Naujoks, B. An EMO algorithm using the hypervolume measure as selection criterion. In *3rd international conference on evolutionary multi-criterion optimization (EMO 2005)* volume 3410 of *LNCS*, Springer; 2005.p. 62–76.
17. Gambardella L,Taillard ED, Agazzi G. MACS-VRPTW: A Multiple Ant Colony System for Vehicle Routing Problems with Time Windows, in F. G. D. Corne, M. Dorigo (ed.), New Ideas in Optimization, McGraw Hill, London, UK; 1999.p. 63-76.
18. Garcia-Martinez C, Cordon O, Herrera F. A taxonomy and an empirical analysis of multiple objective ant colony optimization algorithms for bi-criteria tsp. *European Journal of Operational Research*. July 2007; 180(1):116-148.
19. Gravel M, Price WL, Gagne C. Scheduling Continuous Casting of Aluminium using a Multiple Objective Ant Colony Optimization Metaheuristic, *European Journal of Operational Research*, 2002; 143(1): 218-229.
20. Iredi S, Merkle D, Middendorf M. Bi-Criterion Optimization with Multi Colony Ant Algorithms, *First International Conference on Evolutionary Multi-criterion Optimization* (EMO'01), vol. 1993, Lecture Notes in Computer Science; 2001. p. 359-372.
21. López-Ibáñez M, Stützle T. The automatic design of multiobjective ant colony optimization algorithms. *Evolutionary Computation, IEEE Transactions on Evolutionary Computation* 2012; 16(6): 861-875.
22. Mariano CE, Morales E. A Multiple Objective Ant-Q Algorithm for the Design of Water Distribution Irrigation Networks, Technical Report n HC-9904. Instituto Mexicano de Tecnologa del Agua, Mexico, 1999.
23. McMullen PR. An Ant Colony Optimization Approach to Addressing a JIT Sequencing Problem with Multiple Objectives, *Artificial Intelligence in Engineering* 2001; 15(3): 309-317.
24. Sheskin D J, Handbook of Parametric and Nonparametric Statistical Procedures. 4th ed. Chapman & Hall: CRC; 2007.
25. Stützle T, Hoos H. MAX −MIN Ant System, *Journal of Future Generation Computer Systems* 2000; 16: 889- 914.
26. Zhou A, Qu BY, Li H, Zhao SZ, Suganthan P N, Zhang Q. Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation* 2011;1(1): 32-49.
27. Zitzler E, Thiele L. Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. *IEEE Transactions on Evolutionary Computation* 1999; 3(4): 257–271.
28. Zitzler E, Künzli S. Indicator-based selection in multiobjective search. *8th International Conference on Parallel Problem Solving from Nature (PPSN VIII)*; 2004. p. 832–842.
29. Zitzler E, Brockhoff D, Thiele L. The hypervolume indicator revisited: On the design of Pareto-compliant indicators via weighted integration. *In Evolutionary multi-criterion optimization* ; 2007. p.862-876.
30. Zitzler E, Laumanns M, Thiele L. SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization. In K. Giannakoglou et al., editors, *Evolutionary Methods for Design, Optimisation and Control with Application to Industrial Problems* (EUROGEN 2001), International Center for Numerical Methods in Engineering (CIMNE); 2001. p. 95–100.
31. Zitzler E. Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications, 1999. Ph.D. Thesis, Swiss Federal Institute of Technology, Zurich, Switzerland.
32. Zitzler E, Laumanns M, Thiele L, Fonseca CM, da Fonseca V G. Performance Assessment of Multiobjective Optimizers: An Analysis and review. *IEEE Transactions on Evolutionary Computation*; 2003. p. 117–132.