Proceedings of the 16th Conference on Computer Science and Intelligence Systems pp. 173–180
DOI: 10.15439/2021F22
ISSN 2300-5963 ACSIS, Vol. 25

InterCriteria Analyzis of Hybrid Ant Colony Optimization Algorithm for Multiple Knapsack Problem

Stefka Fidanova
Institute of Information and Communication Technology
Bulgarian Academy of Sciences
Acad. G. Bonchev Str., bl. 25A, Sofia, Bulgaria
E-mail: stefka@parallel.bas.bg

Maria Ganzha
System Research Institute
Polish Academy of Sciences
Warsaw, Poland
E-mail: maria.ganzha@ibspan.waw.pl

Olympia Roeva
Institute of Biophysics and Biomedical Engineering
Bulgarian Academy of Sciences
Acad. G. Bonchev Str., bl. 105, Sofia, Bulgaria
E-mail: olympia@biomed.bas.bg

***Abstract*—The local search procedure is a method for hybridization and improvement of the main algorithm, when complex problems are solved. It helps to avoid local optimums and to find faster the global one. In this paper we apply InterCriteria analysis (ICrA) on hybrid Ant Colony Optimization (ACO) algorithm for Multiple Knapsack Problem (MKP). The aim is to study the hybrid algorithm behavior comparing with traditional ACO algorithm. Based on the obtained numerical results and on the ICrA approach the efficiency and effectiveness of the proposed hybrid ACO, combined with appropriate local search procedure are confirmed.**

***Index Terms*—Local Search, Ant Colony Optimization, InterCriteria analysis, Knapsack Problem**

I. Introduction

ENGINEERING applications normally lead to complex decision make problems. Large scale problems can not be solved with traditional numerical methods. It is a challenge to develop a new techniques, which have simple structure and easy application, and can find near optimal solution even the information about the problem is incomplete. In most of the cases these problems are HP-hard.

Nature inspired methods are more appropriate for solving NP-hard optimization problem, than other methods, because they are flexible and use less computational resources. They are base on stochastic search. The most popular methods are Evolutionary algorithm [28], [45], which simulates the Darwinian evolutionary concept, Simulated Annealing [32] and Gravitation search algorithm [39], Tabu Search [42] and Interior Search [43]. The ideas for swarm-intelligence based algorithms come from behavior of animals in the natures. The representatives of this type of algorithms are Ant Colony Optimization [16], Bee Colony Optimization [29], Bat algorithm [46], Firefly algorithm [47], Particle Swarm Optimization [31], Gray Wolf algorithm [38] and so on.

Between the best methods for solving combinatorial optimization problems is Ant Colony Optimization (ACO). The impulse for this method comes from the behavior of real ants. They always find the shortest path from the food to the nest. The ants leave a trail called a pheromone and follow the trail with the most concentrated pheromone.

The problem is represented with a help of a graph and the solutions are paths in a graph. The optimal solution for minimization problems is a shortest path, and for maximization problems is a longest path in a graph. The solution construction starts from random node of the graph and next nodes are included applying probabilistic rule. The pheromone is imitated by numerical information corresponding to the quality of the solution.

ACO is applied to many types of optimization problems. The idea for application of ant behavior for solving combinatorial optimization problems is done by Marco Dorigo twenty five years ago [15], [16], [18]. At the beginning it is applied on traveling salesman problem. Later it is successfully applied on a lot of complex optimization problems. During the years, various variants of ACO methodology was proposed: ant system [18]; elitist ants [18]; ant colony system [17]; max-min ant system [44]; rank-based ant system [18]; ant algorithm with additional reinforcement [21]. They differ in pheromone updating. For some of them is proven that they converge to the global optimum [18]. Fidanova et all [22]–[24] proposed semi-random start of the ants comparing several start strategies. The method can be adapted to dynamic changes of the problem in some complex biological problems [19], [20], [44].

Sometimes the metaheuristic algorithm, can not avoid local optimums. Appropriate Local Search (LS) procedure can help to escape them and to improve algorithm efficiency. We apply ACO on Multiple Knapsack Problem (MKP). A local search procedure, related with a specificity of MKP is constructed and combined with ACO to improve the algorithm performance and to avoid local optimums [24]. InterCriteria analysis (ICrA) is applied on the numerical results obtained by the traditional ACO and hybrid ACO in order to estimate the algorithms behavior. The approach ICrA has been applied for a large area of problems, e.g. [1], [2], [14], [25]. Published results show the applicability of the ICrA and the correctness of the approach.

The rest of the paper is organized as follows: The definition of the MKP is in Section 2. ACO algorithm is presented in Section 3. Local Search procedure is described in Section 4. Short notes on ICrA approach are presented in Section 5. Numerical results and a discussion are in Section 6. Conclusion remarks are done in Section 7.

II. Multiple Knapsack Problem

In knapsack problem is given a set of items with fixed weights and values. The aim is to maximize the sum of the values of the items in the knapsack, while remaining within the capacity of the knapsack. Each item can be selected only ones.

Multiple Knapsack Problem (MKP) is a generalization of the single knapsack problem and instead to have only one knapsack, there are many knapsacks with diverse capacity. Each item is assigned to maximum one of the knapsacks without violating any of the knapsacks capacity. The purpose is to maximize the total profit of the items in the knapsacks.

MKP is a special case of the generalized assignment problem [36]. It is a representative of the subset problems. Economical, industrial and other types of problems can be represented by MKP. Resource allocation in distributed systems, capital budgeting, cargo loading and cutting stock problems [30] are some of the applications of the problem. One important real problem which is represented as MKP is patients scheduling [3]. MKP is related with bin packing problem where the size of the bins can be variable [40] and cutting stock problem for cut row materials [30]. Other application is multi-processor scheduling on uniformly related machines [35]. Other difficult problem which leads to MKP is crypto-systems and generating keys [30]. One early application of MKP is tests generation [27]. MKP is a model large set of binary problems with integer coefficients [33], [36].

MKP is NP-hard problem and normally is solved with some metaheuristic method such as genetic algorithm [37], tabue search [48], swarm intelligence [34], ACO algorithm [21], [26].

We will define MKP as resource allocation problem, where m is the number of resources (the knapsacks) and n is the number of the objects. The object j has a profit p_j. Each resource has its own budget (knapsack capacity) and consumption r_{ij} of resource j by object i. The purpose is maximization of the profit within the limited budget.

The mathematical formulation of MKP can be as follows:

$$\begin{gathered}\max \sum_{j=1}^{n} p_j x_j \\ \text{subject to} \sum_{j=1}^{n} r_{ij} x_j \le c_i \quad i = 1,\dots,m \\ x_j \in \{0,1\} \quad j = 1,\dots,n\end{gathered} \tag{1}$$

There are m constraints in this problem, so MKP is also called m-dimensional knapsack problem. Let $I = \{1,\dots,m\}$ and $J = \{1,\dots,n\}$, with $c_i \ge 0$ for all $i \in I$. A well-stated MKP assumes that $p_j > 0$ and $r_{ij} \le c_i \le \sum_{j=1}^{n} r_{ij}$ for all $i \in I$ and $j \in J$. Note that the $[r_{ij}]_{m\times n}$ matrix and $[c_i]_m$ vector are both non-negative.

The MKP partial solution is represented by $S = \{i_1, i_2, \dots, i_j\}$ and the last element included to S, i_j is not used in the selection process for the next element. Thus the solution of MKP have not fixed length.

III. Ant Colony Optimization Algorithm

NP-hard problems require the use of huge resources and therefore cannot be solved by exact or traditional numerical methods, especially when they are large scale. We apply metaheuristic method aiming to find approximate solution using reasonable resources [18], [26].

Firs Marco Dorigo applies ideas coming from ants behavior to solve complicate optimization problems 30 years ago [16]. Some modifications are proposed by him and by other authors for algorithm improvement. The modifications concern pheromone updating [18]. The algorithm is problem dependent. Very important is representation of the problem by a graph. Thus the solutions represent paths in the graph. The ants look for an optimal path, taking in to account problem constraints.

The transition probability $P_{i,j}$, is a product of the heuristic information $\eta_{i,j}$ and the pheromone trail level $\tau_{i,j}$ related to the selection of node j if the previous selected node is i, where $i, j = 1, \dots, n$.

$$P_{i,j} = \frac{\tau_{i,j}^{a} \cdot \eta_{i,j}^{b}}{\sum\limits_{k \in Unused} \tau_{i,k}^{a} \cdot \eta_{i,k}^{b}}, \tag{2}$$

where $Unused$ is the set of unused nodes.

At the beginning the pheromone is initialized with a small constant value τ_0, $0 < \tau_0 < 1$. Every time the ants build a solution, the pheromone is bring up to date [18].The elements of the graph with more pheromone are more tempting to the ants.

The main update rule for the pheromone is:

$$\tau_{i,j} \leftarrow \rho \cdot \tau_{i,j} + \Delta\tau_{i,j}, \tag{3}$$

where parameter ρ decreases the value of the pheromone, like evaporation in a nature decreases the quantity of old pheromone. $\Delta\tau_{i,j}$ is a new deposited pheromone, which depends on the value of the objective function, corresponding to this solution.

The first step, when ACO is applied on some combinatorial optimization problem is representation of the problem by graph. In our case the items are related with the nodes of the graph and the edges fully connect the nodes. The pheromone is deposited on the arcs of the graph.

Second step is construction of appropriate heuristic information. This step is very important, because the heuristic information is the main part of the transition probability function and the search process depends mainly on it. Normally the heuristic information is a combination of problem parameters.

Let $s_j = \sum_{i=1}^{m} r_{ij}$. For heuristic information we use:

$$\eta_{ij} = \begin{cases} p_j^{d_1}/s_j^{d_2} & \text{if } s_j \neq 0 \\ p_j^{d_1} & \text{if } s_j = 0 \end{cases} \quad (4)$$

where $d_1 > 0$ and $d_2 > 0$ are parameters. Hence the objects with greater profit and less average expenses will be more desirable. Thus is increased the probability to include more items and most profitable items. This can lead to maximization of the total profit, which is the objective of this problem.

IV. LOCAL SEARCH PROCEDURE

At times is used hybridization of the used method, for algorithm performance improvement. The goal is avoid some disadvantages of the main method. A possibility for hybridization is one of the methods to be basic and the other only helps to improve the solutions. Most used hybridization manner is local improvement or at the end of the iteration to apply some problem dependent local search procedure.

The Local Search (LS) procedure is used to perturbs current solution and to generate neighbor solutions [41]. LS generates neighbor solutions in a local set of neighbors. The best solution from the set is compared with the current solution. If it is better, it is accepted as a new current solution.

A LS procedure which is consistent with MKP has been developed and combined with ACO algorithm in our previous work [26]. The MKP solution is represented by binary string where 0 corresponds to not chosen item and 1 corresponds to item included in the solution. Two positions are randomly chosen. If the value of one of the positions is 0 we replace it with 1 and if the value of other position is 1 we replace it with 0 and vice versa. The feasibility of the new solution is verified. If the solution is feasible we compare it with the current (original) solution. The perturbed solution is accepted if its value of the objective function is greater, than of the original one.

We apply this LS procedure ones on each iteration on each solution, disregarding if the new constructed solution is better than current one or not. Thus the proposed LS works without significant increase of the used computational resources.

V. INTERCRITERIA ANALYSIS

Based on the apparatuses of index matrices [4], [6], [8], [9] and intuitionistic fuzzy sets (IFSs) [5], [7], [10], authors in [11] propose a new approach named InterCriteria analysis. Briefly presented, an intuitionistic fuzzy pair (IFP) [12] is an ordered pair of real non-negative numbers $\langle a, b\rangle$, where $a, b \in [0, 1]$ and $a + b \leq 1$, that is used as an evaluation of some object or process. According to [12], the components (a and b) of IFP might be interpreted as degrees of "membership" and "non-membership" to a given set, degrees of "agreement" and "disagreement", etc.

Let O denotes the set of all objects being evaluated, and $C(O)$ is the set of values assigned by a given criteria C (i.e., $C = C_p$ for some fixed p) to the objects, i.e.,

$$O \stackrel{\text{def}}{=} \{O_1, O_2, O_3, \ldots, O_n\},$$
$$C(O) \stackrel{\text{def}}{=} \{C(O_1), C(O_2), C(O_3), \ldots, C(O_n)\}.$$

Let $x_i = C(O_i)$. Then the following set can be defined:

$$C^*(O) \stackrel{\text{def}}{=} \{\langle x_i, x_j\rangle | i \neq j \,\&\, \langle x_i, x_j\rangle \in C(O) \times C(O)\}.$$

Further, if $x = C(O_i)$ and $y = C(O_j)$, $x \prec y$ if $i < j$ will be written.

In order to find the agreement of different criteria, the vectors of all internal comparisons for each criterion are constructed, which elements fulfill one of the three relations R, $\overline{R}$ and $\tilde{R}$. The nature of the relations is chosen such that for a fixed criterion C and any ordered pair $\langle x, y\rangle \in C^*(O)$:

$$\langle x, y\rangle \in R \Leftrightarrow \langle y, x\rangle \in \overline{R}, \quad (5)$$
$$\langle x, y\rangle \in \tilde{R} \Leftrightarrow \langle x, y\rangle \notin (R \cup \overline{R}), \quad (6)$$
$$R \cup \overline{R} \cup \tilde{R} = C^*(O). \quad (7)$$

For example, if "R" is the relation "$<$", then $\overline{R}$ is the relation "$>$", and vice versa.

When comparing two criteria the degree of "agreement"is determined as the number of matching components of the respective vectors (divided by the length of the vector for normalization purposes).

Let the respective degrees of "agreement" and "disagreement" are denoted by $\mu_{C,C'}$ and $\nu_{C,C'}$. In the most of the obtained pairs $\langle \mu_{C,C'}, \nu_{C,C'}\rangle$, the sum $\mu_{C,C'} + \nu_{C,C'}$ is equal to 1. However, there may be some pairs, for which this sum is less than 1. The difference

$$\pi_{C,C'} = 1 - \mu_{C,C'} - \nu_{C,C'} \quad (8)$$

is considered as a degree of "uncertainty.

VI. COMPUTATIONAL RESULTS AND DISCUSSION

The proposed hybrid ACO algorithm for MKP is tested on 10 test MKP instances from Operational Research Library "OR-Library" available within WWW access at *http://people.brunel.ac.uk/ mastjjb/jeb/info.html*. Every test problem consists of 100 items and 10 constraints/knapsacks. We prepare a software, which realizes our hybrid algorithm. The software is coded in $C{+}{+}$ program language and is run on Pentium desktop computer at 2.8 GHz with 4 GB of memory. The ACO algorithm parameters are fixed experimentally as follows:

- Number of iterations = 300, Number of ants = 20;
- ρ = 0.5, τ_0 = 0.5;
- a = 1, b = 1 and d_1 = 1.

We perform 30 independent runs with every one of the test instances, because the algorithm is stochastic and to guarantee the robustness of the average results. We apply ANOVA test for statistical analysis and thus we guarantee the significance of the difference between the average results. The names of

TABLE I: Test instances

Instance	Name	
	Hybrid ACO	Traditional ACO
MKP 100 $\times$ 10-01	$P1_h$	$P1_t$
MKP 100 $\times$ 10-02	$P2_h$	$P2_t$
MKP 100 $\times$ 10-03	$P3_h$	$P3_t$
MKP 100 $\times$ 10-04	$P4_h$	$P4_t$
MKP 100 $\times$ 10-05	$P5_h$	$P5_t$
MKP 100 $\times$ 10-06	$P6_h$	$P6_t$
MKP 100 $\times$ 10-07	$P7_h$	$P7_t$
MKP 100 $\times$ 10-08	$P8_h$	$P8_t$
MKP 100 $\times$ 10-09	$P9_h$	$P9_t$
MKP 100 $\times$ 10-10	$P10_h$	$P10_t$

TABLE II: Traditional ACO performance

	$P1_t$	$P2_t$	$P3_t$	$P4_t$	$P5_t$	$P6_t$	$P7_t$	$P8_t$	$P9_t$	$P10_t$
run1	22089	22452	20936	21481	21751	21810	21537	21634	22213	40594
run2	21954	22055	20966	21318	21606	21864	21659	21596	22398	40701
run3	21935	21912	21023	21318	21463	21912	21526	21516	22065	40647
run4	22030	21914	20732	21556	21519	21903	21470	21337	22191	40617
run5	21875	21990	21120	21451	21903	21970	21360	21689	22152	40489
run6	21970	21999	21114	21619	21736	21756	21426	21729	22125	40646
run7	21974	21990	21085	21740	21641	21654	21522	21515	22398	40714
run8	22041	22120	21032	21918	21811	22053	21584	21550	22109	40550
run9	21893	21924	21187	21335	21716	21864	21587	21725	22398	40581
run10	21984	22104	20822	21719	21767	21864	21509	21550	22078	40594
run11	21787	21950	21042	21661	21673	22047	21595	22067	22398	40659
run12	21916	22675	21182	21629	21818	21834	21426	21550	22101	40646
run13	22031	22027	20869	21490	21843	22123	21466	21508	22398	40584
run14	22188	21975	21203	21736	21811	21864	21601	21550	22398	40627
run15	21889	22119	21204	21740	21716	21824	21509	21573	22086	40515
run16	22009	22101	20877	21705	21952	21779	21409	21506	22039	40498
run17	21880	21990	21085	21531	21736	21713	21394	21579	22398	40680
run18	21958	22102	20872	21335	21811	22053	21596	21550	22105	40589
run19	22015	21963	20841	21861	21581	21864	21624	21729	22324	40404
run20	22054	22027	21007	21815	21679	22140	21392	21729	22398	40367
run21	22093	22027	20833	21607	21811	21816	21509	21496	22039	40496
run22	22074	22065	20960	21701	21711	21903	21509	21496	22398	40737
run23	22003	22027	20976	21759	21685	21864	21522	21339	22039	40594
run24	22169	22005	21003	21490	21735	21898	21511	21520	22156	40317
run25	22091	22106	20808	21964	21622	21840	21434	21614	22398	40664
run26	21945	21899	21103	21335	21417	22053	21426	21516	22059	40319
run27	22086	22196	21069	21774	21944	22241	21590	21629	22398	40728
run28	21926	21975	21003	21437	21922	21864	21531	21503	22398	40636
run29	21929	22177	20799	21681	21736	21907	21479	21530	22398	40498
run30	22030	21931	20925	22061	21434	21864	21509	21366	22398	40756

the test instances are presented in Table I. In Table II and Table III observed numerical results for all 30 runs are listed.

In Table IV are reported average results for every one of the test instances over 30 runs. We compare ACO algorithm combined with local search procedure (hybrid ACO) with traditional ACO algorithm. On the last row is reported average computational time, in seconds, of the two variants of ACO algorithm.

Table IV shows that for eight of ten instances hybrid ACO algorithm outperforms the traditional one. For the instances MKP 100×10-02 and MKP 100×10-10 the results are statistically the same. The main problem with hybrid algo-

TABLE III: Hybrid ACO performance

	$P1_h$	$P2_h$	$P3_h$	$P4_h$	$P5_h$	$P6_h$	$P7_h$	$P8_h$	$P9_h$	$P10_h$
run1	22206	22047	21292	21885	21811	21940	21509	22004	22270	40647
run2	21968	22186	21089	21962	21811	21957	21509	21616	22097	40557
run3	21970	22074	21233	22139	21716	21934	21530	21550	22087	40598
run4	22130	22028	21687	21701	21811	22024	21415	21729	22294	40679
run5	22107	22168	21020	21885	21811	22047	21522	21729	22285	40647
run6	22138	22027	21222	21736	21798	21980	21522	21551	22257	40522
run7	22367	22027	21416	21962	21811	21900	21437	21729	22140	40538
run8	22051	22028	21261	21420	21790	22048	21509	21648	22125	40710
run9	21867	22104	20861	21780	21811	21985	21531	21729	22398	40710
run10	21910	22027	21090	21666	21811	21987	21537	21729	22398	40647
run11	22133	22027	20758	21854	21844	22011	21655	21729	22398	40583
run12	22164	22074	20848	21885	21798	22053	21330	21653	22154	40662
run13	21949	22044	21090	22099	21736	21987	21409	21729	22398	40664
run14	22067	22102	20954	21921	21798	21987	21587	21729	22117	40683
run15	21889	22027	21236	21801	21811	22113	21509	21729	22069	40689
run16	21999	22213	21185	21561	21811	21900	21509	21650	22429	40636
run17	21926	22065	21017	22174	21914	21937	21509	21550	22191	40489
run18	21914	22027	21097	21542	21855	21864	21509	21550	22247	40714
run19	22008	22028	20953	21490	21796	22053	21624	21683	22398	40677
run20	21840	22151	21166	21893	21804	21987	21418	21729	22193	40728
run21	21993	22155	20839	21656	21811	21891	21584	21550	22398	40565
run22	22130	22106	21134	21962	21798	22063	21533	21658	22479	40742
run23	21958	22060	20881	21885	21811	21987	21426	21729	22152	40503
run24	22014	22027	21373	22023	21811	21924	21511	21729	22152	40514
run25	22072	22027	20925	21953	21811	21987	21509	21697	22123	40650
run26	22088	22104	20857	22052	21974	21987	21509	21550	22429	40751
run27	21928	22027	20934	21864	21811	21987	21509	21550	22218	40658
run28	21933	22110	20916	21885	21811	22121	21509	21689	22398	40499
run29	21970	22027	21281	21793	21832	21987	21509	21729	22398	40598
run30	21993	22027	21064	21951	21811	22050	21509	21550	22479	40540

TABLE IV: Comparison of ACO performance

Instance	Hybrid ACO	Traditional ACO
MKP 100 × 10-01	**22022.73**	21989.43
MKP 100 × 10-02	22071.46	22081.36
MKP 100 × 10-03	**21089.3**	21027.63
MKP 100 × 10-04	**21846**	21635.3
MKP 100 × 10-05	**21814.3**	21717.3
MKP 100 × 10-06	**21989.26**	21869.73
MKP 100 × 10-07	**21506.26**	21477.3
MKP 100 × 10-08	**21672.53**	21606.43
MKP 100 × 10-09	**22272.36**	22257
MKP 100 × 10-10	40626.66	40623.26
computational time	64.052 s	65.552 s

rithms, when some global method is combined with local search procedure, is increasing of computational time. We try to propose efficient and in a same time less time consuming local search. We only change randomly chosen position in a solution to 0, if it is 1 and another randomly chosen position to 1 if it is 0. Thus is generated only one neighbor solution. If this solution is better than the current one, it is accepted and used for pheromone updating instead of the solution constructed by the ant. We apply this procedure to each of the solutions. As is seen from Table IV the increase of computational time, when our local search is applied is only 2.34%.

Thus we can conclude that proposed local search procedure is efficient and effective. The algorithm performance is improved, without significant increase of the computational time.

To support these claims, the obtained numerical results were analyzed using ICrA. The input matrix for ICrA has the following form index matrix Table V:

The obtained by ICrA results are listed in Table VI (μ-values) and Table VII (ν-values). The π-values are also presented (see Table VIII). The results between the same instances but for different ACO algorithms are presented. For example, relations between $P1_t - P1_h$, $P2_t - P2_h$, $P3_t - P3_h$, etc. are considered for further analysis (presented in bold results in Tables VI, VII and VIII).

According to [13] the results show that the considered

TABLE V: Index matrix for ICrA

$$
\begin{array}{c|cccc}
 & \text{run1} & \text{run2} & \ldots & \text{run30} \\
\hline
P1_t & val_{P1_{t,1}} & val_{P1_{t,2}} & \vdots & val_{P1_{t,30}} \\
P2_t & val_{P2_{t,1}} & val_{P2_{t,2}} & \vdots & val_{P2_{t,30}} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
P10_t & val_{P10_{t,1}} & val_{P10_{t,2}} & \vdots & val_{P10_{t,30}} \\
P1_h & val_{P1_{h,1}} & val_{P1_{h,2}} & \vdots & val_{P1_{h,30}} \\
P2_h & val_{P2_{h,1}} & val_{P2_{h,2}} & \vdots & val_{P2_{h,30}} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
P10_h & val_{P10_{h,1}} & val_{P10_{h,2}} & \vdots & val_{P10_{h,30}}
\end{array}
\qquad (9)
$$

TABLE VI: Degree of agreement – $\mu_{C,C'}$-values

μ	$P1_t$	$P2_t$	$P3_t$	$P4_t$	$P5_t$	$P6_t$	$P7_t$	$P8_t$	$P9_t$	$P10_t$
$P1_h$	**0.55**	0.46	0.51	0.5	0.45	0.38	0.45	0.46	0.37	0.55
$P2_h$	0.42	**0.37**	0.48	0.3	0.48	0.39	0.36	0.37	0.34	0.36
$P3_h$	0.57	0.57	**0.44**	0.5	0.46	0.46	0.46	0.43	0.39	0.47
$P4_h$	0.47	0.39	0.55	**0.4**	0.35	0.47	0.4	0.41	0.51	0.61
$P5_h$	0.25	0.34	0.33	0.27	**0.32**	0.36	0.35	0.34	0.34	0.34
$P6_h$	0.4	0.45	0.5	0.57	0.43	**0.5**	0.44	0.46	0.46	0.48
$P7_h$	0.41	0.3	0.44	0.38	0.4	0.38	**0.52**	0.51	0.36	0.38
$P8_h$	0.4	0.42	0.38	0.42	0.4	0.37	0.39	**0.42**	0.42	0.29
$P9_h$	0.45	0.38	0.35	0.44	0.49	0.49	0.39	0.39	**0.42**	0.41
$P10_h$	0.55	0.52	0.5	0.52	0.46	0.63	0.48	0.54	0.38	**0.38**

TABLE VII: Degree of disagreement – $\nu_{C,C'}$-values

ν	$P1_t$	$P2_t$	$P3_t$	$P4_t$	$P5_t$	$P6_t$	$P7_t$	$P8_t$	$P9_t$	$P10_t$
$P1_h$	**0.44**	0.51	0.47	0.48	0.52	0.54	0.52	0.5	0.4	0.43
$P2_h$	0.44	**0.46**	0.38	0.56	0.37	0.4	0.48	0.46	0.38	0.49
$P3_h$	0.43	0.4	**0.55**	0.49	0.52	0.47	0.5	0.54	0.4	0.51
$P4_h$	0.49	0.55	0.42	**0.55**	0.59	0.43	0.54	0.53	0.26	0.36
$P5_h$	0.46	0.36	0.38	0.43	**0.37**	0.35	0.37	0.34	0.27	0.37
$P6_h$	0.51	0.45	0.41	0.33	0.46	**0.36**	0.44	0.42	0.31	0.43
$P7_h$	0.41	0.5	0.37	0.43	0.41	0.4	**0.27**	0.28	0.34	0.43
$P8_h$	0.37	0.35	0.39	0.35	0.36	0.36	0.37	**0.33**	0.24	0.47
$P9_h$	0.47	0.53	0.57	0.48	0.41	0.4	0.52	0.51	**0.34**	0.51
$P10_h$	0.43	0.44	0.48	0.45	0.5	0.29	0.47	0.42	0.39	**0.6**

criteria pairs, are in dissonance or in strong dissonance. This means that the both compared ACO algorithms (hybrid and traditional ones) performed differently in case of all 10 various instances.

The results obtained form ICrA are correct and reliable, taking into account the observed values of $\pi_{C,C'}$-values. Only for relations between $P5_t - P5_h$, $P8_t - P8_h$ and $P9_t - P9_h$, there are some high $\pi_{C,C'}$-values, respectively 0.31, 0.25 and 0.24. The obtained estimates for the degree of agreement and the degree of disagreement have a high degree of uncertainty.

VII. Conclusion

In this paper we propose hybrid ACO algorithm for solving MKP. The algorithm is combination of traditional ACO algorithm and local search procedure. Proposed algorithm is tested on 10 benchmark MKP. The achieved results show the efficiency and effectiveness of the proposed local search procedure. The hybrid algorithm performs better than the traditional one, while the calculation time increases only with 2.34%.

Obtained results are analyzed by ICrA approach. The anal-

TABLE VIII: Degree of uncertainty – $\pi_{C,C'}$-values

π	$P1_t$	$P2_t$	$P3_t$	$P4_t$	$P5_t$	$P6_t$	$P7_t$	$P8_t$	$P9_t$	$P10_t$
$P1_h$	**0.01**	0.03	0.02	0.02	0.03	0.08	0.03	0.04	0.23	0.02
$P2_h$	0.14	**0.17**	0.14	0.14	0.15	0.21	0.16	0.17	0.28	0.15
$P3_h$	0	0.03	**0.01**	0.01	0.02	0.07	0.04	0.03	0.21	0.02
$P4_h$	0.04	0.06	0.03	**0.05**	0.06	0.1	0.06	0.06	0.23	0.03
$P5_h$	0.29	0.3	0.29	0.3	**0.31**	0.29	0.28	0.32	0.39	0.29
$P6_h$	0.09	0.1	0.09	0.1	0.11	**0.14**	0.12	0.12	0.23	0.09
$P7_h$	0.18	0.2	0.19	0.19	0.19	0.22	**0.21**	0.21	0.3	0.19
$P8_h$	0.23	0.23	0.23	0.23	0.24	0.27	0.24	**0.25**	0.34	0.24
$P9_h$	0.08	0.09	0.08	0.08	0.1	0.11	0.09	0.1	**0.24**	0.08
$P10_h$	0.02	0.04	0.02	0.03	0.04	0.08	0.05	0.04	0.23	**0.02**

ysis shows that the both algorithms performs differently for the considered 10 instances, i.e. the behavior of the proposed hybrid ACO is importantly different from that of the traditional ACO algorithm, or the local search procedure perturbs significantly the search process.

Through the application of ICrA approach the efficiency and effectiveness of the proposed hybrid ACO agorithm are confirmed.

ACKNOWLEDGMENT

The development of the proposed hybrid ACO algorithm with local search procedure has been funded by the Grant DFNI DN 12/5 and Grant No BG05M2OP001-1.001-0003, financed by the Science and Education for Smart Growth Operational Program and co-financed by the European Union through the European structural and Investment funds. The study of ACO algorithms behavior based on ICrA approach has been funded by the Grant KP-06-N22/1.

Author Contributions: The authors contributed equally to the work.

REFERENCES

[1] M. Angelova, O. Roeva and T. Pencheva. InterCriteria Analysis of Crossover and Mutation Rates Relations in Simple Genetic Algorithm. In *Proceedings of the 2015 Federated Conference on Computer Science and Information Systems*, Vol. 5, pages 419–424, 2015.

[2] Antonov A., Dependencies Between Model Indicators of the Basic and the Specialized Speed in Hockey Players Aged 13-14, Trakia Journal of Sciences, Vol. 18, Suppl. 1, pp 647-657, 2020.

[3] Idil Arsik, Pinar Keskinocak, Jennifer Coppola, Kirthana Hampapur, Yitong He, Haozheng Jiang,Dani Regala, Nick Tailhardat, and Kristin Goin. Effective and equitable appointment scheduling in rehabilitation centers.INFORMS Annual Meeting 2017, Oct. 22-25 2017.

[4] K. Atanassov. *Index matrices: Towards an augmented matrix calculus*. Springer International Publishing Switzerland, 2014.

[5] K. Atanassov. Intuitionistic Fuzzy Sets. VII ITKR Session, Sofia, 20-23 June 1983, Reprinted: *International Journal Bioautomation*, 20(S1):S1–S6, 2016.

[6] K. Atanassov. Generalized index matrices. *Comptes rendus de l'Academie bulgare des Sciences*, 40(11):15–18, 1987.

[7] K. Atanassov. *On intuitionistic fuzzy sets theory*. Springer, Berlin, 2012.

[8] K. Atanassov. On index matrices, Part 1: Standard cases. *Advanced Studies in Contemporary Mathematics*, 20(2):291–302, 2010).

[9] K. Atanassov. On index matrices, Part 2: Intuitionistic fuzzy case. *Proceedings of the Jangjeon Mathematical Society*, 13(2):121–126, 2010.

[10] K. Atanassov. Review and New Results on Intuitionistic Fuzzy Sets, Mathematical Foundations of Artificial Intelligence Seminar, Sofia, 1988, Preprint IM-MFAIS-1-88. Reprinted: *International Journal Bioautomation*, 20(S1):S7–S16, 2016.

[11] K. Atanassov, D. Mavrov and V. Atanassova. Intercriteria decision making: A new approach for multicriteria decision making, based on index matrices and intuitionistic fuzzy sets. *Issues in on Intuitionistic Fuzzy Sets and Generalized Nets*, 11:1–8, 2014.

[12] K. Atanassov, E. Szmidt and J. Kacprzyk. On intuitionistic fuzzy pairs. *Notes on Intuitionistic Fuzzy Sets*, 19(3):1–13, 2013.

[13] K. Atanassov, V. Atanassova and G. Gluhchev. InterCriteria Analysis: ideas and problems. *Notes on Intuitionistic Fuzzy Sets*, 21(1):81–88, 2015.

[14] V. Atanassova, D. Mavrov, L. Doukovska and K. Atanassov. Discussion on the Threshold Values in the InterCriteria Decision Making Approach. *Notes on Intuitionistic Fuzzy Sets*, 20(2): 94–99, 2014.

[15] M. Birattari, T. Stutzle, L.Paquete, K. Varrentrapp, A racing algorithm for configuring metaheuristics, Proceedings of the Genetic and Evolutionary Computation Conference, pp. 11–18, 2002.

[16] Bonabeau E., Dorigo M. and Theraulaz G., *Swarm Intelligence: From Natural to Artificial Systems*, New York,Oxford University Press, 1999.

[17] M. Dorigo, L. Gambardella, Ant colony system : A cooperative learning approach to the traveling salesman problem, IEEE Transactions on Evolutionary Computation vol. 1, 53–66, 1996.

[18] Dorigo M, Stutzle T., *Ant Colony Optimization*, MIT Press, 2004.

[19] S. Fidanova, I. Lirkov, 3d protein structure prediction, Analele Universitatii de Vest Timisoara, vol. XLVII, pp. 33–46, 2009.

[20] S. Fidanova, An improvement of the grid-based hydrophobic-hydrophilic model, Int. J. Bioautomation, vol. 14, pp. 147–156, 2010.

[21] S. Fidanova, ACO algorithm with additional reinforcement, Int Conf. from Ant Colonies to Artificial Ants, Lecture Notes in Computer Science 2463, 292–293, 2003.

[22] S. Fidanova,K. Atanassov, P. Marinov, Generalized nets and ant colony optimization, Bulg. Academy of Sciences Pub. House, 2011.

[23] S. Fidanova, K. Atanassov, P. Marinov, Start strategies of ACO applied on subset problems, Numerical Methods and Applications, Lecture Notes in Computer Science, 6046, 248–255, 2011.

[24] S.Fidanova, K. Atanassov, P. Marinov, Intuitionistic fuzzy estimation of the ant colony optimization starting points, Large Scale Scientific Computing, Lecture Notes in Computer Science 7116, 219–226, 2012.

[25] S. Fidanova, O. Roeva and M. Paprzycki. InterCriteria Analysis of ACO Start Strategies, In *Proceedings of the 2016 Federated Conference on Computer Science and Information Systems*, Vol. 8, pages 547–550, 2016.

[26] Fidanova S. Hybrid Ant Colony Optimization Algorithm for Multiple Knapsack Problem. 5th IEEE International Conference on Recent Advances and Innovations in Engineering (ICRAIE), IEEE, 2021, DOI:10.1109/ICRAIE51050.2020.9358351, 1-5.

[27] Feuerman, Martin; Weiss, Harvey (April 1973). "A Mathematical Programming Model for Test Construction and Scoring". Management Science. 19 (8): 961–966

[28] D.E. Goldberg, B. Korb, K. Deb, Messy Genetic Algorithms: Motivation Analysis and First Results, Complex Systems, vol. 5(3), pp. 493 – 530, 1989.

[29] D. Karaboga, B. Basturk, Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems, Advances in Soft Computing: Foundations of Fuzzy Logic and Soft Computing, LNCS 4529, 789-798, 2007.

[30] Kellerer H., Pferschy U., Pisinger D. (2004) Multiple Knapsack Problems. In: Knapsack Problems. Springer, Berlin,

[31] J. Kennedy, R. Eberhart, Particle Swarm Optimization. Proceedings of IEEE International Conference on Neural Networks. IV. pp. 1942–1948, 1995.

[32] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, Optimization by simulated annealing Science (New York, N.Y.) (New York, N.Y.), 13 (220) (1983), pp. 671-680

[33] G.Kochenberger, G.McCarl, F.Wymann, An heuristic for general integer programming, Decision Sciences vol. 5, 34–44, 1974.

[34] Jonas Krause, Jelson Cordeiro, Rafael Stubs Parpinelli, and Heitor Silverio Lopes. A survey ofswarm algorithms applied to discrete optimization problems. InSwarm Intelligence and Bio-Inspired Computation, pages 169–191. Elsevier, 2013

[35] E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan, and D. B. Shmoys. Sequencing and scheduling:algorithms and complexity. In S. C. Graves et al. , editor,Handbooks in OR and MS, volume 4, pages445–522. Elsevier Science Publishers, 1993

[36] G.Leguizamon, Z.Michalevich, A new version of ant system for subset problems, Int. Conf. on Evolutionary Computations vol. 2, 1459–1464, 1999.

[37] Qing Liu, Tomohiro Odaka, Jousuke Kuroiwa, Haruhiko Shirai, and Hisakazu Ogura. A newartificial fish swarm algorithm for the multiple knapsack problem.IEICE TRANSACTIONS onInformation and Systems, 97(3):455–468, 2014

[38] S. Mirjalili, S. M. Mirjalili, A. Lewis, Grey Wolf Optimizer, Advances in Engineering Software, vol. 69, pp. 46–61, 2014.

[39] M.R. Mosavi, M. Khishe, G.R. Parvizi, M.J. Naseri, M. Ayat, Training multi-layer perceptron utilizing adaptive best-mass gravitational search algorithm to classify sonar dataset Archive of Acoustics, 44 (1) (2019), pp. 137-151

[40] F. D. Murgolo. An efficient approximation scheme for variable-sized bin packing.SIAM Journal onComputing, 16(1):149–161, 1987.

[41] Schaffer, A. A., Yannakakis, M.: *Simple Local Search Problems that are Hard to Solve*. Society for Industrial Applied Mathematics Journal on Computing, Vol 20 (1991) 56–87.

[42] I.H. Osman, Metastrategy simulated annealing and tabue search algorithms for the vehicle routing problem, Annals of Operations Research, 41 (4) (1993), pp. 421-451

[43] S. Ravakhah, M. Khishe, M. Aghababaee, E. Hashemzadeh, Sonar false alarm rate suppression using classification methods based on interior search algorithm, International Journal of Computer Science and Network Security, 17 (7) (2017), pp. 58-65

[44] T. Stutzle, H. Hoos, Max min ant system, Future Generation Computer Systems, vol. 16, 889–914, 2000.

[45] P.A. Vikhar, Evolutionary algorithms: A critical review and its future prospects, Proceedings of the 2016 International Conference on Global Trends in Signal Processing, Information Computing and Communication (ICGTSPICC). Jalgaon, pp. 261–265, 2016.

[46] X.S. Yang, A New Metaheuristic Bat-Inspired Algorithm, Nature Inspired Cooperative Strategies for Optimization, Studies in Computational Intelligence, 284: 65–74, 2010.

[47] X.S. Yang, Nature-Inspired Metaheuristic Algorithms, Luniver Press, 2008.

[48] Andrew J Woodcock and John M Wilson. A hybrid tabue search/branch and bound approach tosolving the generalized assignment problem.European journal of operational research, 207(2):566–578, 2010