Proceedings of the Federated Conference on
Computer Science and Information Systems pp. 197–203

ISBN 978-83-60810-51-4

Hierarchical Heterogeneous Ant Colony Optimization

Miroslav Rusin, Elena Zaitseva
Department of Informatics,
University of Žilina, Univerzitná
8215/1, Žilina, Slovakia
Email: Miroslav.Rusin,
{Elena.Zaitseva}@fri.uniza.sk

***Abstract*—Ant Colony Optimization (ACO) is used to solve problems with multiple objectives. Various extensions have been implemented to the traditional approach to improve algorithm performance or quality of solutions. In this paper we propose a novel ACO-based method that involves heterogeneity and hierarchy in the area of automated meal plans. The hierarchy consists of 2 levels: at the first there are ants working in a fairly traditional way (a worker); at the second there is an ant manager. Each worker has its own plan and searches the unique environment. The second level ant monitors a group of workers. Experimental results show that this approach is capable to tackle the task in a reasonable time and quality.**

I. Introduction

INFORMATION and communication technologies (ICT) play vital role in our everyday life. One of the application areas is medicine and healthcare. ICT are used here for various purposes: diagnostics, treatment, monitoring, data management, communication, administrative tasks [1], reliability analysis [2], patient assistance, etc.

In real life there are many problems with few objectives that conflict with each other, for instance, to minimize costs, maximize performance; maximize reliability at the same time, etc. Such problems exist in medicine as well. Various methods were developed in order to solve multi-objective problems. On one side there are exact methods that compute optimal solution, on the other side there are empirical methods. In reality we are often interested in finding "good" solutions in reasonable time; therefore we decided to utilize the second group of algorithms also called heuristics. There are Genetic algorithms, Particle Swarm Optimization, Ant Colony Optimization, Tabu Search, Simulated Annealing belonging to this group.

The world faces new crisis due to the rising number of its population and a number of patients with chronic diseases that hurt not only individuals but also damage national economies in form of higher healthcare related costs. India, Russian federation and China were predicted to lose between $200 and $550 billion in national income due to heart disease, stroke and diabetes [3]. Statistics by WHO [4] stated that in 2008 1.5 billion people were overweight, of these 500 million obese. Overweight and obesity are caused by eating more energy-dense food and by a lack of physical activity. Therefore change in eating habits is required and can prevent this condition. For this purpose we proposed a multi-objective model for generating personalized meal plans. The complete description of the model can be found in [5].

We chose to apply Ant Colony Optimization (ACO) for this model as it has been successfully implemented for many combinatorial problems, e.g. travelling salesman problem, the quadratic assignment problem, the sequential ordering problem, production scheduling, timetabling, project scheduling, vehicle routing, telecommunication routing, investment planning, staff scheduling, etc [6]. ACO is capable of solving multi-objective problems, therefore presents a suitable approach for us.

ACO is a metaheuristics inspired by nature. As the name implies the main principle is based on behavior of the ant colony, a group of ants that cooperates together in order to survive. In their everyday life, they are often confronted with the objective to find food and carry it back to the colony by the shortest route. The fact that they are able to communicate is even more interesting when we consider that these animals are almost blind. The medium used for communication is the substance called pheromone. They can sense it and lay it down to mark their trail. Thus they can decide based on the sensed pheromone amount and follow the paths that contain more of this substance, laid down previously by other ants. More pheromone on a trail attracts more ants. This process is characterized as a positive feedback loop. Example result of this process is in Fig. 1. The bolder line represents richer pheromone trail which was heavily used by ants leaving great pheromone amount. This trail is the shortest one from Colony to Food in this scenario.

The main characteristics of this approach are positive feedback, distributed computation, and the use of a constructive greedy heuristic. Positive feedback accounts for rapid discovery of good solutions, distributed computation avoids premature convergence, and the greedy heuristic helps find acceptable solutions in the early stages of the search process.

We identified several reviews concerned with the multi-objective ant colony optimization (MOACO). López-Ibáñez *et al.* in [7] examined several MOACO methods by studying algorithmic choices and performance based on experimental results.

They conclude that many of existing approaches share more similarities than differences, i.e. some can be reformu-

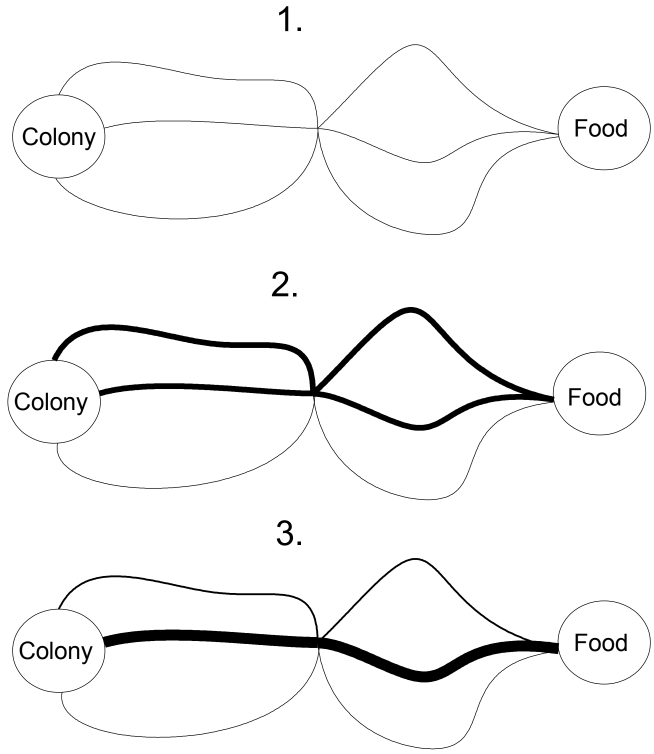

Fig. 1 Illustration of an ant searching for the shortest road. Numbers at the top denote the sequence of phases in the process. The third picture shows the final phase where the shortest road is found.

lated to gain a common model with some modifications. García-Martínez *et al.* [6] proposed taxonomy as well; this classifies methods according to two criteria: the use of only one or several pheromone trails and the use of only one or several heuristic functions, i.e. four algorithm families exist. They also conducted experiments and concluded that the operation mode of ACO algorithm itself has greater influence on the quality of gained solutions than the membership in the specific algorithm family. Angus *et al.* introduced an original taxonomy for MOACO [8] based on the analysis of features common to the reviewed ACO algorithms: the choice of pheromone model, the solution construction process, solution evaluation in terms of individual objectives or all objectives, pheromone matrix (matrices) update method, and Pareto solution archival structure.

In this paper a novel MOACO model is described. We introduce hierarchy of cooperative heterogeneous agents (ants) that operate in heterogeneous search spaces in order to produce a single solution as opposed to other models where a single ant is able to find a complete solution on its own. We apply this approach to the problem of automated meal planning.

The paper is organized as follows: Section 2 provides the analysis results of heterogeneity in ACO algorithms, Section 3 describes the novel MOACO-based method, Section 4 contains experimental part of the work; Section 5 offers discussion and Section 6 conclusion.

II. Heterogeneity and Hierarchy in Ant Colony Optimization Models

Ant or agents that have different properties can improve the search process and sometimes are necessary in order to find a solution. The idea of heterogeneity in ACO is not new. Heterogeneous ants were introduced in [9]. ACO algorithm is used here for the purpose of the global path planning of the mobile robot. In this approach there exist ants with different qualities (sight, speed, function). The use of heterogeneous ants proved to be superior to the classic ACO approach.

Another type of heterogeneous ant approach to ACO can be found in [10]. Authors use heterogeneous ant colonies to balance diversity and convergence. They create communication rule and different pheromone rules for the colonies. However, inside of a colony, ants behave traditionally in order to find the optimal solution.

Hara *et al.* [11] proposed the use of heterogeneous ants to improve search performance in TSP. Totally there are two types of ants: classic and exploratory. The new type exists to overcome some problems of classic ants. They explore, create some short routes, they choose to visit only those cities that are “near” and in some predefined conditions they may give up creating the round tour.

Heterogeneous ACO has been implemented in robotics [12]. Ducatelle *et al.* showed similarity between ants and the robots in their swarm robotic system. There exist foot-bots and eye-bots that mutually adapt for the purpose of foraging task and together are able to find the shortest route. Foot-bots obey instructions issued by eye-bots and eye-bots observe foot-bots to adapt the given instructions.

Different ant sensitivity to the pheromone was proposed by Chira *et al.* [13]. There are ants with higher sensitivity that follow strong pheromone trails. On the other hand, ants with lower sensitivity behave more randomly. This model enables to sustain balance between exploration and exploitation.

Well-known Travelling Salesman problem (TSP), which often serves as a benchmark problem, is partitioned into several smaller TSPs in [14]. Thus a task hierarchy is created, where clusters from TSP form the subtasks. Each ant moves through the cluster and then selects the next cluster to move to. Connection cities between clusters are determined by local greedy search.

Model that joins heterogeneity and hierarchy was introduced by Brown *et al.* [15]. They use three types of ants: queens, workers and zombies to solve clustering problem. A worker behaves as a traditional ant; it is restricted to a cluster region and has a limited lifespan. It gives items that do not fall into that particular cluster to a queen. A queen maintains the cluster and communicates with other queens. A zombie is a worker that exceeded its lifespan and its task is to deliver items still in its possession.

The existing approaches use heterogeneity or hierarchy to improve performance or solution quality. In contrast with these, we propose a new method that exploits a heterogeneous ant group organized in a hierarchy to produce a single solution, i.e. each ant works on a solution component, however, is unable to find a complete solution. This is assembled by combining all of these partial solutions.

III. Hierarchical Heterogeneous Ant Colony Optimization (HHACO)

In this paper we focus on a heuristic approach that tries to produce "good" solutions in reasonable time. We explore the possibility to use ACO for the problem of automated meal planning. However, this requires some adaption to the original concept where all ants are identical and the randomness is the force which helps to produce diverse solutions. In meal plan the solution can be divided into various levels with different conditions to be fulfilled. This requires greater cooperation from the ants than before because one ant can no longer create the required single solution. For this purpose a group of heterogeneous ants, where each ant has its own plan, is needed. The plan in this context consists of a set of constraints for recommended nutrition intake or constraints concerning some other diet recommendations. To illustrate it on the specific problem, see Fig. 2. We use a group of ants instead of a single ant for the whole task as each meal is a separate object with its own constraints and being searched for in its own search space. It seems only logical to assign a task of meal creation to one ant.

Each solution component is searched for by a different agent. At the bottom level there are ants working in a similar way as originally introduced in . However, the process of their work was modified due to the specificity of our problem and will be discussed later. Their progress is monitored by an agent we call ant manager from the upper level. The top level node represents the desired complete solution and an agent called top ant manager is in control. We call this model Hierarchical Heterogeneous Ant Colony Optimization (HHACO).

A. Ant Colony Optimization for meal plan creation

The proposed model in [5] was designed as a multi-objective model. The objective function contained three criteria: personalization, diversity and suitability. We updated the model so now there are these two criteria: personalization, diversity and dietician recommendations. Suitability is included in the admissibility constraint now. Let I be the set of food items and J be the set of meals in a day. We show only the definition of personalization criterion in the view of our experiments (1):

$$P=\sum_{i\in I}\sum_{j\in J} p_i * Y_{ij} \tag{1}$$

The constant p_i expresses the patient's attitude towards the food component.

If $p_i > 0$, $i \in I$, attitude towards i^{th} food component is positive

If $p_i = 0$, $i \in I$, attitude towards i^{th} food component is neutral

If $p_i < 0$, $i \in I$, attitude towards i^{th} food component is negative

Bivalent variable Y_{ij} is defined:

If $Y_{ij} = 1$, $i \in I$, $j \in J$, then then i^{th} food item is used for a meal j

If $Y_{ij} = 0$, $i \in I$, $j \in J$, then then i^{th} food item is not used for a meal j

We want to maximize this criterion (1).

ACO requires graph-based structure where ants construct the solution. In vehicle problems or TSP it is a graph representing the transport network, i.e. neighboring nodes are connected, in assignment problem it is a graph constructed from admissible combination of objects. What similar can we use in a meal plan problem? The model contains admissibility constraint (2). Combination of food items assigned to a particular meal must be admissible. Firstly, let us define a_{ik}:

if $a_{ik} = 1$, $i, k \in I$, then i^{th} food item can be combined with k^{th} food item

$a_{ik} = 0$, $i, k \in I$, then i^{th} food item can be combined with k^{th} food item

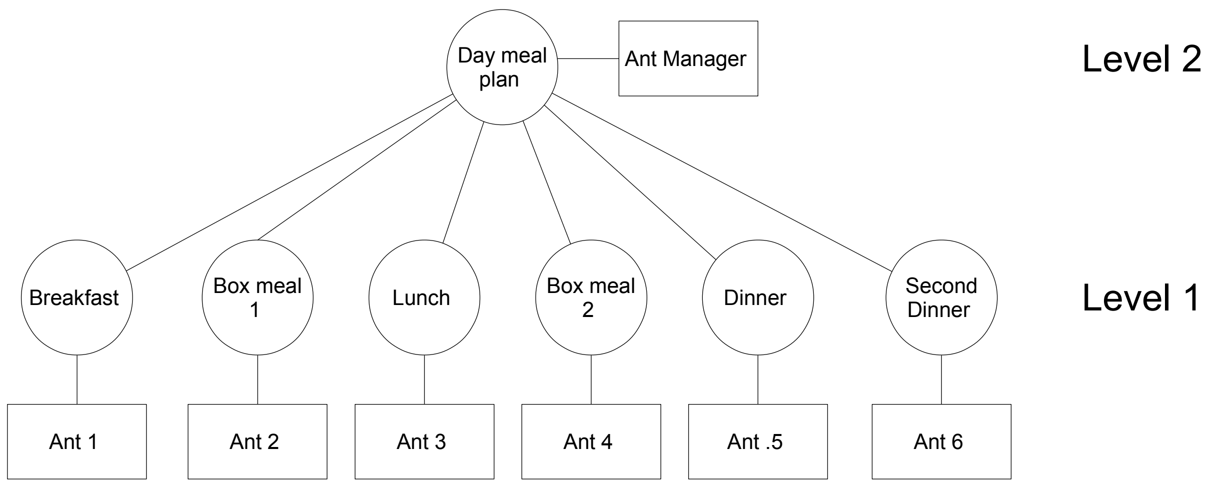

Fig. 2 HHACO for meal plan optimization problem

We can put food items together only if there is no pair among them that cannot be put together.

$$\sum_{i\in I}\sum_{k\in I} a_{ik} * Y_{ij} * Y_{kj} \geq \left(\sum_{i\in I} Y_{ij}\right)^2, \forall\, j\in J \qquad (2)$$

This constraint is to ensure that the combined food items do not cause any health problems to a patient and also form intelligently a meal that would be designed by a human. Although the constraint considers food items, we can apply it to food groups (meat, fruits, vegetables…), e.g. admissibility of milk and pastry will be the same for any two food items that belong to any of these groups. We can create a graph based on these combinations. The edges between nodes are symmetrical, i.e. if milk can be combined with pastry; it applies the other way around. In conclusion, in this graph we have nodes representing food groups and edges connecting these nodes based on admissibility of these groups.

During the day we determined 3 different types of meals that are typical for Slovakia. We consider 3 main courses: breakfast, lunch, dinner. Then we have box meals, one between breakfast (box meal 1) and lunch and one between lunch and dinner (box meal 2). Finally there is a light second dinner. Table I displays the classification.

TABLE I
ASSIGNMENT OF MEALS TO MEAL TYPES

Meal type	**Meal**
Light (snack, fruits, nuts...)	box meal 1, second dinner
Medium (pastry, hams, vegetables...)	breakfast, box meal 2
Heavy (cooked meals, soups...)	lunch, dinner*

*Dinner can be also of Medium type

For each meal type we have different set of suitable food groups as the examples by meal types imply therefore we create a particular graph for each meal type. We can call it a food group graph. In this approach there exist heterogeneous ants with their specific plans in food group graphs assigned to them based on the meal type of the meal they try to create.

To successfully construct a meal plan we need to determine ***food items and amounts*** for each meal of a day. Therefore we assign a list of food items to each node of a food group graph so when an ant visits a node it may choose a food item from it. We divide the admissible amount range for every food item into several intervals. Thus we make admissible amount range discrete and generate food item amount from these intervals. We define 3 types of the pheromone trails: for every combination of food groups, for each food item and for each food amount. At the beginning they are all set to the constant value – "0". As ants visit nodes (food groups) and they choose food items from food groups for meal plans and determine the amounts, these values increase. To place ants randomly in the particular graph, we determine a set of "starters" food groups for every meal type. For instance, at breakfast this set consists of breakfast cereals and pastry. Food items from this food groups cannot be removed from a meal list, once added. By visiting other nodes some food items are added to the chosen starter food item.

To keep the track of visited nodes and chosen food items we use pheromone updating rule. In ACO pheromone is denoted usually as τ .

Update rule for edge pheromone τ^e_{ij} (3):

$$\tau^e_{ij}(t+1) = \begin{cases} \tau^e_{ij}(t) + \Delta_{ed} \\ \tau^e_{ij}(t) \end{cases} \qquad (3)$$

The first applies for the edge from node i to j used by an ant at time t and the second applies otherwise. Δ_{ed} denotes the pheromone value change. It is set to a constant.

We define update rule for food items pheromones τ^f_j in a similar way (4):

$$\tau^f_j(t+1) = \begin{cases} \tau^f_j(t) + \Delta_{fi} \\ \tau^f_j(t) \end{cases} \qquad (4)$$

The first applies for the food item chose by an ant at time t and the second applies otherwise. Δ_{fi} denotes the pheromone value change. It is set to a constant.

Then we define pheromone updating rule also for intervals (5):

$$\tau^i_k(t+1) = \begin{cases} \tau^i_k(t) + \Delta_{\text{int}} \\ \tau^i_k(t) \end{cases} \qquad (5)$$

The first applies for the food item interval chosen by an ant at time t, the second otherwise. Δ_{int} denotes the pheromone value change. It is set to a constant.

The probability that the *k*-th ant goes from node *i* to *j* in time *t,* state transition rule, is defined by this formula (6).

$$p^k_{ij}(t) = \frac{[\tau_{ij}(t)]^\alpha}{\sum_{k\in allowed_k} [\tau_{ik}(t)]^\alpha} [\eta_{ij}]^\beta \quad \text{if } j \in \text{allowed}_k$$

$$p^k_{ij}(t) = 0 \qquad \text{otherwise} \qquad (6)$$

where $\tau_{ij}(t)$ is the pheromone amount laid on a trail between nodes *i and j*;. η_{ij} is a heuristic information constant throughout the process that can speed up the search process. For this problem we define η_{ij} as follows (7):

$$\eta_{ij} = 1 + \frac{p(j) + h(j) + s(j)}{3} \qquad (7)$$

where *p(j)* is a constant expressing patient's preference toward the food item *j,* positive value of *h(j)* expresses that the food item *j* was absent in meal plan for longer time. Constant *s(j)* expresses recommendations of foods for the specific individual made by his dietician. These three constants can be in range <-1, 1>, so using this formula we ensure that η_{ij} is in the range <0, 1>. Allowed_k is a set of nodes where an ant is allowed to go based on the admissibility constraint (2).

After choosing the next node to visit, an ant can add a food item from that group based on the food item pheromone value. Finally, it will choose a food item interval based on the food item interval pheromone value from which the exact amount is generated using the uniform probability. Then pheromone update rules are applied.

The whole algorithm is defined next in Algorithm 1.

Algorithm 1 Hierarchical Heterogeneous Ant Colony Optimization

1: {Initialize}
2: Create graphs for each type of meals;
3: Model parameters initialization;
4: {Run}
5: **For** I = 1 to Ant_Manager_Count **do**
6:   Create an ant manager with its plan and group of managed ants;
7: **End for**;
8: **Foreach** ant **in** each ant manager's group
9:   Place an ant in a starting position in the graph for its meal;
10 **End foreach;**
11: **Foreach** ant manager **in** ant manager list **do**
12: **Loop**
13: **Foreach** ant **in** ant manager's group
14: **If** the ant has not fulfilled its plan **or** (**not** End_Condition) **then**
15: Ant moves to another node based on the transition rule;
16: Ant remembers the transition;
17: **If** Ant has not visited a node **and** the transition is admissible **then**
18:   Ant adds food item from transition;
19:   Ant checks its meal plan;
20:   **If** plan not held **then**
21:   Ant applies removal;
22:   **End if**
23: **Else**
24:   Ant applies modification;
25: **End if**
26: Ant updates its position in the graph;
27: Ant updates pheromone values;
28: **End if;**
29: **End foreach;**
30: **Until** all ant managers have feasible partial solutions **or** End_Condition;
31: **End foreach;**

The other difference from the classic adaptation of ACO for TSP is that ants creating meal plans can return to the nodes once visited. In such a situation an ant uses Mutation for the already chosen food item and does not add new ones. To prevent cycling it keeps number of backward moves and stops the moving if that number equals the predefined constant. Also the moving is stopped if admission constraints prohibit it to visit any of neighbor nodes.

In the algorithm above there are two unexplained modifications of the existing solution. The first is called Modification and the second is Removal. Their purpose is to allow amount modification.

Modification (line 24 in Algorithm 1) is applied to modify the amount of a food item. It can be of two types:

- Decreasing – the amount is decreased
- Changing – the amount can increase or decrease

Modification algorithm is shown in Algorithm 2. τ_{ij}^{old} denotes the pheromone value for the interval before Modification, τ_{ij}^{new} denotes the pheromone value for the interval after Modification, Δ_1 and Δ_2 are constant values.

Algorithm 2 Modification of food item in the solution

An ant changes the food item amount;
If the new amount is in different interval than the old amount **then**

$$\tau_{ij}^{new}(t+1) = \tau_{ij}^{new}(t) + \Delta_1;$$

  If pheromone update value > Δ_2 **then**

$$\tau_{ij}^{old}(t+1) = \tau_{ij}^{old}(t) - \Delta_2$$

  Else

$$\tau_{ij}^{old}(t+1) = 0;$$

  End if;
End if;

Operation Removal (line 21 in Algorithm 1) is applied when the existing solution's constraints are exceeded, for instance energy intake is higher than recommended. Because the ACO algorithm is constructive, i.e. the more visited nodes, the more food items added to the solution, we have to check for the constraints. There has to be mechanism for removing elements (food items) from the solution. Algorithm Removal is shown in Algorithm 3.

Algorithm 3 Removal of food item from the solution

Do
  Ant applies decreasing mutation to the food item added lastly;
  If constraints are held **then**
    Return;
  End if;
  If food item's quantity is at minimum value **then**
    Remove food item from solution;
    Update pheromone values;
  End if;
While constraints are broken;

IV. Experiments

We used a food database developed by United States Department of Agriculture [17]. This database contains over 6,500 food items and presents a suitable data repository for our purpose. As a test case we refer to a bulletin issued by Ministry of Health in Slovakia [18]. The experiments were executed on the Intel Core 2 CPU, 2.1 GHz and 3 GB RAM. We consulted some model parameters with a professional from National institution of Endocrinology and Diabetes in Ľubochňa, Slovakia.

A. Test case 1

We choose nutrition requirements data from [18] for a grown-up man in the age of 35–59 years independent of the physical activity. He prefers cheese, chicken, beans and dislikes fish and sweets. Table II shows these requirements along with the obtained experimental results – mean values with standard deviation.

TABLE II
COMPARISON OF DAILY RECOMMENDED NUTRITION REQUIREMENTS AND OBTAINED RESULTS IN TEST CASE 1

Test case 1		
	Recommended daily nutrition intake	**Obtained nutrition values**
Energy [kJ]	11000 – 14500	11636.08± 502.56
Proteins [g]	64 - 72	67.37± 3.06
Carbohydrates [g]	424 - 581	453.98± 29.73
Fats [g]	75 - 95	82.08 ± 6.16
Personalization criterion	0.55 ± 0.588	
Time[s]	9.7 ± 1.5	

We see that using our approach we were able to generate meal plans that hold nutrition constraints and provide certain measure of personalization very quickly. As we created a plan for one day, we did not consider a diversity criterion that expresses menus from the historic viewpoint nor dietary recommendations. For that reason its value is always equal to “0”.

B. Test case 2

Test case 2 represents a scenario for teenage girls, aged 15 – 18. It is questionable whether we should provide meal plans for this user group; however, its stricter nutrition requirements provide a greater challenge compared to the Test case 1. She prefers cheese, chicken, beans and dislikes fish and sweets. Table III shows these requirements and the obtained results – mean values with standard deviation. The computation time has significantly increased and mean value of personalization criterion is not that high compared to the results from Test case 1, however it has still positive value. In some marginal cases nutrition constraints are not held.

TABLE III
COMPARISON OF RECOMMENDED NUTRITION REQUIREMENTS AND OBTAINED RESULTS IN TEST CASE 2

Test case 2		
	Recommended daily nutrition intake	**Obtained nutrition values**
Energy [kJ]	9600 - 11500	10050.93± 382.01
Proteins [g]	50- 55	53.17 ± 2.97
Carbohydrates [g]	378 - 453	398.47± 21.80
Fats [g]	65 - 80	70.01 ± 5.66
Personalization criterion	0.28 ± 0.623	
Time[s]	73.1 ± 633.7	

V. DISCUSSION

Firstly, let us note that in this paper we wanted to show that using our model we were able to create automatically meal plans tailored to the specific individuals. Experimental results look promising; however, there is still a great space for improvement in terms of the computation time or the solution quality. We realize many possibilities in experimental work, e.g. different pheromone updating rules in different situations or environments (food group graphs), a method to improve solution once the ants are finished, etc. It could be interesting to explore how the managing ant can direct the whole group towards good solutions, i.e. organizational influence on ants.

A difficulty that we encountered after experiments was how to compare our results to the existing approaches so we can state whether we improved the current situation or not. Meal plan evaluation is challenging. We reviewed methods assembled in [19] for this information and found out that some state only qualitative information (“Results of the simulation were satisfactory” in [20], “the feedback received on diet plan and diet menu generated by the system is acceptable” in [21]), other provide quantitative information but in the form of model parameters or sample menus [22. 23]. Aberg in [24] offered questionnaire to a test group and evaluates the quality of provided menus, which is one possible way we can go. Another possibility is to consult meal plans with a dietitian for professional evaluation. However, it is still up to a patient if (s)he accepts them.

In the future we would like to do the research about meal plans for several days. In that case we are going to propose the rule for updating coefficients related to the diversity criterion. This will also require us to extend the current ant hierarchy to capture broader time span, i.e. weeks, months, etc.

VI. CONCLUSION

In this paper we introduced a novel method based on a known metaheuristics, Ant Colony Optimization. Our contribution to this field is the extension of the traditional model by hierarchy and heterogeneity in a specific way. There exist heterogeneous ants in a hierarchy working in heterogeneous search spaces. Such model organization offers new research possibilities, mainly how such heterogeneous group can be managed in the efficient way. The model was applied in the area of nutrition and meal plans. We showed on experiments results that using our method we were able to automate such plans.

REFERENCES

[1] WHO. "STRATEGY 2004-2007 eHealth for Health-care Delivery"; http://www.who.int/eht/en/eHealth_HCD.pdf.
[2] E. Zaitseva, "Reliability analysis methods for healthcare system." in Proceedings of the 3rd Conference on Human System Interactions (HSI), 2010, pp. 211-216.
[3] WHO. "Preventing Chronic Disease," http://www.who.int/chp/chronic_disease_report/contents/part1.pdf.
[4] WHO."Obesity and overweight," http://www.who.int/mediacentre/factsheets/fs311/en/.
[5] M. Rusin, “Designing optimization model for diet recommendations,” in Digital technologies 2011, Žilina, Slovakia, 2011.

[6] C. García-Martínez, O. Cordón, and F. Herrera, “A taxonomy and an empirical analysis of multiple objective ant colony optimization algorithms for the bi-criteria TSP,” European Journal of Operational Research, vol. 180, no. 1, pp. 116-148, 2007.

[7] M. Lopez-Ibanez, and T. Stutzle, “The impact of design choices of multiobjective antcolony optimization algorithms on performance: an experimental study on the biobjective TSP,” in Proceedings of the 12th annual conference on Genetic and evolutionary computation, Portland, Oregon, USA, 2010, pp. 71-78.

[8] D. Angus, and C. Woodward, “Multiple objective ant colony optimisation,” Swarm Intelligence, vol. 3, no. 1, pp. 69-85, 2009.

[9] L. Joon-Woo, and L. Ju-Jang, "Novel Ant Colony Optimization algorithm with Path Crossover and heterogeneous ants for path planning." in Proceedings of IEEE International Conference on Industrial Technology (ICIT), 2010, pp. 559-564.

[10] Z. Peng, and L. Jie, “An Adaptive Heterogeneous Multiple Ant Colonies System,” in International Conference of Information Science and Management Engineering (ISME), 2010 2010, pp. 193-196.

[11] A. Hara, S. Matsushima, T. Ichimura et al., "Ant Colony Optimization using exploratory ants for constructing partial solutions." in IEEE Congres on Evolutionary Computation, 2010, pp. 1-7.

[12] F. Ducatelle, G. A. D. Caro, and L. M. Gambardella, “Cooperative self-organization in a heterogeneous swarm robotic system,” in Proceedings of the 12th annual conference on Genetic and evolutionary computation, Portland, Oregon, USA, 2010, pp. 87-94.

[13] C. Chira, D. Dumitrescu, and C. M. Pintea, “Heterogeneous sensitive ant model for combinatorial optimization,” in Proceedings of the 10th annual conference on Genetic and evolutionary computation, Atlanta, GA, USA, 2008, pp. 163-164.

[14] E. J. Dries, and G. L. Peterson, “Scaling ant colony optimization with hierarchical reinforcement learning partitioning,” in Proceedings of the 10th annual conference on Genetic and evolutionary computation, Atlanta, GA, USA, 2008, pp. 25-32.

[15]] J. B. Brown, and M. Huber, “Pseudo-hierarchical ant-based clustering using a heterogeneous agent hierarchy and automatic boundary formation,” in Proceedings of the 12th annual conference on Genetic and evolutionary computation, Portland, Oregon, USA, 2010, pp. 85-86.

[16] M. D. T. Stutzle, Ant Colony Optimization, Cambridge, MA: MIT Press, 2004.

[17] U. S. D. o. Agriculture, "USDA National Nutrient Database for Standard Reference Release 18," 2005.

[18] Ministry of Health of. Slovak Republic, "Bulletin," 1997.

[19] M. Rusin, and M. Zabovsky, “The review of methods for automated meal plans,” in ICTIC 2012 : proceedings in information and communication technologies - international conference, 2012, pp. 75-78.

[20] J. Bulka, A. Izworski, J. Koleszynska et al., "Automatic meal planning using artificial intelligence algorithms in computer aided diabetes therapy." pp. 393-397.

[21] N. Jothi, W. Husain, and F. Damanhoori, "MyHealthCentral.com: Integrating knowledge based system in personalized diet plan and menu construction." pp. 1-6.

[22] B. Koroušić-Seljak, “Dietary menu planning using an evolutionary method,” Electrotechnical Review, vol. 74, no. 5, pp. 285-290, 2007.

[23] H. Jen-Hao, and H. Chang, "SmartDiet: A personal diet consultant for healthy meal planning." in IEEE 23rd International Symposium on Computer-Based Medical Systems (CBMS), 2010, pp. 421-425.

[24] J. Aberg, “An evaluation of a meal planning system: ease of use and perceived usefulness,” in Proceedings of the 23rd British HCI Group Annual Conference on People and Computers: Celebrating People and Technology, Cambridge, United Kingdom, 2009, pp. 278-287.