

A Bio-inspired Approach for Dynamic Product Bundling in Enterprise Networks

Anthony Karageorgos^{1*}, Elli Rapti², Vassilis C. Gerogiannis³

¹ Department of Wood and Furniture Design and Technology, T.E.I. of Thessaly, Larissa, Greece.

² Institute of Research and Technology Thessaly, CERTH, Volos, Greece.

³ Department of Business Administration, T.E.I. of Thessaly, Larissa, Greece.

* Corresponding author. Tel.: +30 24410 64712; email: karageorgos@computer.org

Manuscript submitted August 4, 2016; accepted October 20, 2016.

doi: 10.17706/jsw.11.11.1111-1118

Abstract: Product bundling is a marketing strategy that concerns offering several products for sale as one combined product. Current bundling technology mainly focuses on the creation of static bundles, which involves pre-computing product bundles and associated discounts. However, due to the inherent dynamism and constant change of current and potential customer information, as is particularly the case in enterprise networks, static product bundles often prove to be inefficient. In this paper, we propose a bio-inspired approach for dynamic generation of personalized product bundles. The approach is based on the Ant Colony Optimization algorithm aiming to optimize the match between the available products and individual customer preferences. The applicability of the proposed approach is discussed through an exemplar application scenario.

Key words: Ant colony optimization, dynamic product bundling, enterprise networks, personalized product bundles.

1. Introduction

Bundling is a marketing strategy involving creation and offering of product or service combinations that are expected to be of interest to customers [1]. Through bundling many opportunities can be offered both to potential customers and to businesses, such as improved customer service, reduced logistics, packaging, and transaction costs, and increased sales and market share, all of which contribute to increased profitability [2]. As an example of common bundling applications in daily life are various vacation packages, software services, insurance and telecommunication packages [3]. The problem of product bundling is a challenging task since product bundles are not restricted to contain products originating from one supplier only; instead it is quite common to combine products from different suppliers whose availability and price varies dynamically, as is often the case in enterprise networks [4]. For example, it is common for shippers in logistics networks to collaborate and combine their shipment requests in order to negotiate better rates, while in the food industry businesses often combine products in bundles, such as organic cold meat and wines, in order to increase their sales.

A number of approaches have been proposed in the literature for product bundling, such as [5]-[9]. However, none of the proposed approaches achieves dynamic product bundling while taking into account individual user preferences. Therefore, to address the shortcomings of current approaches, in this paper a bio-inspired approach for dynamic generation of personalized product bundles is proposed, based on the

well-known Ant Colony Optimization (ACO) algorithm [10] and the product bundling approach introduced in [11]. In ACO, artificial ants traverse a graph in search for good solutions depositing pheromone along each trail. Following that analogy, in this paper individual products are considered to form a graph where artificial ants travel in search for product bundles, aiming to optimize the fitness of each bundle with respect to user preferences. The remainder of this paper is structured as follows. Section 2 provides the necessary problem formulation and in Section 3 the proposed bio-inspired approach for the dynamic generation of personalized product bundles based on the ACO algorithm is described. An exemplar application scenario including a brief discussion is presented as a proof of concept in Section 4. Finally, Section 5 concludes this paper and discusses future research plans.

2. Problem Formulation

Considering a traditional centralised enterprise network architecture where all enterprises store their product data in a central server, let P be a set containing all products available in the enterprise network. In accordance to ISO standards [12] each product $p \in P$ is a member of at least one product category set C . A product category set refers to a set of products having a particular common feature. For example, a sofa will commonly belong to a product category set $C_1 = \text{'sofas'}$. In addition, each product carries a set of specifications SP , where each specification $sp \in SP$ is an attribute used to describe a product. For example, if a product has white colour then a respective specification could be $sp = \text{"colour white"}$.

Each product $p \in P$ is considered to be associated with two distinct product sets, a substitute product set SUB^p and a complementary product set $COMP^p$. A substitute product set contains products that are similar to product p , including p , and could substitute p in a purchase. A substitute product $p' \in SUB^p$ must belong to at least one product category set C together with p , that is $SUB^p = \{p' | p' \in P, \exists C \in \mathbb{C}, p \in C, p' \in C\}$ where \mathbb{C} is the set of all possible product categories. A complementary product set $COMP^p$ contains products that have been bought together with product p in a certain frequency. That is $COMP^p = \{p'' | p'' \in P, \exists T \in \mathbb{T}, p \in T, p'' \in T\}$, where T is a transaction defined as a set of product items that were bought together at a particular time and \mathbb{T} is the set of all transactions. For each product $p' \in SUB^p$ a substitution degree $sim(p', p)$ can be calculated, which is defined as a function reflecting the degree to which product p' is similar to and can substitute product p , that is $sim(p', p): P \times P \rightarrow [0,1]$. The substitution degree can be calculated using a similarity measure on the set of specifications $SP^{p'}$ and SP^p of p' and p respectively, such as cosine similarity [13]. For each product $p'' \in COMP^p$ a complementarity degree, denoted as $com(p'', p)$, can be calculated, which refers to the degree to which a complementary product p'' complements a product p in user transactions, that is $com(p'', p): P \times P \rightarrow [0,1]$. Complementary products and their respective complementarity degree can be extracted using Apriori algorithm [14] for itemsets of two.

Product bundling is considered as initiating when a user U inquires information for a product $p^{pr} \in P$, termed 'primary' product, in an interaction between the user and the system. Such a user-system interaction for a particular product $p_1^{pr} \in P$ is defined as the time between the initial user information request about product p_1^{pr} and a subsequent user information request about a different product $p_2^{pr} \in P$, or a user disconnection from the system. A product bundle B is then defined as a pair (p', CP) comprising a substitute product $p' \in P$ and a set of complementary products $CP = \{p'' | p'' \in P\}$.

For any individual product included in a product bundle the product correlation index cor can be calculated, which is a function calculating the degree to which a user U is expected to prefer a product p included in the bundle. The product correlation index is calculated according to the following formula:

$$cor(p) = \frac{support(p)}{\max(support(p) | \forall p \in P)} \quad (1)$$

where, $support(p)$ is a function calculating the number of user navigations containing product $p \in B$, that is $support: P \rightarrow \mathbb{N}$, and $max(support(p)|\forall p \in P)$ is the maximum number of user navigations in which user U inquired information about a product $p \in P$. A user navigation NAV is defined as a pair $\langle ID, BS \rangle$, where ID is a unique identifier representing a specific user session, that is the period from user login to user logout, and $BS = \{p|p \in P\}$ is a set of products for which the user inquired information in that session.

For each product bundle B the bundle *affinity* can further be calculated, which is a function calculating the degree to which user U is expected to prefer product bundle B . That is $affinity: B \rightarrow [0, 1]$, and is calculated according to the following formula:

$$affinity(B) = compatibility(B) \frac{max(cor(p)|\forall p \in B)}{q} \quad (2)$$

where, $compatibility(B)$ is function calculating the number of user navigations containing all individual products included in product bundle B , and q is the total number of user navigations of user U .

Finally, the suitability of bundle B for user U can be estimated according to the following formula:

$$fitness(B) = \alpha \cdot sim(p', p^{pr}) + \beta \frac{\sum_{\forall p'' \in CP} com(p'', p')}{|CP|} + \gamma \cdot affinity(B) \quad (3)$$

where, p^{pr} is the primary product for which user U inquired information, α, β and γ are parameters in the range of $[0, 1]$, with $\alpha + \beta + \gamma = 1$, representing the degree of influence of each individual factor in the bundle's fitness, $|CP|$ is the total number of products included in CP and $fitness(B) \in [0, 1]$. The higher the value of the fitness, the more suitable the bundle is for user U . Given the above formalization, the problem is to find product bundles that maximise the fitness for a given primary product p^{pr} .

3. Ant Colony Optimization for Dynamic Product Bundling

3.1. Ant Colony Optimization

The Ant Colony Optimization (ACO) algorithm was originally introduced in [10] for the solution of hard combinatorial optimization problems. ACO is a probabilistic technique for solving computational problems which can be reduced to finding good paths through graphs without always guaranteeing convergence to a globally optimal solution. ACO was inspired by the foraging behavior of real ants. When searching for food, ants initially explore the area surrounding their nest in a random manner. When an ant finds food it returns to its nest depositing pheromone on the trail. Other ants which come across such a trail will follow the trail in search for food instead of wandering randomly. Eventually, all the ants in search for food will take the shortest path [15].

3.2. ACO-Inspired Dynamic Product Bundling

Taking inspiration from the ACO algorithm described above, it is assumed that products in P construct a weighted graph $G = (P, E)$, with n nodes and m edges. Each product $p \in P$ represents a node in the graph and each edge $e(i, j) \in E$ between two products p_i and p_j ($p_i, p_j \in P, i, j \in \mathbb{N}$) represents a link reflecting a complementarity relation among the respective products. The weight w_{ij} of an edge $e(i, j) \in E$ represents the complementarity degree between the respective products, that is $w_{ij} = com(p_j, p_i)$. If the complementarity degree between two products in P is below a certain threshold $th \in [0, 1]$, then no edge is considered to link the respective products. Products with which a product p_i is directly linked in graph G are considered its neighborhood $N(p_i) = \{p_j | p_j \in P, \exists e(i, j) \in E\}$.

Product bundling initiates when a user U inquires information about a product $p \in P$, which is subsequently considered as the primary product p^{pr} for this particular user-system interaction. According

to primary product p^{pr} , the set of substitute products of p^{pr} , $SUB^{p^{pr}}$, is the set of products that are considered as initiating nodes for the ants. To build bundles, a number of k ants initiate from each substitute product node $p' \in SUB^{p^{pr}}$ on graph G . Each ant is able to travel the graph along the edges connecting the products in P , in search for the most suitable complementary products to form a bundle with, depositing pheromone on each edge traversed. Finally, from the k bundles generated for each substitute product in $SUB^{p^{pr}}$, a number of $S \in \mathbb{N}$ bundles with the highest fitness are presented to the user who performs the final bundle selection.

3.2.1. Bundle generation

Considering node $p_i \in P$, each ant calculates the probability to move towards each neighboring node $p_j \in N(p_i)$ ($i, j \in \mathbb{N}, i \neq j$) according to the following formula:

$$prob_{ij} = \frac{\tau_{ij}(t)^{\delta} w_{ij}^{\lambda}}{\sum_{\forall p_l \in N(p_i)} \tau_{ij}(t)^{\delta} w_{il}^{\lambda}} \quad (4)$$

where, $\tau_{ij}(t)$ is the amount of pheromone on edge $e(i, j)$ on the current iteration t , w_{ij} is the weight of edge $e(i, j)$, and $\delta, \lambda \in \mathbb{N}$ are parameters representing the degree of importance of pheromone and complementarity degree in product selection respectively. For each substitute product all edges initially have $\tau_{ij}(t) = 1$. The greater the amount of pheromone $\tau_{ij}(t)$ and the higher the weight w_{ij} on edge $e(i, j)$, the higher the probability that node p_j will be selected as next node. Finally, the ant will select to move towards one of the neighboring nodes by sampling from a discrete distribution formed by the probabilities of all neighboring nodes.

When an ant moves towards a selected neighboring node $p_j \in N(p_i)$, a candidate bundle is formed including the substitute product $p_i \in SUB^{p^{pr}}$ from which the ant initiated the search and complementary products from the nodes visited so far. For the selected node p_j the ant calculates the fitness of the candidate bundle according to (3). If the fitness of the candidate bundle is higher, then product p_j is included in the bundle, that is $B = (p', CP \cup p_j)$, else p_j is not included in the bundle, that is $B = (p', CP)$. The fitness of a bundle containing only the substitute product, that is $CP = \emptyset$, is defined to be equal to 0. In both cases an ant from node p_j selects the next node from $N(p_j)$ considering the probabilities calculated according to (4). Each ant's travel through the graph is terminated either when the number of nodes visited exceeds the maximum TTL (time to live), or when there are no neighboring nodes to visit, that is $N(p_j) = \emptyset$.

3.2.2. Pheromone update

When an ant exceeds its TTL or the neighboring set of the current node is empty, it terminates its search and returns back to the initiating node depositing pheromone on each edge traversed. Inspired by [16] where the amount of pheromone deposited on each edge depends on the length of the path, the amount of pheromone deposited on edge $e(i, j) \in E$ by an ant, denoted as $\Delta\tau_{ij}$, is calculated according to formula (5):

$$\Delta\tau_{ij}^k = \begin{cases} \frac{Q}{|CP|} & \text{if ant } k \text{ has traversed edge } e_{ij} \text{ and } |CP| \neq 0 \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

where, $|CP|$ is the number of complementary products included in the bundle and $Q \in \mathbb{N}$ is a constant.

Finally, when an ant completes its search, the total amount of pheromone $\tau_{ij}(t+1)$ on each edge $e(i, j) \in E$ is updated according to formula (6):

$$\tau_{ij}(t+1) = (1 - \rho)\tau_{ij}(t) + \Delta\tau_{ij} \quad (6)$$

where, $\tau_{ij}(t + 1)$ represents the amount of pheromone on edge $e(i, j)$ on the next iteration $t + 1$ and $\rho \in [0, 1]$ is a constant representing the pheromone evaporation coefficient used to avoid convergence in locally optimal solutions.

3.3. ACO for Dynamic Bundle Generation Algorithm

Based on the above model, a simple algorithm for dynamically generating personalized product bundles is described below. The algorithm takes as input product information and user navigation history and produces as output a personalized list of product bundles.

For each substitute product the following steps are executed:

- 1) For all $p_i \in SUB^{pr}$ do
 - a) For all k ants do
 - i. $currentnode \leftarrow p_i$
 - ii. $CP \leftarrow \emptyset$
 - iii. $B \leftarrow (p_i, CP)$
 - iv. $fitness(B) \leftarrow 0$
 - v. While $hop < TTL$ and $N(currentnode) \neq \emptyset$
 1. For all $p_j \in N(currentnode)$ do
 - a. Calculate $prob_{ij}$ according to (4)
 2. Set as $currentnode$ the node selected according to $prob_{ij}$
 3. $candidate_bundle \leftarrow (p_i, CP \cup currentnode)$
 4. If $fitness(candidate_bundle) > fitness(B)$ then
 - a. $B \leftarrow candidate_bundle$
 - vi. Update pheromone according to (6)
- 2) Present S higher ranked bundles

4. Exemplar Application Scenario and Discussion

In an exemplar scenario, a number of 6 products are assumed to be stored in a central database and, considering pairwise complementarity relationships extracted from product transaction history, they form a graph as presented in Fig. 1. Considering a user U , whose navigation history is presented in Table 1, inquiring information about product $p_1 = \text{"sofa 1"}$ the bundling process begins with product $p_1 = \text{"sofa 1"}$ representing the primary product p_1^{pr} for this interaction; therefore bundles will be generated according to product $p_1^{pr} = \text{"sofa 1"}$. The substitute product set is $SUB^{p_1^{pr}} = \{p_1, p_4\}$, with $sim(p_1, p_1^{pr}) = 1$ and $sim(p_4, p_1^{pr}) = 0.61$. For demonstration purposes, the parameters considered are set as follows: $TTL = 2$, $\alpha = 0,3$, $\beta = 0,3$, $\gamma = 0,4$, $\delta = 1$, $\lambda = 1$, $Q = 0.3$ and $\rho = 0,1$. Initially, all edges have $\tau_{ij}(t) = 1$.

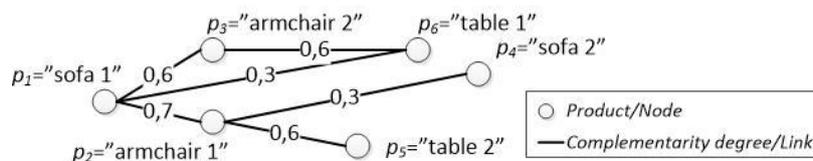


Fig. 1. Product graph with pairwise complementarity degree.

Ants travel through the graph using as initiating nodes the products included in the substitution set, that is $SUB^{p_1^{pr}} = \{p_1, p_4\}$. Considering an ant initiating its travel from node $p_1^{pr} = \text{"sofa 1"}$, the probabilities for each neighbouring node in $N(p_1^{pr}) = \{p_2, p_3, p_6\}$ are calculated according to (4). Specifically, for node

p_2 the probability is:

$$prob_{12} = \frac{\tau_{12}(t)w_{12}}{\sum_{\forall p_l \in N(p_1)} \tau_{1l}(t)w_{1l}} = \frac{1 \cdot 0.7}{1 \cdot 0.7 + 1 \cdot 0.6 + 1 \cdot 0.3} = \frac{0.7}{1.6} = 0.44$$

Accordingly, the computation of equation (4) yields the probabilities for nodes p_3 and p_6 , which are $prob_{13} = 0.38$ and $prob_{16} = 0.19$ respectively. Therefore, considering the distribution defined by the probabilities calculated, node p_2 is selected as next node towards which the ant travels. Considering node p_2 , the affinity of the candidate bundle $B = (p_1, CP = \{p_2\})$ is calculated according to (2) as follows:

$$affinity(B) = compatibility(B) \frac{\max(cor(p_1); cor(p_2))}{q} = 2 \frac{\max(1; 1)}{4} = 2 \frac{1}{4} = 0.5$$

Subsequently, the fitness of candidate bundle B is calculated according to (3) as follows:

$$fitness(B) = 0.3 sim(p_1, p_1) + 0.3 \frac{com(p_2, p_1)}{1} + 0.4 affinity(B) = 0.3 \cdot 1 + 0.3 \cdot 0.7 + 0.4 \cdot 0.5 = 0.71$$

Since bundle B before adding p_2 as a complementary product had an empty complementary product set, that is $CP = \emptyset$, its fitness was equal to 0; therefore, product p_2 is added to the bundle as it increases its fitness.

Table 1. User Navigation History with Product Support and Correlation Index

NAVIGATION ID	PRODUCT p_i					
	p_1	p_2	p_3	p_4	p_5	p_6
1	1	1	0	0	0	1
2	1	0	0	0	0	1
3	1	1	0	1	1	0
4	0	1	0	0	1	0
$support(p_i)$	3	3	0	1	2	2
$cor(p_i)$	1	1	0	1/3	2/3	2/3

From node p_2 and hop count equal to 1 ant $k = 1$, following the same process as before, calculates the probabilities for each neighboring node $N(p_2) = \{p_4, p_5\}$ according to (4). The computation of equation (4) yields the probabilities for nodes p_4 and p_5 , which are $prob_{24} = 0.33$ and $prob_{25} = 0.67$ respectively. Therefore, considering the distribution defined by the probabilities calculated, node p_5 is selected as next node towards which the ant travels. Considering node p_5 , the affinity and the fitness of the candidate bundle $B = (p_1, CP \cup p_5 = \{p_2, p_5\})$ are calculated according to (2) and (3) respectively. The computation of equation (2) yields the affinity equal to 0.25 while equation (3) yields the fitness equal to 0.43. Since adding product p_5 to the bundle lowers the fitness of the bundle, product p_5 is discarded and the final bundle is $B = (p_1, CP = \{p_2\})$.

In node p_5 and hop count equal to 2, ant has reached its TTL; therefore, it updates the pheromone of each edge traversed. The amount of pheromone deposited on each edge according to (5) is $\Delta\tau_{ij} = \frac{Q}{|CP|} = \frac{0.3}{1} = 0.3$. Since the ant traversed edges $e(1, 2)$ and $e(2, 5)$, the total amount of pheromone for $e(1, 2)$ is calculated according to (6) as follows:

$$\tau_{12}(t + 1) = (1 - 0.1)\tau_{12}(t) + \Delta\tau_{12}^1 = 0.9 \cdot 1 + 0.3 = 1.2$$

Accordingly, the computation of equation (6) yields $\tau_{25}(t + 1) = 1.2$ for edge $e(2, 5)$. All ants for each substitute product in $SUB^{p_1^{pr}}$ will sequentially follow the aforementioned process to generate a list of proposed bundles from which a number of bundles with the highest fitness will be presented to the user.

Up to our level of knowledge, this is the first approach in the literature comprising both an effective approximation method and a computation mechanism with the aim to consider user preferences when dynamically generating product bundles. A significant advantage of applying such bio-inspired approximation methods is that they are able to handle the inherent problem complexity by using a set of relatively simple rules. Therefore, when the number of products increases or product availability changes dynamically, the proposed approach is expected to outperform other centralized approaches in the literature.

5. Conclusion

In this paper a bio-inspired approach for the dynamic generation of personalized product bundles in enterprise networks is proposed based on the Ant Colony Optimization algorithm. Future plans include an extensive evaluation of the proposed approach in terms of performance and scalability by comparing the results with other approaches. In addition, as bundling in this paper is based only on product information stored in a central repository, our future work will address additional research issues arising when considering obtaining product information directly from partner distributed repositories, including semantic compatibility of product representations and performance of distributed product searches.

Acknowledgment

This research has been supported by the Greek “National Strategic Reference Framework 2007-13”, Act: “Support for SMEs groups for Research and Technological Development”, Project title: “e-Furniture: Dynamic Networking of Furniture and Wood Product Manufacturing Enterprises”.

References

- [1] Batra, V. S., Deshpande, P. M., Mohania, M. K., Nambiar, U. B., & Negi, S. (2012). Dynamic product and service bundling. USA Patent, 2012.
- [2] McCardle, K. F., Rajaram, K., & Tang, C. S. (2007). Bundling retail products: Models and analysis. *European Journal of Operational Research*, 177, 1197-1217.
- [3] Sheikhzadeh, M., & Elahi, E. (2013). Product bundling: Impacts of product heterogeneity and risk considerations. *International Journal of Production Economics*, 144, 209-222.
- [4] Azevedo, A. L., Toscano, C., Sousa, J. P., & Soares, A. L. (2004). An advanced agent-based order planning system for dynamic networked enterprises. *Production Planning and Control*, 15, 12.
- [5] Garfinkel, R., Gopal, R., Tripathi, A., & Yin, F. (2006). Design of a shopbot and recommender system for bundle purchases. *Decision Support Systems*, 42, 13.
- [6] Li, S. (2008). Method and system for flexible product and service bundling. USA Patent.
- [7] Zakas, D. L., Hayes, K. M., & Watkins, T. P. (2012). Systems and methods for bundling goods and services. USA Patent.
- [8] Sedeh, S. S., Nematbakhsh, M. A., & Mofakham, F. N. (2011). A bidding strategy in combinatorial auctions. *Proceedings of the 2011 Second International Conference on the in Intelligent Systems, Modelling and Simulation (ISMS)*.
- [9] Guo, L. R., & Xi, Z. Z. (2006). Collaborative filtering based recommendation system for product bundling. *Proceedings of ICMSE'06. 2006 International Conference on the in Management Science and Engineering*.
- [10] Dorigo, M. (1992). Optimization, learning and natural algorithms. Ph.D. Thesis, Politecnico di Milano,

Italy.

- [11] Karageorgos, A., & Rapti, E. (2013). Dynamic generation of personalized product bundles in enterprise networks. *On the Move to Meaningful Internet Systems: OTM 2013 Workshops*, 208-217.
- [12] ISO. (2006). *Industrial Automation Systems and Integration — Product data Representation and Exchange — Part 236: Application Protocol: Furniture Catalog and Interior Design*.
- [13] Lee, M. D., Pincombe, B., & Welsh, M. B. (2005). An empirical evaluation of models of text document similarity. *Cognitive Science*.
- [14] Bramer, M. (2013). Association rule mining II. *Principles of Data Mining*, 253-269.
- [15] Deng, Y., Wang, F., & Ciura, A. (2009). Ant colony optimization inspired resource discovery in P2P Grid systems. *The Journal of Supercomputing*, 49, 4-21.
- [16] Dorigo, M., Birattari, M., & Stutzle, T. (2006). Ant colony optimization. *IEEE Computational Intelligence Magazine*, 1, 28-39.



Anthony Karageorgos holds a B.Sc. in applied mathematics, an M.Sc in computer science and a PhD in agent-based software engineering. He has participated in numerous research projects in Greece and UK, which resulted in patents and several international journal and conference publications.

Furthermore, he has taught undergraduate and postgraduate courses in the areas of Information Systems and Software Engineering, and since January 2014 he is with the Department of Wood and Furniture Design and Technology, TEI of Thessaly, Greece, holding an assistant professor position on the subject of "applied informatics".

Dr. Karageorgos is an associate editor of the International Journal (ACM Transactions on Autonomous Adaptive Systems). Furthermore, he serves as a reviewer in various international journals, and he regularly participates in the Organisational or Program Committees of several international conferences. His current research interests include intelligent software, self-organising and service-oriented systems, multi-agent applications, and simulation modelling. Dr Karageorgos is a member of IEEE Computer Society and ACM.



Elli Rapti is a PhD candidate at the Dept. of Electrical and Computer Engineering, University of Thessaly, Greece. She received her diploma degree in electrical and computer engineering from the Democritus University of Thrace, Xanthi, Greece, in 2010, and her master's degree in science and technology of computers and telecommunications from the University of Thessaly, Greece, in 2014.

She is currently a research assistant at the Technological Educational Institute of Thessaly, Greece specializing in the area of distributed intelligent systems.



Vassilis C. Gerogiannis holds a diploma and a PhD in software engineering from the University of Patras, Greece.

He is an associate professor at the Business Administration Dept. of the Technological Education Institute of Thessaly, Greece. He is also an adjunct professor at the Hellenic Open University. He has published a number of papers in international journals and conference proceedings. His research interests include software engineering and project management. He is currently the coordinator of the SPRINT SMEs project, an R&D project focusing on the improvement of software development processes in Small & Medium-sized SW Enterprises. Dr. Gerogiannis has been involved as project manager in a number of EU and national R&D projects.