

SCant-Design: Closed Loop Supply Chain Design using Ant Colony Optimization

Vasco M.C. Esteves, João M.C.

Sousa, Carlos A. Silva

Center of Intelligent Systems, IDMEC,
LAETA,

Instituto Superior Técnico, U.T.L.,
Lisbon, Portugal

e_vasco@hotmail.com

Ana P. B. Povo

Centre for Management Studies of IST
Instituto Superior Técnico, U.T.L.,
Porto Salvo, Portugal

Maria I. Gomes

Department of Mathematics
Faculdade de Ciências e Tecnologia -
UNL
Caparica, Portugal

Abstract—This paper proposes a new optimization methodology for Supply Chain Design using Ant Colony Optimization. The objective of this methodology is to choose the facilities that will take part in a multi-product closed-loop supply chain, such as factories, warehouses and disassembly centers, in order to minimize the costs related to the installation and operation of these facilities and the costs related to transportation costs, both in the forward and reverse chains. Considering that total production quantities for factories, expected cross-docking stocks for warehouses, and disassembly centers are determined by this methodology, it can be considered that it undertakes both strategic and tactical Supply Chain Management (SCM) problems at once. The developed algorithm, SCant-Design, is sufficiently general to solve any SCM configuration, with linear and nonlinear cost function and constraints. The algorithm results were compared to a MILP approach for a particular case study and the obtained value for the cost function is very similar, with the advantage of using less facilities.

Keywords- Ant Colony Optimization; Optimization; Metaheuristics; Supply Chain Design; Closed-loop supply chains; Planning; Reverse logistic networks

I. INTRODUCTION

The modern vision of a supply chain (SC) shows that it is more than the flow of products in different directions. A SC can be defined as life cycle processes comprising physical, information, financial, and knowledge flows whose purpose is to satisfy end-user requirements with products and services from multiple linked suppliers. [1].

Supply Chain Management (SCM) can be defined as the systemic, strategic coordination of the traditional business functions and the tactics across these business functions within a particular company and across businesses within the supply chain, for the purposes of improving the long-term performance of the individual companies and the supply chain as a whole. It is also the coordination of production, inventory, location, and transportation among the participants in a supply chain to achieve the best mix of responsiveness and efficiency for the market being served [2] [3].

According to [4], Supply Chain Design, the strategic component of SCM, is the process of determining the supply chain infrastructure (plants, distribution centers, transportation modes and lanes, production processes, workforce and others) that will be used to satisfy customer demands. These studies use a time horizon of many months or years, and typically

assume little or no uncertainty with the data. The same author defines Supply Chain Execution as the process of determining solutions to more tactical and operational issues such as local inventory policies and deployment, manufacturing and service schedules, transportation plans, and so on. In these instances, production and transportation data are usually assumed to vary according to a known probability distribution, while the infrastructure is assumed fixed (or nearly so). The time period for the analysis typically spans days, weeks or months, and focuses on implementing detailed short-term plans.

Every entity in the supply chain will at some point return products under certain circumstances, e.g. a client being unsatisfied with a damaged product or the case where a product has reached the end of its life cycle and shall undergo a recycling process. Activities related to these matters are called reverse logistics. Among other definitions available, Fleischmann [5] in 2001 defined reverse logistics as the process of planning, implementing and controlling the efficient, effective inbound flow and storage of secondary goods and related information opposite to the traditional supply chain directions for the purpose of recovering value and proper disposal.

Several studies, e.g. [6] [7], point out that reverse distribution is not necessarily a symmetric problem of forward distribution, and that in fact, most logistics systems are not equipped to handle product movement in a reverse channel. Moreover, one must bear in mind that the roles of each character invert in those situations, so if in the forward chain there generally is a small number of suppliers and a large number of customers, in the reverse network there are many suppliers (the original customers) and few customers (that may or not be the original suppliers)[7].

II. MATHEMATICAL MODELING OF SUPPLY CHAINS

The main approach in supply chain design is to develop a mathematical representation of the supply chain and then design it through optimization, an analysis technique that determines the best possible method of designing a particular supply chain through the maximization (generally of the total system profit) or minimization (generally of the total system cost) of an objective function stated in the mathematical formulation. Most SCM problems are solved using Mixed Integer Linear Programming (MILP) or some sort of Linear or Integer Programming to which another optimization process is applied, such as Branch & Bound techniques or others. The

main disadvantage of this method is the difficult development of a sufficiently detailed mathematical model (while retaining enough simplicity to be solved). As an example, a MILP formulation was proposed by Salema, Póvoa and Novais in 2005 [7] for the design of a multi-product reverse distribution network where both forward and reverse flows are considered simultaneously. The model was applied to several cases, including the instance used in this work, which will be discussed further in the following sections.

Combinatorial optimization problems consist of analyzing a mathematical problem by searching its solution space (that usually consists of discrete variables) in order to find the lowest-cost solution. In general, problems are so complex that performing an exhaustive search to its solution space is not feasible. In such cases, other approaches must be followed. Heuristics are approximate algorithms that try to obtain near-optimal solutions at relatively low computational cost, but fail to guarantee the optimality of solutions [8]. However, these algorithms are generally unable to escape local optima. In response to this characteristic, new solution methods called metaheuristics have been proposed. Metaheuristics act as an orchestrator of interactions between local improvement procedures and higher level strategies in order to enable the capability of escaping local optima and enhancing the search of a solution space. One of these methods is the Ant Colony Optimization (ACO) metaheuristic.

ACO was inspired by the natural behavior of some species of ants, who communicate indirectly by modifying their surrounding environment using pheromones, which they lay while walking. The term stigmergy was first presented in the mid twentieth century by French entomologist Pierre-Paul Grassé, who observed the building patterns of termites and how they reacted to “significant stimuli” from pheromones deposited by previous insects. This communication mechanism results in a natural ability to optimize (reduce) the length of the path between two points of interest. In ACO, artificial ants roam freely throughout the search space while building stochastic solutions derived from probabilistic decisions based on possibly available heuristic information on the problem in question and artificial pheromones, which change dynamically at run-time to reflect the agents’ acquired search experience. In case heuristic information is used, one can interpret ACO as being an extension of traditional construction heuristics through the integration of the dynamically changing artificial pheromones [8] [9].

The use of Ant Colony Optimization techniques in Supply Chain Design is yet in an embryonic stage. The few works executed on the subject rarely address large instances of such problems, and there were no known cases involving multi-product arrangements. An example of an ACO application to Supply Chain Management, regarding mainly the tactical and operational levels, is the work by Silva et al [10], which introduced a new supply chain management technique, based on modeling a generic supply chain with suppliers, logistics and distributors as a distributed optimization problem, where each member of the SC optimizes its own performance, albeit knowing that collaboration with other partners can improve the individual and global performance. The different operational activities are solved by an ACO algorithm which allows the

exchange of information between different optimization problems by means of a pheromone matrix. A hierarchy of systems is imposed where the logistic system oversees both the production and the distribution systems.

No literature regarding SC design using ant colony optimization was found, and to best knowledge of the authors, nor an application of ACO to a SC with reverse chain was published to the date. The objective of this work is to present a first approach to the design of closed-loop supply chains using ant colony optimization, by developing an algorithm described in the next section.

A. Supply Chain Model

SCant-Design was developed as a generic algorithm for the design of closed-loop supply chains based on ACO. The SC mathematical model in which it is inspired comes from the work by Salema et al presented in [11], which should be consulted for further reference. In that paper, a mathematical model for a multi-product SC with both forward and reverse flows is developed.

Starting from sets of potential factories (including a disposal facility), warehouses, disassembly centers and clients, several parameters are considered. The disposal option contemplated above does not necessarily imply sending products to an incinerator or a landfill, since it can also model the option for external processing. This parameter is represented as a percentage of the received products by $\gamma \in [0,1]$, the minimal disposal fraction. Other parameters include total demand and return per customer and per product type, unit cost of transportation per product type between each echelon of the SC and opening costs concerning facilities. There are also variables concerning facility usage and product flows between SC constituents (regarding both demand and return).

In order to satisfy the conditions in which the SC will operate, several constraints need to be specified, namely constraints that ensure that both demand and return needs are considered. The disposal fraction is always sent to a “special factory”, which emulates for example an external company, a landfill or an incinerator. With this constraint, the amount of disposed products by each disassembly centre is known, which allows managers to assign costs and thus estimate the expenditure involved in recycling. From the total amount of returned product from each client, a percentage γ of each type of product is returned for disposal (through disassembly centers). Constraints that ensure that both warehouses and disassembly centers do not create stock are also included, since a cross-docking assumption is considered. There will also be capacity constraints, since facilities always have limited production and storage capacities (as a result of space confinement, available machinery and workforce, among other reasons).

Please take into consideration that all constraints in this problem are linear.

III. CLOSED-LOOP SC DESIGN USING ANTS: SCANT-DESIGN

The objective of this work is to design a closed-loop supply chain by selecting the location of factories, distribution centers

and disassembly centers from a set of pre-established possible locations in order to ensure that all clients' demands are satisfied. While making those choices, the developed algorithm also determines the production and storage amounts of factories, warehouses and disassembly centers, as well as which facilities serve which products to specified clients. The name of the developed algorithm comes from Supply Chain Design (SCD), from the use of Ant Colony Optimization and from the fact that the verb scant, among other interpretations, means "to limit in size or quantity", hence the acronym SCant-Design.

A. Implementation

In a closed-loop SC, end-of-life products are returned from costumers for proper disposal or recycling purposes. A schematic of a generic closed-loop SC is presented in Fig. 1. Just as a new product needs to go from factories through warehouses until it reaches its final user, end-of-life products have their own return logistics, mainly due to different handling specifications. Instead of going from clients to factories through warehouses, returning items go from clients to intermediary facilities –Disassembly Centers – in order for them to be properly sent to factories for reconditioning or remanufacturing, or to third party facilities for treatment, recycling or proper disposal.

In order to better realize the way the SCant-Design meta-heuristic was put out together, it is of interest to explain the graph on which it was idealized, presented in Fig. 2. The graph is assumed to have one echelon per type of facility (the four rows of circles from left to right), where facilities are located on the nodes (circles), and pheromone deposition will take place on the arcs that connect them (the arrows on both the forward and the reverse chain).

Notice that the cycle described in Fig. 1 is respected by the connections in Fig. 2, where a factory cannot directly serve a costumer, or a warehouse directly connect to a disassembly center. Another mentionable fact is that, alongside factories, a disposal facility is present.

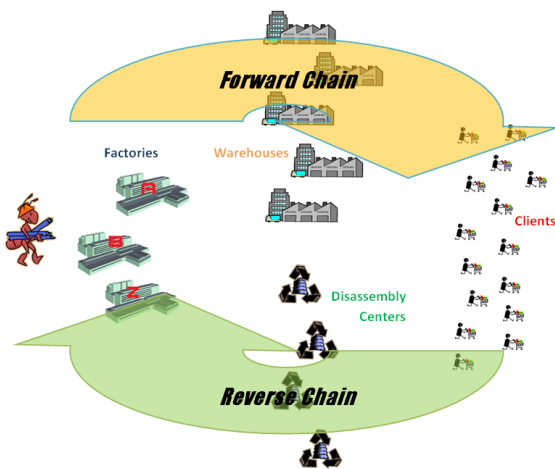


Figure 1. Representation of a Closed-Loop Supply Chain.

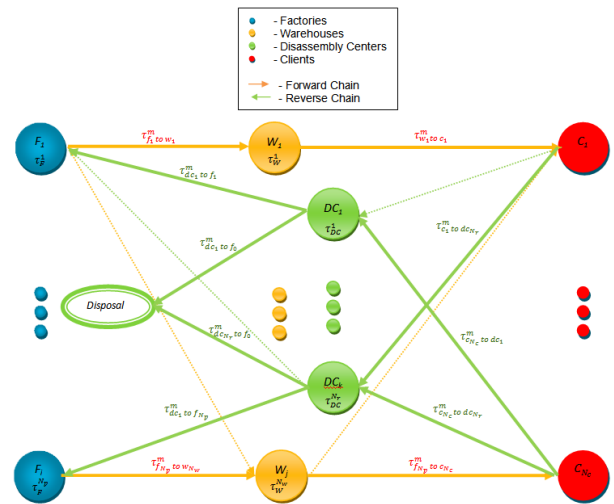


Figure 2. Implemented Graph

This facility is only present in the reverse chain and represents the possibility of sending returning items to a third party facility, which provides a service of interest to the SC such as recycling, treatment or others. The orientation of the arrows represents the way products and returning items are going to flow on the SC. On the algorithm described in this chapter, ants will move from echelon to echelon by selecting the appropriate connection.

Pheromone matrix entries on top of nodes represent Structural Pheromone Matrices (STPM), which will be thoroughly discussed in following sections. The values in these entries will represent the probability of a given facility being used (or not) in an ant-constructed solution. Pheromone matrix entries on top of connections (arrows) represent Connection-related Pheromone Matrices (CPM), and will also be thoroughly discussed next. The values in these entries will influence the probability of such connection being used to satisfy either demand (orange arrows) or return (green arrows) of costumers.

B. Developed Algorithm

SCant-Design's overall structure resembles a generic ACO algorithm. However, Algorithm III.1 represents only what is considered to be the "outer shell" of the proposed algorithm. The objective of this section is to explain the processes,

Procedure SCant-Design

```

Initialize Parameters and Pheromones
while (termination condition not met) do
    daemon choice ← Pheromone Update Daemon(iteration)
    solution ← Construct Ants Solutions
    solutionbest ← Stagnation Control Daemon
    Update Pheromones(daemon choice)
end-while
return solutionbest
end procedure

```

Algorithm III.1 – High Level pseudo-code of the developed algorithm: SCant-Design.

variables and interactions among them.

1) Initialization processes

In the Initialize Parameters and Pheromones process, parameters regarding the developed ACO heuristic are defined, such as the pheromone evaporation rate ρ , the importance of the pheromones in the ant's decisions α , the importance of the visibility of a node in the ant's decision β , and the visibility of a node (defined as the inverse of the distance between nodes) η . There will actually be four instances of the visibility matrix, one for each echelon of the SC (e.g. as from factories to warehouses).

The stopping criterion used in this algorithm is a maximum number of iterations, given by the multiplication of user-defined parameters $\text{cycle}_{\text{iterations}}$ and repetitions.

The pheromone matrices' initialization is also performed by this process. There will be three types of pheromone matrices: sequential pheromone matrices (SPM), connection-related (or *classic*) pheromone matrices (CPM) and structural pheromone matrices (STPM).

a) Sequential Pheromone Matrices (SPM)

These data structures accumulate knowledge from previous ants regarding the sequence in which costumers' demands and returns are going to be satisfied.

SPMs concerning product type sequence have a bi-dimensional structure, being the number of columns equal to the number of types of products present in the SC, while the number of rows depends on how many deliveries are necessary for ants to deliver all products.

There will be three product type SPMs, one per each cycle of delivery (from factories to clients, from clients to disposal and from clients to factories). SPMs regarding client demand satisfaction sequence, client return satisfaction sequence and disassembly center return sequence are tridimensional structures. As with product type SPM's, the number of rows depends on how many deliveries are necessary for ants to deliver all products. The number of columns will depend on the number of entities concerning the structure (e.g. the number of clients) and the depth is determined by the number of types of products present in the SC.

b) Connection-related Pheromone Matrices (CPM)

Connection-related (or classic) pheromone matrices (CPM) are pheromone data structures analogous to those used in general ACO algorithms. Each entry in these matrices refers to a connection between nodes (the arrows in Fig. 1). Like with visibility matrices η_{ij} , there will be four CPM matrices, one for each echelon of the SC. These matrices have a tridimensional structure, where the number of rows and columns are dictated by the number of entities in question (e.g. the number of possible factories and the number of possible warehouses) and the depth by the number of products used in the Supply Chain.

c) Structural Pheromone Matrices (STPM)

These matrices relate to the probability of a certain facility being used by an ant while constructing a solution. In the *Facilities Selection* sub-process presented in Algorithm III.2, the ant probabilistically chooses the facilities it will use while it constructs its solution, according to the values present in STPM

Process Construct Ants Solutions

```

for (ant=1:number ants) do
  Initialize Forward Parameters
  Transition Probability
  Facilities Selection
  while (costumer demand still unsatisfied) do
    Node Selection Forward
    Delivery Lot Size Forward
    Facility and Product Tabu Control Forward
    Write Path
  end-while
  Initialize Reverse Parameters
  while (disposals from costumers to factories not fulfilled) do
    Node selection Reverse
    Return Lot Size
    Facility and Product Tabu Control Reverse
    Write Path
  end-while
  Initialize Reverse B Parameters
  while (remaining returns from costumers to factories not fulfilled) do
    Node selection Reverse
    Return Lot Size
    Facility and Product Tabu Control Reverse
    Write Path
  end-while
  Trail Length
  solution ← Compare Colony Ants
  return
end-for
end process

```

Algorithm III.2 – Pseudo-code for the Construct Ants Solutions Process

(located on the nodes of Fig. 2). There is one STPM per type of facility. For example, ants choose the factories they will use considering the total demand of products by clients and the maximum production capacity for factories, choosing at least the minimum amount of factories necessary to answer for the total demand.

2) Construct Ant Solutions Process

In the *Construct Ants Solutions* procedure, ants incrementally build solutions for the optimization problem. This procedure is actually the core of the algorithm, where a generalized ACO algorithm is adapted to the specifications of Supply Chain Design, and an overview of it is present in Algorithm III.2, followed by a brief description of relevant sub-processes involved.

This algorithm has two stages. First all clients' demand is served (the first while loop). The next two sub-stages determine the way that end-of-life products are processed, either being disposed off or sent back to factories for reprocessing (the second and third while loops, correspondingly).

a) Transition Probability sub-processes

Transition Probability is a sub-process where the probability of an ant choosing a certain connection is calculated. The equation that computes the transition probability between factories and warehouses is shown as an example:

$$p_{ij}^m = \frac{[\tau_{ij}^m]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{i \in \text{non-tabu factories}} \sum_{j \in \text{non-tabu warehouses}} [\tau_{ij}^m]^\alpha \cdot [\eta_{ij}]^\beta} \quad \text{III.1}$$

Since the $\tau_{\text{factory to warehouse}}^{\text{product type}}$ matrix is a CPM, and consequently is tridimensional, so will matrix

$p_{factory\ to\ warehouse}^{product\ type}$ assume a tridimensional shape, again with the depth being determined by the number of products. A similar approach is used to compute all other transition probabilities, namely the ones regarding the transition probability between warehouses and clients, clients and disassembly centers and also disassembly centers and factories.

b) Node Selection sub-process

In all *Node Selection* sub-processes, the ant initially chooses the type of product it will deliver, as long as that product is not tabu (i.e. demand has not been satisfied), according to the number of the delivery it is doing (in the delivery sequence) and the product delivery sequence pheromone matrix previously introduced. Considering *Node Selection forward* as an example, the next step for the ant is to choose which client it will serve, taking into account if the client's order for that type of product has been served. This is done taking into consideration both the clients tabu list and the relevant client demand satisfaction SPM's delivery row. After choosing which client to serve, the ant selects the warehouse that will supply such customer. Such selection is made taking into account the warehouse tabu list and the previously calculated client to warehouse transition probabilities. The variable that controls the number of times that warehouses are used is also updated. In the end, the ant will choose the factory that will produce those items taking into consideration both the factory tabu list and the previously calculated warehouse to factory transition probabilities. The variable that controls the number of times that factories are used is also updated.

The behavior described above is repeated until all customers' demand has been satisfied. *Node Selection reverse* proceeds in a similar manner, but considering the reverse SC.

c) Delivery Lot size Forward sub-process

This simple sub-process analyzes the client's demand for the chosen product and compares it to both the remaining storage capacity of the warehouse it will go through and the remaining production capacity of the selected factory. The smallest of these will determine the size of the delivered lot, so that no capacity restrictions are violated.

d) Trail Length sub-process

The *Trail Length* sub-process is responsible for evaluating the total cost inherent to the solution obtained by the ant. This process starts by analyzing the number of facilities (factories, warehouses and disassembly centers) used by the ant to construct the solution, through inspection of the variables that account for flows between all facilities and costumers. Then, it analyzes the paths the ant constructed, and considering the type and amounts of products, distances between nodes and associated costs, it calculates the transportation costs for both the forward and the reverse chains. The total cost of the supply chain is then trivially computed, according to equation (1)

3) Pheromone Update Daemon Process

The *Pheromone Update Daemon* process decides which solution will be used to update the pheromone matrices, either the iteration-best or the best-so-far solution. This is achieved by gradually increasing the frequency in which the best-so-far tour is chosen for the trail update.

4) Stagnation Control Daemon process

One of the most important processes in the developed algorithm, regarding escape from local optima, is the *Stagnation Control Daemon* process. This process keeps a record concerning the last few iterations and checks for stagnation behavior. If such behavior is detected (i.e. the best solution so far has not changed in a predetermined number of consecutive iterations) this process is responsible for a sequential re-initialization of pheromone data.

The fact that pheromone matrix structures are sequentially re-initialized is due to the need to keep some previously gathered information about the problem, instead of re-initializing all pheromone matrices, which in practice would be the same as starting the optimization process from scratch. Other ACO metaheuristics use this type of search space exploration methods, such as *MMAS* [8]. According to Dorigo [8], experience has shown that pheromone trail re-initialization, when combined with appropriate choices for the pheromone trail update, can be very useful to refocus the search on a different search space region. When such methods were applied, considerable improvement over only using iteration-best or best-so-far update was observed.

In some runs of the algorithm (by user choice), another effective measure to combat stagnation behavior is to augment the number of individuals in the ant colony. This is especially effective when the algorithm is started with a lower number of ants per ant colony. One of the benefits of adopting this measure is the fact that similar, if not better, results are created in shorter runtimes as if compared with running an algorithm with a higher (constant) number of ants per colony. This is done simultaneously with the reset of pheromone structures.

5) Update Pheromones process

According to the update method defined in the *Pheromone Update Daemon* process, pheromone trails will be modified by the *Update Pheromones* process. One thing that is common to all pheromone structures is that they are all subject to evaporation before any pheromone is laid on a path. The pheromone deposition stage is where differentiation between structures occurs.

a) Update of Sequential Pheromone Matrices

Three product type sequence pheromone matrices are to be updated, one concerning the forward chain, and two regarding the two stages of the reverse chain, being the process the same for all of them. A quantity inversely proportional to the total cost resultant from the ant's path is added to the product it used in the corresponding delivery, according to the sequence in which they were delivered described by *ant path*.

Pheromone matrices regarding the sequences in which clients are served and disassembly centers return products are updated in similar fashion, but take into consideration both the client and the product that was served in a certain delivery. A quantity inversely proportional to the total cost resultant from the ant's path is added to the matrix entry that corresponds to the row of a given delivery, to the column of a certain entity and the depth that corresponds to the type of product delivered.

After the pheromone deposition is terminated, all entries in the matrices are checked. If an entry has a value higher than the

one with which the pheromone was initiated, it will be decreased to that initial value. However, there are no minimum bounds for those values.

b) Update of Connection-related Pheromone Matrices

Let us consider as an example the update of factory-to-warehouse classic pheromone matrices. While the flow between a certain factory and a warehouse tells us how much a product was manufactured in a factory and was distributed through that warehouse, it is also of interest to keep track of how many times a certain connection between these same facilities was used. If a connection had a certain flow of product pass through it in only one delivery, and another connection had that same amount of product pass through in two or more passes, than probably the second connection is of additional interest to more clients than the first. In accordance, the second connection should be *awarded* with a higher value of pheromone deposit. The idea behind this is that, if a connection was used more often, it is of higher importance to more elements of the supply chain. This also aims at reducing the importance of error made by an ant when delivering a big amount of products in just one delivery using a sub-optimal connection.

Taking into consideration the above mentioned, how much pheromone will be deposited in this connection will be a result of both the amount of flow passing through a connection and the number of times this connection was used. This will weigh a quantity inversely proportional to the total cost resultant from the ant's path and other user-defined parameters.

c) Structural Pheromone Matrices Update

Structural Pheromone Matrices are updated with a quantity inversely proportional to the total cost resultant from the ant's path, in a comparable fashion to what happens to SPMs. Remember that STPMs have a bi-dimensional configuration where the number of rows is determined by the number of possible facilities and has two columns, that store data concerning pheromones related to the previous use of such entities. If a given factory / warehouse / disassembly center is used by an ant while constructing the solution used to update pheromones, a pheromone quantity inversely proportional to the total cost resultant from the ant's path is deposited in the column that will make the selection of such facility more desirable to future ants. If, on the contrary, the facility was not used, the pheromone deposition takes place on the column that will make the selection of such facility less desirable to future ants.

IV. CASE STUDY – PROBLEM INSTANCE

The case study that will be analyzed consists of a supply chain formed by six possible locations for factories, ten possible locations for warehouses, ten possible locations for disassembly centers, fifty costumers and three types of products, as described in detail in Table I.

Customers are considered as being grouped into 50 clusters located in the same geographical area (for simplicity these clusters will be referred to as customers). It is assumed that each customer has a known demand for every type of product and each customer will return a certain amount of end-of-life-cycle products of every kind of product.

TABLE I. DATA FOR THE SUPPLY CHAIN FROM THE CASE STUDY.

Description	Parameter	Values
Fixed cost per factory [m.u.]	f_i^p	9000000
Fixed cost per warehouse [m.u.]	f_j^w	1000000
Fixed cost per disassembly center [m.u.]	f_l^r	1000000
Disposal fraction	γ	0,1
Transportation cost per Km per product [m.u.]		Product type (m)
		1 2 3
From factory to warehouse	c_{mij}^{f1}	5,72 4,93 4,78
From warehouse to customer	c_{mjk}^{r2}	7,3 6,94 5,93
From customer to disassembly center	c_{mkl}^{r1}	7,19 5,48 5,95
From disassembly center to factory	c_{mli}^{r2}	7,9 7,77 6
Description		Units
Maximum factory production	g_i^p	50000
Minimum factory production	t_i^p	5000
Maximum warehouse capacity	g_j^w	50000
Minimum warehouse capacity	t_j^w	5000
Maximum disassembly center capacity	g_l^r	50000
Minimum disassembly center capacity	t_l^r	5000

Considering these variables, the objective function to minimize, the total SC cost, is defined as:

$$\begin{aligned}
 Total_{SC\ cost} = & \sum_{i \in I} f_i^p Y_i^p + \sum_{j \in J} f_j^w Y_j^w + \sum_{l \in L} f_l^r Y_l^r + \sum_{m \in M} \sum_{i \in I} \sum_{j \in J} c_{mij}^{f1} X_{mij}^{f1} \\
 & + \sum_{m \in M} \sum_{j \in J} \sum_{k \in K} c_{mjk}^{r2} X_{mjk}^{r2} + \sum_{m \in M} \sum_{k \in K} \sum_{l \in L} c_{mkl}^{r1} X_{mkl}^{r1} + \sum_{m \in M} \sum_{k \in K} \sum_{l \in L} c_{mli}^{r2} X_{mli}^{r2}
 \end{aligned} \quad (1)$$

The first three terms of the total SC cost function (1) correspond to the fixed cost for opening the facilities (factories, warehouses and disassembly centers, respectively), the fourth term defines the total transportation cost of the products between each pair factory/warehouse, the fifth term sets the total transportation cost from warehouses to customers, and, the last two terms refer to the reverse costs are, namely, the total transportation cost from the customer to the disassembly centre and the total transportation cost from the disassembly centre to the factory (which includes the disposal option). Please note that this objective function does not have nonlinear terms.

V. CASE STUDY RESULTS

This section presents the results obtained from the application of the SCant-Design algorithm to the instance of a closed-loop supply chain described above and in [7]. The following sections describe the results obtained and a sensitivity analysis to relevant parameters of the algorithm.

A. Benchmark results

The results of the proposed methodology are compared to the results of a Mixed Integer Linear Programming (MILP) formulation presented in [11], regarding the SC in question. The proposed MILP formulation solved this instance to a Total Supply Chain Cost of $84\ 842 \times 10^3$ m.u.. Flows regarding some of the facilities are presented in Fig. 3, which describes forward flows.

From Fig. 3, it is possible to observe that factories 1, 4 and 6 are never used, as well as warehouses 4, 7 and 8. It is also clear that not all open factories (factories 2, 3 and 5) produce all products. Only factories 2 and 5 produce product 1, while factory 3 does not produce product 1 at all.

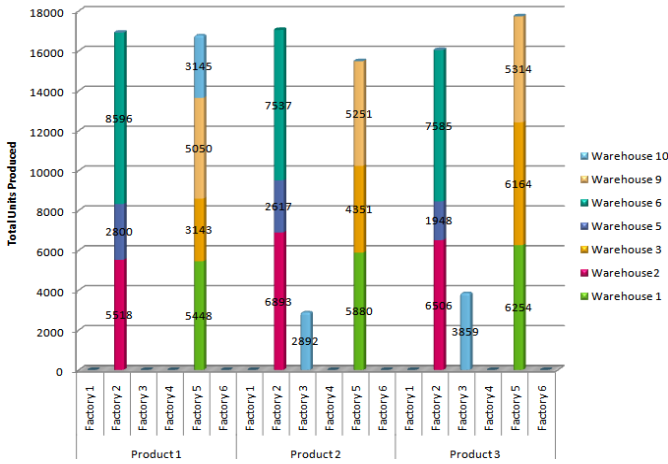


Figure 3. Flows from Factories to Warehouses (Benchmark Results)

Flows between warehouses and clients and between clients and disassembly centers are not presentable in a similar fashion, since the 50 clusters of customers versus the 10 possible locations for both warehouses and disassembly centers make it difficult to achieve graphical treatability.

There are 3 factories, 6 open warehouses and 4 disassembly centers opened in this solution.

1) SCant-Design Results

Preliminary tests and knowledge from available literature lead to the definition of parameters α and β to values of 1 and 2 respectively, values that were used in all the tests presented in this section.

All tests were done in laptop with a Core2Duo T7300, 2.00GHz processor and 2GB of RAM. The Matlab R2009b numerical computing environment was run on a Windows 7 O.S.

a) Best Result

The best solution found by the algorithm corresponded to a Total Supply Chain Cost of $85\,127 \times 10^3 m.u.$, a result that differs 0,33% from the target value of $84\,842 \times 10^3 m.u.$ found by the MILP formulation presented in [11].

From inspection of Fig. 4, it is clear that this solution does not use warehouse 10, even if all of the other warehouses present in the Benchmark Solution are open in this solution as well. The fact that one less warehouse is used may be considered as a desirable feature for the Supply Chain, since in case of demand fluctuation, storage capacity will still be guaranteed (notice that no warehouse' use is near its maximum capacity). In addition, this will mean less maintenance costs concerning SC facilities over its lifespan.

The main difference regarding the reverse flow of products resides in the role of factory 3, which now produces all types of products. This may provide the Supply Chain for extra responsiveness in case of demand fluctuations.

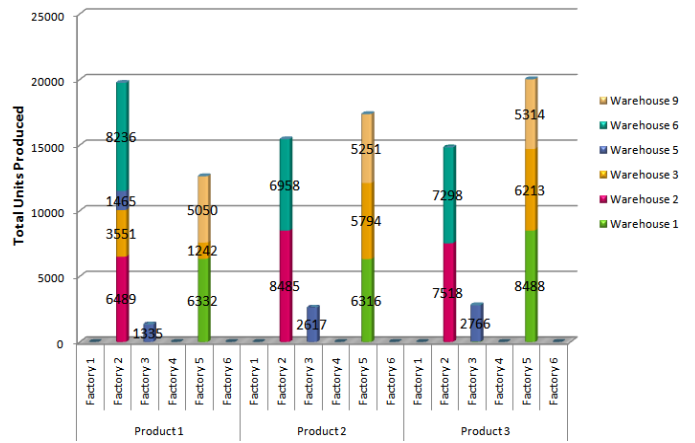


Figure 4. Flows from Factories to Warehouses (Best Result)

While the effect of re-initializing SPMs might not appear as important for improvement of the search space, a detailed inspection of the evolution of the algorithm solution's cost, shown in Fig. 6, clearly shows an improvement after the ants have re-adapted to the search space between iterations 11 000 and 13 000. This improvement is due to the re-initialization of the SPMs, which occurs around iteration 11 000. In the same figure, the re-initialization of STPMs (around iteration 9 500) and CPMs (around iteration 14 300) also shows that the re-adaptation of ant colonies to the new search space results in improved solutions.

B. Non-linear transport costs

According to [12], [13] and [14], the unit cost per distance of transporting cargo decreases with the total length of the route chosen for the transportation. Taking this into account, a set of theoretical non-linear transportation cost was conceived taking into consideration both the pre-assumed linear costs and the maximum distance between the various echelons of the SC. An example of the evolution of this cost in comparison with a linear cost is presented in Fig.6.

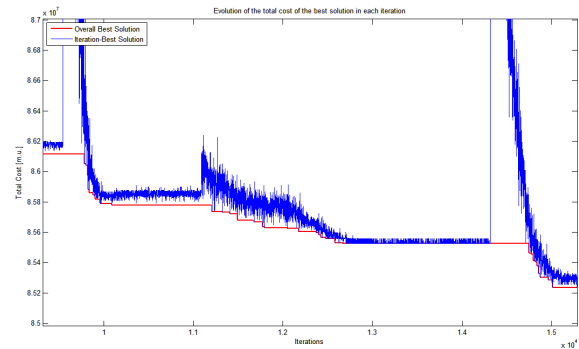


Figure 5. Detail regarding the effect of re-initializing the different Pheromone Structures

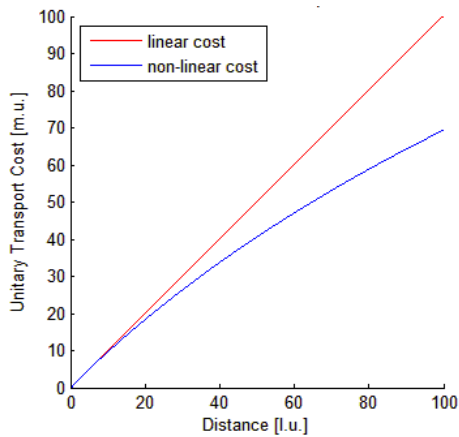


Figure 6. Linear vs non-linear transport costs

The conceptual non-linear costs were then applied to the algorithm using the SC structures found by both the original algorithm and the MILP formulation, as a theoretical exercise. Tests were conducted with the same algorithm, although the Facilities Selection sub-process was bypassed, since the SC structure was already defined. Results show that the SC structure found by the SCant-Design algorithm achieves lower total SC costs when the non-linear cost is considered, when compared to results using the SC structure from the MILP formulation. A total SC cost of $81\,378 \times 10^3 m.u.$ resulted from the application of non-linear costs to the SC structure derived from the MILP formulation, whereas the SCant-Design based SC structure achieved a total SC cost of $80\,824 \times 10^3 m.u.$. This represents a slight improvement of around 0.69%. These results were to be expected, since, as stated before, the SCant-Design solution uses one less warehouse than the MILP-derived solution.

VI. CONCLUSIONS

The work described in the previous sections resulted in the development of a new ACO algorithm, SCant-Design, applied to closed-loop supply chain design and management. The proposed approach achieves a very similar value for the objective function as a MILP formulation, using fewer warehouses in the forward chain.

Considering that Ant Colony Optimization was never applied to a Closed-Loop Supply Chain, the results obtained by the developed meta-heuristic can be considered as a promising starting ground. Taking into account that there were no available solutions for the problems that arose during the development of the algorithm in the known available literature, the solutions developed and the results found can be considered a success. For example, the way that pheromone structures were constructed concerning sequences in which actions are taken, paths to follow between facilities and structural decisions that change the very graph in which artificial ants move are, to the author's best knowledge, new.

Since this is a new approach to the design of closed-loop supply chains, there are many aspects open for future development of this algorithm.

Some of the areas that can be further studied are the pheromone update procedure and the pheromone re-initialization parameters. Since these are some of the most fundamental parameters for the success of any ACO algorithm (especially pheromone update), research into these subjects can further enhance the capabilities of the developed algorithm.

Another point of interest regarding future work is the parallelization of the algorithm, which could be somehow easily implemented if a partially asynchronous parallel implementation was considered, as suggested in [8]. This could lead to the improvement of the quality of solutions, as well as considerable computational time savings.

Nonlinear transportation costs should be adopted in order to better mimic the reality of industrial product transportation. Due to the potentialities of the ACO-based SCant-Design algorithm, their inclusion outcomes in better results with inexistent or negligible computational time differences.

REFERENCES

- [1] James B. Ayers, *Handbook of supply chain management*. N.W. Corporate Blvd., Boca Raton, Florida 33431: CRC Press LLC, 2001.
- [2] Michael Hugos, *Essentials of Supply Chain Management*, 2nd ed. Hoboken, New Jersey: John Wiley & Sons, Inc., 2006.
- [3] John T. Mentzer et al., "Defining Supply Chain Management," *Journal of Business Logistics*, vol. 22, 2001.
- [4] Terry P. Harrison, Hau L. Lee, and John J. Neale, *The practice of supply chain management: where theory and application converge.*: Springer Science&Business Media, Inc., 2004.
- [5] M. Fleischmann, "Quantitative Models for Reverse Logistics," in *Lecture Notes in Economics and Mathematical Systems*, vol. 501, Berlin., 2001.
- [6] Vaidyanathan Jayaraman, Raymond A. Patterson, and Erik Rolland, "The design of reverse distribution networks: Models and solution procedures," *European Journal of Operational Research*, vol. 150, no. Discrete Optimization, pp. 128–149, 2003.
- [7] MI Salema, APB Povia, and AQ Novais, "A warehouse-based design model for reverse logistics," *Journal of the Operational Research Society (advance online publication, 9 November 2005)*, pp. 1–15, November 2005.
- [8] Marco Dorigo and Thomas Stützle, *Ant Colony Optimization.*: A Bradford Book, 2004.
- [9] Fred Glover and Gary A. Kochenberger, *Handbook Of Metaheuristics*, 1st ed. New York, Boston, Dordrecht, London, Moscow, United States of America: Kluwer Academic Publishers, 2003.
- [10] C.A. Silva, J.M.C. Sousa, T.A. Runkler, and J.M.G. Sá da Costa, "Distributed supply chain management using ant colony optimization," *European Journal of Operational Research*, 2009.
- [11] MI Salema, APB Povia, and AQ Novais, "A warehouse-based design model for reverse logistics," *Journal of the Operational Research Society (advance online publication, 9 November 2005)*, pp. 1–15, November 2005.
- [12] Jean-Paul Rodrigue, Claude Comtois, and Brian Slack, *The Geography of Transport Systems*. London and New York: Routledge, Taylor & Francis Group, 2006.
- [13] Bahri Sahin, Huseyin Yilmaz, Yasin Ust, Ali Fuat Guneri, and Bahadır Gulsun, "An approach for analysing transportation costs and a case study," *European Journal of Operational Research*, vol. 193, Available online 24 October 2007.
- [14] David J. Forkenbrock, "External costs of intercity truck freight transportation," *Transportation Research Part A*, vol. 33, 1999.