A Multi Ant Colony Optimization algorithm for a Mixed Car Assembly Line

Raúl Pulido, Álvaro García-Sánchez, Francisco Javier Diego, Carlos Andrés-Romano

Technical University of Madrid. Research Group in Industrial Engineering and Logistics, Madrid, Spain, Polytechnic University of Valencia, Valencia, Spain

{raul.pulido,alvaro.garcia,javier.diego}@upm.es
candres@omp.upv.es

Abstract. This paper presents an ant colony optimization algorithm to sequence the mixed assembly lines considering the inventory and the replenishment of components. This is a NP-problem that cannot be solved to optimality by exact methods when the size of the problem growth. Groups of specialized ants are implemented to solve the different parts of the problem. This is intended to differentiate each part of the problem. Different types of pheromone structures are created to identify good car sequences, and good routes for the replenishment of components vehicle. The contribution of this paper is the collaborative approach of the ACO for the mixed assembly line and the replenishment of components and the jointly solution of the problem.

Keywords: Ant Colony Optimization, Vehicle Routing Problem, Car Sequencing

1 Introduction

There is a growing interest on solving multi-objective problems, this has led the researcher to combine algorithm and create an extension of the classical algorithms to achieve their objective [1]. This paper presents an ACO approach to deal with the sequencing of mid-size problems in a mixed car assembly line considering the inventory and the replenishment. Unfortunately, only small in-stances can be solved up to optimality; thus, the computational paradigm, Ant Colony Optimization, introduced by Dorigo et al. [2] was used.

Car sequencing: The mathematical model of the operation of the car assembly line was based upon the approach of sequence rules, where the assembly line can handle a pre-established production ratio for each option. In case that the task cannot be finished a utility worked is called to allow the task to be finished. “Instead of a detailed scheduling of work content, car sequencing considers and controls the succession of work intensive product options (e.g. sunroof, air conditioning) in order to avoid work overload'' [3]. Giard [4] presents a model which offers the option of hiring utility workers to allow the violation of spacing constraints. This problem has been tackled with greedy, local search, Ant Colony Optimization, Genetic Algorithm, and so on. Gottlieb [5] constructs the solution using the difference of two cars to decide how to construct the solution.

Solnon [6] uses two pheromone structure, the first to identifying good car sequences, and the second one to identify critical cars. This paper is the continuation of our previous work [7]. Where small instance of this problem is solve using a MIP. In that paper a compound approach is used to minimize the cost of replenishment of components keeping and a proper inventory. The formulation of the MIP is the same, and it will not be detailed here as the main objective of the problem is the heuristic formulation.

Inventory Routing Problem The replenishment and inventory control is a classical problem defined as the Inventory Routing Problem (IRP) is defined as "a starting point for studying the integration of different components of the logistics value chain, i.e. inventory management and transportation'' [8]. Traditionally; production and transportation have been dealt with separately. It is expected that savings could be obtained by coordinating production and transportation. Gendreau [9] categorizes the different heuristic used to solve this problem, which is ACO, genetic algorithms, greedy, Simulated Annealing, Tabu Search, and Variable Neighborhood. Bell [10] uses multiple ant colonies to identify typically used for each vehicle.

Ant colony optimization has been applied to solve large and difficult problems. Recently, ACO has been used for the solution of constraints satisfaction problems like the Car Sequencing,

We introduce different pheromone structures for the different parts of the problem to help the ants to construct the solution.

The main motivation behind this paper is to answer the following question, would it be possible to use the ACO characteristic to solve car assembly lines problem with a compound approach, without integrating problem-dependent heuristics?

2 Problem Statement

The process to assemble a car requires to pass through several workstations. Some configurations of cars require more than a takt time at the workstation to assemble the component. Other configurations require less time. Then it is necessary to find sequences that keep the ratio in each workstation. Each time that the production ratio is above this number is considered as a rule violation. Additionally, all the components required in the workstation have to be replenished from the warehouse.

Each model has a set of characteristics, such as types of wheels and tires, radio, sunroof, car seat, etc. In every workstation, a kit of components is installed; these components can have different trim levels. The combination of components and trims gives us the characteristics. To make it clearer; in the case of radio, High trim could mean radio/MP3, and Low trim could mean radio/CD (see table 1).

We define that the car is assembled when passed through all workstations which install the different components. The present problem is an extension to the problem proposed for the ROADEF challenge in 2005, where we also consider the inventory level, and the replenishment of these components.

Table 1. Example of the classes with their components

	Radio	GPS	Sunroof	…	Comp n
Class A	Low	Low	Low	…	Low
Class B	Low	High	High	…	High
Class N	High	High	High	…	High

2.1 Definition

The problem is defined by a 8-tuple (C, O, S, h, n, r, V, T DIS) such that.

- $C = \{c_1, .., c_n\}$ is the set of cars to be produced.
- Classes the set of all different cars sharing the trim level for all components.
- $O = \{o_1, ..., o_m\}$ is the set of different components.
- A: trim levels
- J : characteristic $J \subseteq O \times A$
- $S = \{s_1, ..., s_m\}$ is the set of stations to install the different components.
- Hj : Nj at most H_j of N_j successively cars may have characteristic j.
- $r_{ij} : C \times O \rightarrow \{0, 1\}$, $r_{ij} = 1$ if in station s_i the component with the characteristic H_j is installed, $r_{ij} = 0$ otherwise.
- V : it defines a maximum number of transportation vehicles.
- $TDIS(s_i, s_j)$: defines displacement times from station S_i to station S_j.

2.2 Solution

The problem will be solved when finding the sequence that violates the minimum number of sequence rules and the routes for replenish all the components, such that the capacity and constraints are met. Each route will be attended by only one vehicle. The following notation is used to denote the change of sequences:

- a sequence is noted $\pi = (c_{i1}, c_{i2}, ..., c_{ik})$
- subset of options required by a car is defined as a class classOf $(c_i) = \{h_j \in H | r_{ij} = 1\}$
- a route is defined as a nonempty subset of stations attended by each vehicle, $Rv_i = \{s_0, s_1, ..., s_{m+1}\}$ where $s0 = s_{m+1}$ denotes the depot
- the set of all sequences that may be built is π_c
- the concatenation $\oplus$ of two sequences is the first followed by the second
- a sequence π_1 is a subsequence of another π_2 , $\pi_1 \subseteq \pi_2$, if there exist two sequences that can be concatenated to π_1 to create π_2
- τ cycle (takt) time

– the cost of the sequence π and the route R depend upon the number of violated constraints, the vehicles used and the distance traveled by each vehicle, and the amount of stock in the assembly line (see equation 1).

$$cost(\pi, R)=\sum_{o_i \in o}\sum_{\pi_k \ \pi} violation(\pi_k, O_i) \times violCost + \sum_n (travelCost(V_n) + vehiclesUsed(V_n) \times Cost) + \sum_m holdingCost(s_m) \quad (1)$$

where

$$violation(\pi_k,O_i);\ violation(\pi_k,O_i)=0 \text{ if } \sum_{cl \ \pi_k} r_{lj} \le H_j;\ 0 \text{ otherwise} \quad (2)$$

$$travelCost(V_n)=\sum_{l \in R} TDIS_{ll'} \times unitCostKm \quad (3)$$

$$vehicleUsed(V_n)=0 \text{ if } distanceTraveled(V_N)=0;\ 1 \text{ otherwise} \quad (4)$$

$$holdingCost(s_m)=\sum_{j\tau} stock_{j\tau} \times unitCostStock \quad (5)$$

A solution will be defined as the set of production sequence and the routes for the vehicles that permit replenishment of the components for the given production requirements.

2.3 Construction of Solutions

We give a detailed formulation of the construction algorithm in algorithm 1. The algorithm constructs the solution from an empty sequence and empty replenishment route. Cars are iteratively added until the sequence is complete. At every step, candidates are restricted to the ones that generated the minimum cost, it means the election is restricted only to cars create the minimum extra cost. With this set of candidates (cand) the next car is chosen using transition probability (eq. 6 or 7). Then the demand over the time is calculated (operation 16) and the replenishment route is built. We start from an empty route. We duplicate the depot the number of transportation vehicles. We keep doing this, until all the cars are sequenced, we repeat this until stop criteria, and we keep the best solution and calculate the demand over the time.

We start the solution from empty routes; we duplicate the depot a number of times equal to the maximum number of vehicle. We start to add stations from the non-attended locations (candS) between the ones that generate the minimum cost, we choose for each vehicle the one that adds the minimum cost; the probability (eq. 8) will depend on T3 and η values. Once that all the station has been attended, we decrease the number of vehicles and repeat the creation of routes, unless that the number of vehicles cannot attend all the stations on time. We should keep the best solution to lay pheromones. Finally, we repeat the entire process.

```
Algorithm 1
Initialize pheromones
Repeat
  π←empty {Start to sequence the cars}
  while |π|≤|C| do let C−π denote the set of cars of C
that are not sequenced
  cand←the minimum cost generated by {c_k∈C−π}|∀c_j∈C−π,
cost(π<c_k>)≤cost(π<c_j>)
  if ∀c_i∈cand, cost(π<c_k>)≤cost(π<c_j>) then
    for every car class cc ∈{classOf(c_i)}∈ C – π do
      T_2(cc) ←T_2(cc)+ cost(π<c_j >) – cost(π)
    end for
  end if
  Choose c_i ∈ cand with the probability p(c_i, candCar, π)
  π ←<c_i >
  end while
  keep the best sequence
  calculates the instant demand for the best sequence
  repeat
    R← empty {Routes for transportation vehicles}
    for v_n = maxVehicles to v_n =1 do
      while |R|≤|S| do
        let (R−S) denote the set of non-attended stations
        duplicate the depot a number=V_n
        the ants select a candS←the min cost generated by
{sk ∈ R – S} with p(s_i, candS, R)
      end while
      if stock at station ≤ safety stock then
        break
      end if
      decrease one replenishment vehicle v_n--
      keep the best route for the transportation vehicles
    end for
  until stop criteria
  calculate the cost
  keep the best solution an update the pheromones
until stop criteria
```

The algorithm use two types of ants, the sequencing ants and the routing ants. The sequencing ant will be complete when the ant contains a full production sequence of cars. A routing ant will be complete when all the stations are visited.

The probability to build the sequence is the one described by Solnon for combining two pheromones [6] in section 6. The following part is inspired in the approach of Baran [11]. The first colony minimizes the number of vehicles, while the second colony minimizes the inventory cost. Both colonies use independent pheromone and collaborate sharing a global best solution. This solution is used to update the pheromones.

$$\mathrm{p}(\mathrm{c}_i, \mathrm{candCar}, \pi) = \frac{[\tau_1(c_j,c_i)]^{\alpha 1}[\tau_2 classOf(c_i)]^{\alpha 2}}{\sum_{c_k \in cand}[\tau_1(c_j,c_i)]^{\alpha 1}[\tau_2 classOf(c_i)]^{\alpha 2}} \text{ if the last car of } \pi \text{ is } c_j \quad (6)$$

$$\mathrm{p}(\mathrm{c}_i, \mathrm{candCar}, \pi) = \frac{[\tau_2 classOf(c_i)]^{\alpha 2}}{\sum_{c_k \in cand}[\tau_2 classOf(c_i)]^{\alpha 2}} \text{ if } \pi \text{ is empty} \quad (7)$$

$$\mathrm{p}(\mathrm{s}_i, \mathrm{candS}, R_{v_i}) = \frac{[\tau_3(s_i,s_j)]^{\alpha 3}[\ (s_i,s_j)]^{\beta}}{\sum_{s_k \in candS}\left[[\tau_3(s_i,s_j)]^{\alpha 3}[\ (s_i,s_j)]^{\beta}\right]} \text{ if } s_j \quad \text{candS, 0 otherwise} \quad (8)$$

Where $\alpha 1, \alpha 2, \alpha 3, \beta$ are relative weights for the pheromone and heuristic values respectively.

2.4 Pheromones

The three proposed pheromone structures achieve complementary goals; the first try to identify good sequence, the second try to identify critical cars, the third try to identify vehicle routes that could deliver on time the components.

- *pheromone* τ_1. Ants lay pheromone on a couple of cars (c_i,c_j) ∈ C×C associated with the amount of pheromone τ_1(c_i,c_j). It represents the past experience of sequence car c_j after c_i. This pheromone is bounded with [Tmin, Tmax] and it is initialized in the maximum.
- *pheromone* τ_2. Ants lay pheromone on car classes cc ∈ Classes(C) and the amount of pheromone τ_2(cc) represents the past experience with the car sequence of this class without violating constraints. This pheromone is bounded with [Tmin, Tmax] and it is initialized in the minimum.
- *pheromone* τ_3. Ants lay pheromone on the path between the current location and the possible location (s_i,s_j) ∈ Station(S) and the amount of pheromone levels of τ_3T3(s_i,s_j), indicating how proficient it has been visit station j after i. This pheromone is bounded with [Tmin, Tmax] and it is initialized in the minimum.
- heuristic η(s_i s_j). the dynamic attractiveness of the arc (i,j) will be: η(s_i s_j) = 1/$stock_j$, it will be computed dynamically depending of the inverse of the stock level in each station at each time.

2.5 Pheromone Update

Each pheromone will be laid and update according to with their characteristic.

Updating Pheromone τ_1 Once every ant has construed a sequence, pheromone trials are updated decreasing in order to simulate evaporation multiplying every arch by (1−ρ1). Then the best ant deposit along their paths a trail of pheromone inversely proportional with the total cost generated by the violated constrains. If the value is lower or higher than the range, it will be adjusted to the closest bound.

Updating Pheromone τ_2 Ants lay pheromone on car classes during the construction, when no more cars can be scheduled without new constraints, some pheromone is laid in the classes of the cars that have not been scheduled. The pheromone update occurs during the construction step. Every ant adds pheromone not only the best ant. In order to simulate evaporation each class is multiplying by $(1 - \rho 2)$.

Updating Pheromone τ_3 First, local updating is conducted by reducing the amount of pheromone on all visited arcs by multiplying by $(1 - \rho 3)$. Global trail updating is performed to all the arcs included in the best route founded by one of the m ants.

3 The algorithm

The algorithm follows the ACO scheme. Where each part of the problem is modeled as the search for a best Hamiltonian path in the graph. Solution are constructed using a pheromone model, then the solutions are used to modify the pheromone values. As we use utility workers for the sequence part, all the sequences are feasible. A big enough set of transportation vehicles is defined to ensure that all the routes are feasible and capable of deliver the components when are needed.

A full solution is defined as the sequence and the route of vehicles to replenish the components. After each iteration, we obtain full feasible solution to the problem which is improved after each iteration. The decrease of the use of the vehicle is given after several iterations where vehicles select the nil route.

Specialized ants are created to different part to the problems; this is intended to differentiate paths typically used by each vehicle. First, pheromone trails are initialized, then at each cycle sequence ants construct a sequence, and each route ant creates a route; the global cost function evaluates the results, and pheromone trails are updated. The algorithm stops iterating when a maximum number of cycles were performed.

1. Pheromones trials are initialized.
2. For the maximum iteration allowed to do:
   (a) The sequence ants construct a sequence.
   (b) Calculate the demand over the time
   (c) Each route ant constructs a full route.
   (d) Each route ant constructs a full route using one less vehicle.
   (e) Evaluate the solution using the global function.
   (f) Update the pheromone trails with their respective pheromone and rules.

The vehicles departure from the depot with a load of components equal to the number of vehicles v divided by the number of stations s times the number of car produced n, always respecting the vehicle capacity (see eq. 9)

$$\frac{v \times n}{s} =\leq \text{capacityOfVehicle} \tag{9}$$

Each vehicle from the replenishment route departs from the warehouse ($s_i = 0$), visit the stations and finish in the warehouse. The transportation time includes, traveling from station s_i to station s_j and unload the components.

In order to promote the exploration of different solutions each routing ant start to explore from a different part, we multiply the probability matrix for the equation 10 as a factor of the selection of the candS in the algorithm 1 in the operation 23.

$$\frac{(\text{ord}(\text{ant}) + 1)}{\text{totalNnumberOfAnts}} \quad (10)$$

The solution is driven by three main costs, the cost to the use utility workers for overloading of the station (violation of the sequence rule), the use of the transportation vehicles, and the inventory cost of the components.

The ACO was tuned using the instance small instance where an exact solution can be found by the MIP algorithm. The ACO algorithm finds solutions with the gap of 5% in less than 1 minute, when the MIP algorithm takes more than one hour. As bigger instance was not possible to solve by exact algorithm and no public instance for the entire problem was founded to compare with the optimal. We use the car sequencing instances reported by [12], and then we create the replenishment data, such as capacity, vehicle number, and so on.

4 Conclusion

The advantage of joint decision takes more importance when the cost of the space is higher than in a low-cost facility. The production space is a limited resource; the space has to be used in activity that adds value to the product and decreases the holding space. Becoming a key factor in factories where the inventory is limited, and there is no possibility to store more than one or two hours of inventory.

This paper uses the natural cooperative behavior of the ants to obtain a solution of joint problems. The contribution of this work consists in the development of a collaborative ant colony optimization system to obtain a good-quality solution for problems that cannot be solved to optimality, and the jointly solution to the problem using ACO which as far as the knowledge of the authors has not been described throughout the literature.

Future research should focus for larger problems, combining different techniques like additional types of pheromones, ranking methods, different construction strategies for the route, such as the local exchange, or candidate list.

References

1. M. López-Ibáñez, T. Stüutzle, Automatic configuration of multi-objective aco algorithms, in: Swarm Intelligence, Springer, 2010, pp. 95–106.

2. M. Dorigo, V. Maniezzo, A. Colorni, Ant system: optimization by a colony of cooperating agents, Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on 26 (1) (1996) 29–41.
3. N. Boysen, M. Fliedner, A. Scholl, Sequencing mixed model assembly lines survey classification and model critique, European Journal of Operational Research 192 (2) (2009) 349–373.
4. V. Giard, J. Jeunet, Optimal sequencing of mixed models with sequence-dependent setups and utility workers on an assembly line, International Journal of Production Economics 123 (2) (2010) 290–300.
5. J. Gottlieb, M. Puchta, C. Solnon, A study of greedy, local search, and ant colony optimization approaches for car sequencing problems, in: Applications of evolutionary computing, Springer, 2003, pp. 246–257.
6. C. Solnon, Combining two pheromone structures for solving the car sequencing problem with ant colony optimization, European Journal of Operational Research 191 (3) (2008) 1043–1055.
7. R. Pulido, A. García-Sánchez, M. Ortega-Mier, The inventory routing problem for the mixed car model assembly line, in: International Conference on Industrial Engineering and Industrial Management., 2013.
8. A. Campbell, L. Clarke, A. Kleywegt, M. Savelsbergh, The inventory routing problem, Fleet management and logistics 1 (1998) 95–113.
9. M. Gendreau, J.-Y. Potvin, O. Braumlaysy, G. Hasle, A. Lokketangen, Metaheuristics for the vehicle routing problem and its extensions: A categorized bibliography, Springer, 2008.
10. J. E. Bell, P. R. McMullen, Ant colony optimization techniques for the vehicle routing problem, Advanced Engineering Informatics 18 (1) (2004) 41–48.
11. B. Barán, M. Schaerer, A multiobjective ant colony system for vehicle routing problem with time windows, in: IASTED International Multi-Conference on Applied Informatics, Vol. 21, 2003, pp. 97–102.
12. J.-C. Régin, J.-F. Puget, A filtering algorithm for global sequencing constraints, in: Principles and Practice of Constraint Programming-CP97, Springer, 1997, pp. 32–46.