A metaheuristic approach for the operational planning of freight intermodal transportation

D. Anghinolfi, M. Paolucci, S. Sacone, S. Siri
Department of Communications, Computer and Systems Sciences
Italian Centre of Excellence for Integrated Logistics
University of Genova - Italy

1 Introduction

The objective of the present paper is the definition of an optimization problem, named Intermodal Assignment Problem (IAP), for the operational planning of freight intermodal transportation operations. More specifically, we are considering transportation operations of fruit and vegetable products in order to meet some specific constraints (concerning requested place and time of picking up and delivering, required service level, availability of trucks and trains, and so on), and minimizing the total transportation costs.

In order to face the IAP, an Ant Colony Optimization (ACO) metaheuristic approach is proposed. Ant Colony Optimization (ACO) is a recent metaheuristic which tries to emulate the successful behaviour of real ants in cooperating to find shortest paths to food for solving combinatorial problems [5]. Real ants have an effective indirect way to communicate each other the most promising trail, and finally the optimal one, towards food: ants produce a natural essence, called pheromone, that they leave on the followed trail to food in order to mark it. The pheromone trail evaporates and disappears on the paths abandoned by ants. On the other hand, the pheromone trail can be reinforced by the passage of further ants: thus, effective (i.e., shortest) paths leading to food are finally characterized by a strong pheromone track, and they are followed by most of ants. The ACO metaheuristic was first introduced by Dorigo and colleagues [4] and since then it has been the subject of both theoretical studies and successful applications to several combinatorial optimization problems (e.g., the travelling salesman problem [4], vehicle routing problems [6] and scheduling problems [8]).

2 Problem description

The problem we are facing concerns the planning of transportation operations of fruits and vegetables, moved in special refrigerating containers, whose length can be either 20 feet or 40 feet. These transportation operations are necessary in order to satisfy a specific demand, represented by a set of orders (characterized by a given product quantity, an origin, a destination, a deadline and so on). These products have to cover a given route, as indicated in Fig. 1, which starts from an agro-centre AC where fruits and vegetables are stored in pallets and, then, are put into boxes. When the agro-products are loaded in containers onto vehicles, they must be transported to their destinations, which are the general markets GM. For this transportation operation two alternatives are possible: all the route can be covered by road vehicles or an intermodal transport by road and rail can be performed. In this latter case, road transport is needed from AC to the origin rail terminal RTo (first road route) and, then, from the destination rail terminal RTd to the GM (second road route).

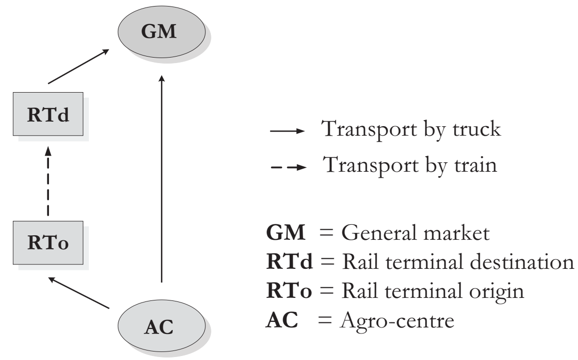

Figure 1: Layout of the logistic chain

Each road carrier owns a set of vehicles, which can be either tractors or trucks, a set of trailers and a set of flatcars (20 foot and 40 foot flatcars). Differently from road vehicles, trains present a fixed timetable, and fixed time windows in which the transhipment of containers from and on the train must be realized. Moreover, there are 5 possible different configurations of vehicles and boxes: 1) a tractor, a 20' flatcar and a 20' box, 2) a tractor, a 40' flatcar and a 20' box, 3) a tractor, a 40' flatcar and a 40' box, 4) a truck and a 20' box, 5) a truck, a trailer and two 20' boxes.

For each *vehicle* $v \in V$, the following data are considered: the corresponding road carrier $c_v \in C$, the vehicle type $vt_v \in \{0,1\} \to \{\text{tractor, truck}\}$, a set of possible configurations $I_v \subset \{1,2,3,4,5\}$, the availability at time t (state variable) $a_v(t) \in \{0,1\} \to \{\text{not available, available}\}$.

Moreover, the following quantities are associated with each *train* $r \in R$: the origin rail terminal $rto_r \in RTo$ (departure place), the destination rail terminal $rtd_r \in RTd$ (arrival place), the departure time $dt_r \in \mathbb{N}$, the arrival time $at_r \in \mathbb{N}$, the number of slots available $sl_{b,r} \in \mathbb{N},\ b \in \{20,40\}$.

For each *order* $o \in O$, the following data are considered: the origin (agro-center) $ac_o \in AC$, the destination (general market) $gm_o \in GM$, the number of pallets $np_o \in \mathbb{N}$, the deadline $dl_o \in \mathbb{N}$, the requested service level $\hat{s}_o \in \{1,2,3\}$, the set of possible road carriers for the first road route $C_o^1 \subset C^1$, the maximum time interval for products to remain in the box Δt_o^{max}.

Moreover, for each *train* $r \in R_o$ (where R_o is a set of trains compatible with order o, computed in a pre-processing analysis phase), we have to consider the following data: start time for the first road route $st_{o,r}^1 \in \mathbb{N}$, time length of the first road route $\Delta t_{o,r}^1 \in \mathbb{N}$, start time for the second road route $st_{o,r}^2 \in \mathbb{N}$, time length of the second road route $\Delta t_{o,r}^2 \in \mathbb{N}$, the set of possible road carriers for the second road route $C_{o,r}^2 \subset C^2$.

For each *road carrier* $c \in C = C^1 \cup C^2$ (respectively, the set of road carriers for the first and the second road route), the following data are considered: the provided service level $s_c \in \{1,2,3\}$, the number of flatcars available at time t (state variable) $fl_{b,c}(t) \in \mathbb{N},\ b \in \{20,40\}$, the number of trailers available at time t (state variable) $tr_c(t) \in \mathbb{N}$.

Other data of the problem concern: the fixed loading time on train ΔT_L, the fixed unloading time from train ΔT_U, the maximum number of pallets in a box $p_b,\ b \in \{20,40\}$, a matrix TT whose generic element $tt(i,j)$ represents the travelling time between node i and node j.

A *pre-processing phase* has been defined, in order to simplify the problem structure, with two main objectives: the definition of the set R_o of trains compatible with a generic order o and the determination of a different time scale, in which each time step corresponds to an assignment decision.

For the determination of the set R_o, it is necessary to consider each train and to verify three conditions: 1) whether train r is compatible with the deadline of order o or not, 2) whether the vehicle departure time with train r is non negative or not, 3) whether the route time length using train r is compatible with Δt_o^{max}. If these conditions are verified, then train r is compatible with order o and all the time instants and time intervals concerning the route associated with the couple (o, r) are determined (i.e., $st^1_{o,r},\ \Delta^1_{o,r}\ \forall o \in O,\ \forall r \in R_o;\ st^2_{o,r},\ \Delta^2_{o,r}\ \forall o \in O,\ \forall r \in R_o \setminus \{0\}$). Moreover, $r = 0$ is inserted in the set R_o, corresponding to the case in which no train is used and all the route is covered on road. More specifically, this procedure can be formalized as follows:

In order to simplify the problem structure, we suppose that decisions are taken only in some particular time instants, instead of at each time $t = 1, 2, \ldots$. More in particular, time instants in which decisions are taken are those related to the departures of vehicles ($st^1_{o,r}\ \forall o \in O,\ \forall r \in R_o;\ st^2_{o,r}\ \forall o \in O,\ \forall r \in R_o \setminus \{0\}$), also called *departure events*. These departure events make up the time set T of the considered decision problem and, for simplicity, are rewritten by means of the variable $\tau \in T$. Thus, a relation φ must be defined such that $\varphi(t) = \tau$ means that the departure event indexed by τ refers to time instant t. Moreover, the durations of the different routes ($\Delta t^1_{o,r}\ \forall o \in O,\ \forall r \in R_o;\ \Delta t^2_{o,r}\ \forall o \in O,\ \forall r \in R_o \setminus \{0\}$) are transformed into a corresponding number of departure events ($\Delta N^1_{o,r}\ \forall o \in O,\ \forall r \in R_o;\ \Delta N^2_{o,r}\ \forall o \in O,\ \forall r \in R_o \setminus \{0\}$).

3 Definition of the Intermodal Assignment Problem

Starting from the results of the pre-processing phase, an optimization problem has been formalized as a linear mixed-integer programming problem that, from now on, will be referred to as the *Intermodal Assignment Problem* (*IAP*).

The *decision variables* of the IAP are the following:

- $y^1_{o,r,v,i},\ o \in O,\ r \in R_o,\ v \in V : c_v \in C^1_o,\ i \in I_v$: binary variable equal to 1 if order o associated with train r is assigned to vehicle v in the configuration i (for the first part of the road route) and 0 otherwise;
- $y^2_{o,r,v,i},\ o \in O,\ r \in R_o \setminus \{0\},\ v \in V : c_v \in C^2_{o,r},\ i \in I_v$: binary variable equal to 1 if order o associated with train r is assigned to vehicle v in the configuration i (for the second part of the road route) and 0 otherwise;
- $x_{b,o,r},\ b \in \{20, 40\},\ o \in O,\ r \in R_o \setminus \{0\}$: integer variable corresponding to the number of boxes (20 foot or 40 foot long) assigned to order o and transported on train r;
- state variables: $a_v(\tau) \in \{0, 1\},\ v \in V$, for vehicles, $fl_{b,c}(\tau) \in \mathbb{N},\ b \in \{20, 40\},\ c \in C$, for flatcars, $tr_c(\tau) \in \mathbb{N},\ c \in C$, for trailers.

The *cost function*, given by the sum of the transportation cost for the first road route, the transportation cost for the train route and the transportation cost for the second road route, can be stated as:

$$\sum_{o\in O}\sum_{r\in R_o}\sum_{v\in V:c_v\in C^1_o}\sum_{i\in I_v}(k^1_v + p^1_{o,r,v,i})\cdot y^1_{o,r,v,i} + \sum_{b\in\{20,40\}}\sum_{o\in O}\sum_{r\in R_o\setminus\{0\}} p_{b,r}\cdot x_{b,o,r} + \\ + \sum_{o\in O}\sum_{r\in R_o\setminus\{0\}}\sum_{v\in V:c_v\in C^2_{o,r}}\sum_{i\in I_v}(k^2_v + p^2_{o,r,v,i})\cdot y^2_{o,r,v,i} \quad (1)$$

where k^1_v and k^2_v represent a fixed cost for vehicle v (for the first and the second road route, respectively), $p^1_{o,r,v,i}$ and $p^2_{o,r,v,i}$ are the transportation costs for products of order o, associated with train r, assigned to vehicle v in the configuration i (for the first and the second road route, respectively), $p_{b,r}$ is the cost for a slot of type $b \in \{20, 40\}$ on train r.

The mathematical formalization of the *constraints* of the IAP is not reported in the present paper for the sake of brevity, thus a brief description of their meanings is given in the following. Assignment and availability constraints for vehicles are considered in order to assure that a vehicle is assigned to an order only if it is available (for both road routes) and that the availability of the vehicle becomes equal to 0 if it is assigned and keeps equal to 1 otherwise. Analogous assignment and availability constraints have been considered for flatcars and trailers. Moreover, other constraints have the purpose of assuring that only vehicles whose road carriers can guarantee a sufficient service level are chosen. Other types of constraints are those assuring that a sufficient number of boxes are assigned to each order (characterized by a given number of pallets). Finally, another set of constraints is considered, assuring that, for a given order, the same number and type of boxes are assigned to road vehicles and trains.

4 A metaheuristic Ant Colony Optimization approach

A suitable Ant Colony Optimization (ACO) metaheuristic approach is proposed, in order to face the Intermodal Assignment Problem (IAP) also for large problem instances. The ACO algorithm developed for the IAP is basically inspired by the Ant Colony System (ACS) [3] and *Max-Min* Ant System (MMAS) [7] versions of the basic algorithm; the general framework of this ACO algorithm is analogous to the one described in [1], where an approach based on a new global pheromone update mechanism has been proposed. The basic structure of the implemented ACO algorithm is summarized in Fig. 2: at each iteration of the main loop, any ant constructs a solution performing an inner loop of progressive assignment decisions.

```
Initialization;
k=1;
While <termination condition not met>
  {
   For each ant a∈A
      {
       Construction of solution x_a^k;
       Local pheromone update;
      }
   Local Search phase;
   Global pheromone update;
   k=k+1;
  }
```

Figure 2: The overall ACO algorithm

An ant has a state that is updated during the construction process to dynamically check its feasibility. In particular, an ant must know the number of pallets that remains to be served for each order, the number of slots available for each train, and the partial solution (assignment of pallets to vehicle configurations and trains) constructed so far. The steps executed in the inner loop of the ACO algorithm correspond to the decision stages associated with the set T of discrete departure events. Assuming $n = |T|$, the decision stages for an ant can be represented as in Fig. 3: at each stage τ there is a square node S_τ representing the state of the ant before the decision (in the initial square node with $\tau_0 = 0$, the ant starts with an empty partial solution).

The circle nodes $h = 1, \ldots, L_\tau$, at stage τ, are associated with different sets of boxes that can be used to serve the order o up to the number of remaining pallets to send. The ant selects (according to a suitable ACO construction rule, as in [2]) which arc to follow from S_τ leading to a circle node h that will change the ant state. The circle node generation process is based on the computation of the maximum number of 20' and 40' boxes needed to completely serve the remaining pallets. A *greedy configuration heuristic* procedure is called to establish whether the selected set of boxes can be actually transported by vehicles, trailers and flatcars currently available, or if the selected node is unfeasible. This procedure tries to determine the feasible vehicle configurations to ship the boxes of the node selected by the ant, taking into account the current resource availabilities and the fixed and the transportation costs. If no finite

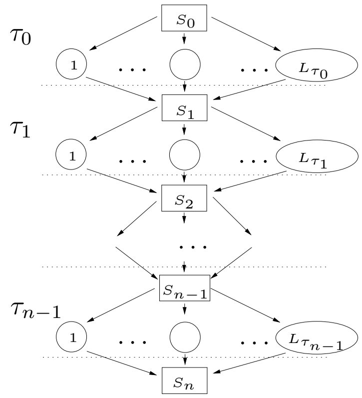

Figure 3: The solution construction stages for an ant

cost assignment is found for the selected node, the ant eliminates it and proceeds with a new selection, until a feasible assignment is determined. When the ant has considered all the decision stages, the cost of the solution is computed; if a complete feasible solution has not been found, the cost will include a penalty term.

The node selection is performed by an ant in two steps. First, the ant determines which node selection rule to use between an *exploitation* and *exploration* one, by extracting a random number from the uniform distribution $U[0, 1]$. The exploitation rule selects the node h^* in a deterministic way, whereas the exploration rule is computed according to a selection probability. The pair (τ, h) identifies a so-called solution component, i.e., a possible selection during the ant solution construction process. The quantity $\pi_k(\tau, h)$ is the pheromone trail at iteration k associated with the selection of node h at stage τ: this pheromone trail represents a measure of the appropriateness of selecting a component during the construction of good solutions, which are progressively learned from the solution space exploration.

5 Experimental results

In order to evaluate the effectiveness of the proposed ACO algorithm, a problem generator has been implemented, which randomly produces different instances depending on the values of some input parameters. Thus, several sets of different problem instances, for different problem dimensions, have been solved by applying two separate approaches: the former solution method is the application of the mathematical programming formulation of the IAP using the Cplex 9.0 MIP solver, while the latter solution approach is the ACO metaheuristic algorithm, implemented in C++.

The generated instances have been classified into 6 difficulty groups (in increasing order) according to the number of variables and constraints produced for the corresponding MIP formulation. Table 1 reports the summarizing results of the comparison between the MIP solver and the ACO solution, aggregated by instance groups. The average results of the ACO approach put into evidence that this algorithm was able to solve all the instances in quite acceptable computation times; also for instances of group 6, not solved by Cplex, the ACO was able to find solutions serving all the orders with a 236.6 s average CPU time. Moreover, the ACO algorithm performance is even more appreciable when the difficulty of the instances increases (ACO algorithm terminated with an average CPU time that is usually a fraction of the time needed by the MIP solver, finding a solution very close to the optimal one). Finally, it is important to note that the average and best ACO results are quite close and the standard deviations for the ACO average results are very small, thus implying that the ACO algorithm shows a stable behaviour.

Table 1: Summarizing results for the 6 groups of problem instances

	MIP solver	**ACO**		
Group	CPU time	Time dev. (%)	Avg ACO obj. dev. (%)	Best ACO obj. dev. (%)
1	39.23	-35.39	2.09	1.87
2	154.44	-57.15	1.61	0.88
3	345.48	-72.09	0.64	0.60
4	408.50	-58.69	1.21	1.08
5	836.44	-75.94	0.82	0.51
6	-	-	-	-
Global	356.82	-59.85	1.27	0.99

6 Conclusions

In this paper, a model and an optimization problem for the operational planning of transportation operations have been defined. For this problem, a mathematical programming formulation has been provided (linear mixed-integer programming problem), together with a metaheuristic approach based on the development of an Ant Colony Optimization (ACO) algorithm.

The proposed ACO algorithm has been tested on different randomly generated problem instances, by comparing the solutions obtained with those coming from the application of the MIP formulation. The obtained experimental results have shown the effectiveness of the proposed metaheuristic approach.

References

[1] D. Anghinolfi and M. Paolucci. A new ant colony optimization approach for the single machine total weighted tardiness scheduling problem. *accepted for publication on International Journal of Operations Research*, 2007.

[2] M. Dorigo and C. Blum. Ant colony optimization theory: A survey. *Theoretical Computer Science*, 344:243–278, 2005.

[3] M. Dorigo and L.M. Gambardella. Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Trans. on Evolutionary Computation*, 1:53–66, 1997.

[4] M. Dorigo, V. Maniezzo, and A. Colorni. Ant system: optimization by a colony of cooperating agents. *IEEE Trans. on Systems, Man, Cybernetics-Part B*, 26:29–41, 1996.

[5] M. Dorigo and T. Stützle. The ant colony optimization metaheuristics: algorithms, applications and advances. In F. Glover and G. Kochenberger, editors, *Handbooks of metaheuristics*, Vol. 57 - Int. Series in Operations Research & Management Science, pages 252–285. Kluwer, Dordrech, 2002.

[6] M. Reinmann, K. Doerner, and R.F. Hartl. D-ants: savings based ants divide and conquer the vehicle routing problems. *Computers & Operations Research*, 31:563–591, 2004.

[7] T. Stützle and H.H. Hoos. Max-min ant system. *Future Generation Computer System*, 16:889–914, 2000.

[8] G.C. Ying and C.J. Liao. Ant colony system for permutation flow-shop sequencin. *Computers & Operations Research*, 31:791–801, 2004.