

# Cooperative Mission Planning for a Class of Carrier-Vehicle Systems.

Emanuele Garone <sup>◦</sup>, Roberto Naldi <sup>†</sup>, Alessandro Casavola <sup>◦</sup> and Emilio Frazzoli <sup>‡</sup>

<sup>◦</sup> Università della Calabria, Rende (CS), 87036, Italy.

<sup>†</sup> Università di Bologna, Bologna, 40133, Italy.

<sup>‡</sup> Massachusetts Inst. of Technology, Cambridge, MA, 02139, USA.

**Abstract**—In this work we focus on mission planning problems in scenarios in which a carrier vehicle, typically slow but with virtually infinite range, and a carried vehicle, which on the contrary is typically fast but has a shorter range, are coordinated to make the faster vehicle visit a given collection of points in minimum time. In particular in this paper we will address two mission planning problems: a first one, in which we have to sequentially visit a list of points under the hypothesis the takeoff/landing sequence is not determined *a priori* and a second one, a Traveling Salesman Problem (TSP), in which the optimal visiting sequence of points has to be determined. Those two problems will be analyzed, sub-optimal heuristics will be presented and their properties pointed out.

## I. INTRODUCTION

The complexity of many applications envisioned for future autonomous vehicle networks, ranging from planetary exploration to rescue missions, requires a broad range of capabilities for individual units—ranging from air, ground or sea mobility, to sophisticated multi-modal sensor suites and actuation devices—which cannot be implemented on a single platform class. Rather, it may be necessary to coordinate several specialized units to attain complex objectives in a reliable, timely, and efficient fashion [1]. While considerable progress has been made on cooperative control of networks of homogeneous vehicles (see for example [2], [3], [4], [5]), heterogeneous networks are still relatively poorly understood. In such a direction recent developments aiming at spreading the adoption of unmanned systems in real-world operational scenarios—[6], [7]—consider the employment of cooperating mobile robots [8], often denoted as multiple mobile robot systems, combining the characteristics of heterogeneous vehicles with complementary features. To understand how to optimally exploit the different capabilities of each individual unit and obtain the desired final behavior, the team is required to be suitably coordinated through advanced planning and control algorithms. In this paper, we concentrate on a very simple system of heterogeneous vehicles, arising from the combination of (i) a slow autonomous surface carrier (typically a ship) with long operating range capabilities and (ii) a faster vehicle (typically a helicopter, a UAV, or an offshore vehicle) with a limited range. The carrier is able to transport the faster vehicle, as well as to deploy, recover, and service it. Even though this two-vehicle system is very simple, it reveals new aspects of many path

This research has been funded by the research project “Giovani Ricercatori” sponsored by the University of Calabria. Corresponding author: Roberto Naldi, email: roberto.naldi@unibo.it

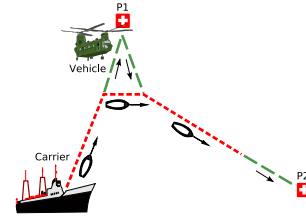


Fig. 1. The carrier-vehicle system on a rescue scenario.

planning and coordination problems of interest, including those introduced in [10], [9], [11] for groups of homogeneous robots. In the preliminary works [12], [13] the determination of the optimal trajectories connecting  $n$  has been detailed. Here we extend those results for the cases in which the number of points visited between a take-off and a landing or the visiting sequence is not given *a priori*.

## II. THE CARRIER-VEHICLE SYSTEM

The system we are going to deal with is composed of two different vehicles, a *vehicle carrier* (also denoted in the following as *carrier*), whose variables and functions will be denoted by subscript  $\cdot_c$ , and a *carried vehicle* (compactly referred to as the *vehicle*), denoted by subscript  $\cdot_v$ . In the following we will refer to the combined system as the *carrier-vehicle system*. To derive a mathematical model for the system, we will consider the vehicles as points belonging to the Euclidean space  $\mathbb{R}^2$ . Let  $p_c(t) = [x_c(t) y_c(t)]^T$ ,  $p_v(t) = [x_v(t) y_v(t)]^T$  be, respectively, the position of the carrier and of the vehicle at time  $t$ . We will assume that the position of the carrier  $p_c(t)$  evolves accordingly to the first order O.D.E.

$$\dot{x}_c = V_c \cos(\phi_c), \quad \dot{y}_c = V_c \sin(\phi_c) \quad (1)$$

with  $V_c \in \mathbb{R}^+$  the given velocity of the carrier and  $\phi_c \in \mathbb{R}$  the control input. This implies that the class of the admissible paths for the carrier are all continuous curves in the two-dimensional Euclidean space. The carrier travel on these paths with a speed bounded in magnitude by  $V_c$ . In modeling the dynamics of the carried vehicle we distinguish between two different situations:

- 1) when the vehicle is not on board the carrier it evolves following its free planar motion:

$$\dot{x}_v = V_v \cos(\phi_v), \quad \dot{y}_v = V_v \sin(\phi_v) \quad (2)$$

with  $V_v \in \mathbb{R}^+$ ,  $V_v > V_c$  and  $\phi_v \in \mathbb{R}$  the control input for the vehicle.

- 2) when it is on board the carrier, its position coincides with the carrier position,  $p_v(t) = p_c(t)$ .

From the above arguments it appears that the carried vehicle dynamics shows an intrinsically hybrid behavior. Because one of the distinguish features of the carried vehicle is to have a finite operating range (e.g., due to maximum *fuel* capacity), we assume that, after leaving the carrier deck, it can operate as a stand-alone vehicle only for a limited time  $\bar{a}$ . For the sake of simplicity, it is supposed that any time the faster vehicle comes back to the carrier its operating range is instantaneously restored.

### III. PREVIOUS RESULTS - ORDERED VISIT OF $n$ POINTS

The first step to deal with the mission planning problems described in this paper is the availability of an effective optimization procedure able to solve in a reasonable amount of time the following basic path planning problem

**Problem 1** — given an initial point  $p_0$  such that  $p_c(0) = p_v(0) = p_0$ , a desired final point  $p_f$  and a list of  $n$  points  $q_{list} = [q_1, \dots, q_n]$ , determine the minimum-time trajectory such that each point is visited by the carried vehicle in an ordered way by following, for each point  $q_i$ , a given sequence of *takeoff - visiting the new point - landing* prescriptions and finally both the carrier and the vehicle approach the point  $p_f$ .  $\square$

In [13] it has been shown that such a problem can be rephrased into a convex optimization problem that may be solved with a very low computational effort. Moreover it has been shown that, even if up to our knowledge an exact closed form formula for the optimal cost is not known in the general case (beside some special cases like those discussed in [12]), it is possible to analytically characterize an upper bound and a lower bound to the optimal solution of Problem 1. Namely, it is possible to prove that a lower bound to the optimal cost of Problem 1 is given by:

$$t_L(\ell, n) = \max \{ (\ell/V_c - nV_v\bar{a}/V_c + n\bar{a}), \ell/V_v \}. \quad (3)$$

where  $\ell$  denotes the sum of all the distances between the points of interest, i.e.,  $\ell = \sum_{i=1}^{n+1} d_{i-1,i}$ . and where  $d_{0,1} := \|p_0 - q_1\|$ ,  $d_{i-1,i} := \|q_{i-1} - q_i\|$ ,  $i \in \{2, 3, \dots, n\}$ ,  $d_{n,n+1} := \|q_n - p_f\|$ . Note that, by construction,  $\ell$  denotes the *length of the shortest path* visiting all the points of interest. To derive an upper bound it is possible to proceed as follows. Let us denote with  $d_{min} := \min_{i=1, \dots, n} d_{i-1,i}$  and let  $\bar{a}' = \min\{d_{min}/V_v, \bar{a}\}$ ; then, the following upper-bound may be obtained

$$t'_U(\ell, n, \theta_{list}) = t_L(\ell, n) + \sum_{i=1}^n \frac{\Delta'(\theta_i, \bar{a}, \bar{a}')}{V_c} \quad (4)$$

with

$$\Delta'(\theta, \bar{a}, \bar{a}') := \begin{cases} (\bar{a} - \bar{a}')V_v & \text{if } \theta \leq 2 \arcsin\left(\frac{V_c}{V_v}\right) \\ \bar{a}V_v - \bar{a}'V_c/\sin(\theta/2) & \text{else} \end{cases}, \quad (5)$$

where  $\theta_{list} := [\theta_1, \theta_2, \dots, \theta_n]$  denotes list of the  $n$  angles such that  $\theta_i \in [0, \pi]$ ,  $i = 1, \dots, n-1$ , i.e. the set of the

minimum amplitude angles formed by the segments that connect two consecutive points to be visited (for a graphical intuition see also Figure 2). Please note that if  $\bar{a}' = \bar{a}$  and if  $\theta_i \leq 2 \arcsin(V_c/V_v)$  for all  $\theta_i \in \theta_{list}$ , the proposed upper bound precisely matches the cost of the lower bound (3), and indeed it represents one of the optimal solutions. As highlighted in [13] tighter upper bound maybe given under some particular assumptions.

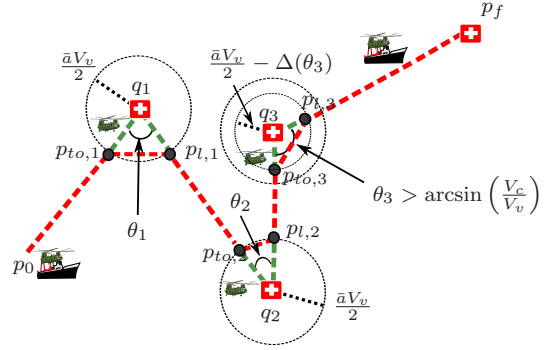


Fig. 2. The geometric interpretation behind the proposed algorithm to build a feasible solution to visit a set of  $n$  points (3 in the figure).

### IV. MISSION PLANNING

In this paper we discuss two mission planning problems for the carrier-vehicle system. By mission planning problems we mean those problems in which discrete decisions on the behavior of the vehicles are variables of the optimization problem. Typically this kind of planning problems requires mixed-integer optimization for their solution, and are in general in the class of NP-Hard problems

#### A. Ordered Visit of $n$ points without prescribed takeoff-landing sequences

In Section III we have recalled the problem of visiting an ordered sequence of points under the constraint that for each takeoff only one target is reached by the vehicle before returning to the carrier. The goal of this section is then to remove this constraint in order to address the more general scenario in which the vehicle is allowed to visit more than one point before landing back on the carrier's deck. Thus, given an initial point  $p_0 = p_v(0) = p_c(0)$  and a final point  $p_f$ , we want to determine the minimal time trajectory allowing the ordered visit of a list of  $n$  points  $q_{list} = [q_1, \dots, q_n]$  and, after the last landing, the return of both vehicles to  $p_f$ . In order to model the fact that after a takeoff the faster vehicle may visit more than one point belonging to  $q_{list}$ , let us introduce the binary variables

$$\alpha_{ij} \in \{0, 1\}, \quad i \leq j. \quad (6)$$

whose semantic is that if  $\alpha_{ij} = 1$  then all and only the points belonging to the sublist  $q_{list,i,j} = [q_i, q_{i+1}, \dots, q_{j-1}, q_j]$  will be sequentially visited by the carried vehicle without returning to the carrier. For any sublist  $q_{list,i,j}$  of targets we will also denote with  $p_{to,i}$  and  $p_{l,j}$  the corresponding takeoff and landing points. Because every point has to be visited exactly once, any point may belong to one and only one

group of points to be visited in a row. This can be modeled by the following  $n$  constraints on the  $\alpha_{i,j}$  variables:

$$\sum_{(i,j):i \leq k \leq j} \alpha_{i,j} = 1, \quad k = 1, \dots, n \quad (7)$$

Let us introduce also the time  $t_{ij}^{to,l}$  representing the elapsed time between a takeoff, at the point  $p_{to,i}$ , and a landing event, at the point  $p_{l,j}$ . During that interval of time the carried vehicle visits all and only the points belonging to the list  $q_{list,i,j}$ . By exploiting the above discussion on the binary variables  $\alpha_{i,j}$  and by using the same arguments detailed in [13], such a time can be bounded as follows

$$\begin{aligned} \alpha_{i,j} \frac{1}{V_v} \left( \|q_i - p_{to,i}\| + \sum_{k=i}^{j-1} \|q_{k+1} - q_k\| + \|q_j - p_{l,j}\| \right) &\leq t_{ij}^{to,l} \\ \alpha_{i,j} \frac{1}{V_c} (\|p_{to,i} - p_{l,j}\|) &\leq t_{ij}^{to,l} \\ \alpha_{i,j} t_{ij}^{to,l} &\leq \bar{a} \end{aligned} \quad (8)$$

$$i = 1, \dots, n, j = i, \dots, n.$$

Note that the latter constraints reduce to  $t_{ij}^{to,l} \geq 0$  if  $\alpha_{i,j} = 0$ . Similarly if we introduce the time  $t_i^{l,to}$  representing the interval of time between the landing at point  $p_{l,i-1}$  and the takeoff at point  $p_{to,i}$  after which the first point to be visited will be  $q_i$  (it is assumed  $p_{l,0} = p_0$ ) we obtain

$$\begin{aligned} \sum_{j=i}^n \alpha_{i,j} \frac{1}{V_c} \|p_{l,i-1} - p_{to,i}\| &\leq t_i^{l,to}, \quad i = 1, \dots, n \\ \frac{1}{V_c} \|p_{l,n} - p_f\| &\leq t_{n+1}^{l,to} \end{aligned} \quad (9)$$

Note that the first  $n$  constraints above reduce to  $t_i^{l,to} \geq 0$  for each value of  $i$  such that  $q_i$  is not the initial point of a sublist of targets to be visited in a row. Finally, the ordered visit of  $n$  points without prescribed takeoff/landing sequences can be formally rewritten as the following mixed-integer nonlinear programming problem:

$$\begin{aligned} \min \sum_{i=1}^n \sum_{j=i}^n t_{ij}^{to,n} + \sum_{i=1}^{n+1} t_i^{l,to} \\ \text{subject to : (6), (7), (8), (9).} \end{aligned} \quad (10)$$

Some observations on the form of the above problem may be of interest. First, note that if the binary variables  $\alpha_{i,j}, i = 1, \dots, n, j = i, \dots, n$  have been assigned, then

- the problem becomes a convex optimization problem;
- if  $\alpha_{i,j} = 0$ , then in the optimal solution of the corresponding problem it would result  $t_{ij}^{to,l} = 0$ ;
- if  $\alpha_{i,j} = 0, \forall j = i, \dots, n$ , then in the optimal solution of the corresponding problem it would result  $t_i^{l,to} = 0$ .

Moreover, if we focus on the constraints (6)-(7), it is worth noticing that the number of possible ways of partitioning a list into subsequences is equal to  $2^{n-1}$  (see [14] amongst others). However, interestingly enough, in most practical path planning cases many solutions can be discarded *a priori* by exploiting the result of the following Lemma:

**Lemma 2** - A necessary and sufficient condition to ensure that the optimization problem (10) admits a feasible solution

with  $\alpha_{i,j} = 1, i \leq j$ , is that there exist points  $p_{to}, p_l \in \mathbb{R}^2$  and a positive scalar  $t \in \mathbb{R}_{\geq}$  ensuring that:

$$\begin{cases} \left( \|p_{to} - q_i\| + \sum_{k=i}^{j-1} \|q_{k+1} - q_k\| + \|q_j - p_l\| \right) \leq V_v t \\ (\|p_{to} - p_l\|) \leq V_c t \\ t \leq \bar{a} \end{cases} \quad (11)$$

*Proof:* It is enough to note that if no feasible solutions for (11) exist when  $\alpha_{i,j} = 1$ , then the constraints (8) cannot be satisfied altogether. Otherwise, if a solution to (11) exists then a feasible solution for (10) can be build by using  $\alpha_{i,j} = 1, p_{to,i} = p_{to}, p_{l,j} = p_l$  ■

The previous result can be used to pre-process the problem and discard solutions that consider groups of points too far from each others in the same group. Then, if we denote with  $\Omega$  the set of indexes  $(i, j)$  for which the previous lemma's results allow  $\alpha_{i,j}$  to be equal to one, we can simplify (10) by imposing

$$\alpha_{i,j} = 0, \quad \forall (i, j) \notin \Omega. \quad (12)$$

It is also worth noticing that the computation of  $\Omega$  using the convex feasibility problem (11) is solved in at most  $n(n-1)/2$  steps. While in some cases of interest the above pre-processing task renders the optimization problem tractable, in most of the cases the problem remains computationally prohibitive. For such a reason, heuristic approaches will be proposed in this paper. In particular, we will build up a solution that minimizes the number of takeoffs of the faster vehicle. Interestingly enough, the minimal number of takeoffs can be easily determined by means of the following Lemma

**Lemma 3** - Given a list of points  $q_{list} = [q_1, \dots, q_n]$ , the minimum number of takeoffs required to visit it can be determined by solving the following linear programming optimization problem

$$\begin{aligned} \min_{\alpha_{i,j} \in \mathbb{R}} \sum_{(i,j) \in \Omega} \alpha_{i,j} \\ \text{subject to} \\ \sum_{(i,j) \in \Omega: i \leq k \leq j} \alpha_{i,j} = 1, \quad k = 1, \dots, n \\ \alpha_{i,j} \geq 0 \quad \forall (i, j) \in \Omega \end{aligned} \quad (13)$$

*Proof:* As a result of Lemma 2, for any admissible solution of (6)-(10) we have that  $\alpha_{i,j} = 0, \forall (i, j) \notin \Omega$ . Then, the minimum number of  $\alpha_{i,j}$  to be equal to one can be computed by solving the following Integer Linear Programming problem

$$\begin{aligned} \min \sum_{(i,j) \in \Omega} \alpha_{i,j} \\ \text{subject to (6), (7),} \\ \alpha_{i,j} = 0, \quad \forall (i, j) \notin \Omega. \end{aligned} \quad (14)$$

It remains to prove that by relaxing the integer constraints in (14), all vertices of the resulting Linear Programming problem (13) are integer. This can be proved by means of standard technicalities here omitted for the sake of brevity (see Appendix for details). ■

An heuristic algorithm based on the above Lemma may be summarized as follows

**HEURISTIC 1 - MINIMUM NUMBER OF TAKE-OFFS HEURISTIC**

- 1) Determine the set  $\Omega$  of couples  $(i, j)$  such that  $\alpha_{i,j} = 1$  is admissible for (11);
- 2) Determine the values of  $\alpha_{i,j}^*, i = 1, \dots, n, j = i, \dots, n$  as the optimal solution of the linear programming problem (13);
- 3) Substitute  $\alpha_{i,j} = \alpha_{i,j}^*, i = 1, \dots, n, j = i, \dots, n$  into (10) and solve the resulting convex optimization problem.

As it will be shown in the numerical simulations, the above heuristic enjoys a poor performance w.r.t. the real optimal solution. However, its structure may be exploited to build up more efficient heuristics. To this end, it is important to understand that, typically, the visit of a set of points “in a row” is not cost effective when the limited autonomy of the faster vehicle forces the carrier to make large deviations to “rescue” it. Such an anomalous behavior usually appears when the distance covered by the vehicle in visiting the points amongst  $q_i$  and  $q_j$ , that we will denote hereafter as  $d_{i,j}^v = \sum_{k=i+1}^j \|(q_i - q_k)\|$  or the distance between the first and the last point of the sequence, i.e.  $d_{i,j}^c = \|(q_i - q_j)\|$  is “large.” To justify the above statement, let us consider a very simple case in which  $p_0 = [0\ 0]^T$ ,  $p_f = [8\ 0]^T$ ,  $q_1 = [3.75\ 0.1]^T$ ,  $q_2 = [4\ r]^T$  and  $q_3 = [4.25\ 0.1]^T$  and where  $r \in [2, 2.3850]$  is a scalar parameter that it is used to increase  $d_{1,3}^v$ . The above benchmark has been intentionally and specifically constructed so that the optimal solution for each admissible value of  $r$  is achieved by visiting each point separately. In Figure 3 it is shown that the optimal value and the value obtained by visiting the three points all in a row coincides for small values of  $d_{1,3}^v$  while they diverge once a certain threshold is trespassed. As depicted in Figure 4

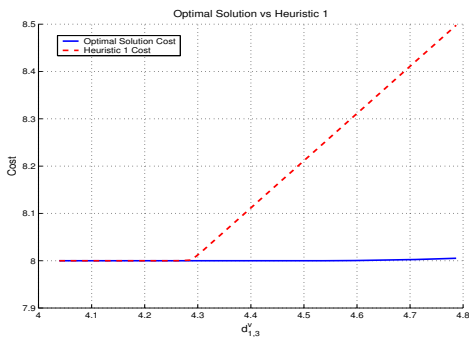


Fig. 3. Comparison between the optimal solution and the solution obtained by means of the Heuristic 1 for growing  $d_{1,3}^v$

this is due to the big deviation from the straight line the carrier is forced to follow so as to recover the vehicle. An analogous situation is obtained by increasing the distances  $d_{i,j}^c$  for a constant  $d_{i,j}^v$ . The above considerations have been exploited to design two new heuristics based on the idea of excluding all that combinations of points such that  $d_{i,j}^c$  (or respectively  $d_{i,j}^v$ ) is larger than a certain threshold. To this

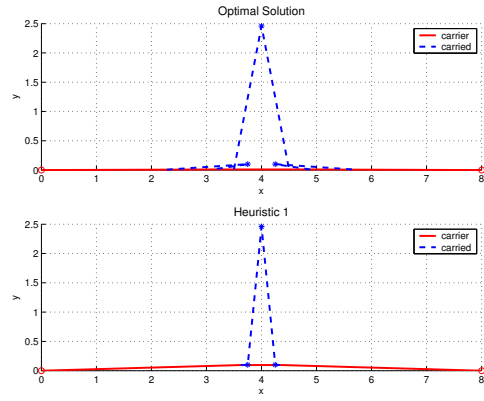


Fig. 4. Comparison between the optimal solution and the solution obtained by means of the Heuristic 1 for  $d_{1,3}^v = 4.5374$

end we will first build two lists  $list^c$  and  $list^v$  each one of them containing all the couples  $(i, j) \in \Omega$  and ordered so as to have decreasing values of  $d_{i,j}^c$  and  $d_{i,j}^v$  respectively. Let  $k_f$  be the cardinality of the set  $\Omega$ , the following heuristics can be defined:

**HEURISTIC 2 [3] - MINIMUM NUMBER OF TAKE-OFFS HEURISTIC WITH THRESHOLD ON  $d^c$  [ $d^v$ ]**

0. Initialization

- 0.1 Determine the set  $\Omega$  of couples  $(i, j)$  such that  $\alpha_{i,j} = 1$  is admissible for (11);
- 0.2 Determine the list  $list^c$  [ $list^v$ ] of couples  $(i, j) \in \Omega$  such that  $d_{i_k, j_k}^c \geq d_{i_{k+1}, j_{k+1}}^c$  [ $d_{i_k, j_k}^v \geq d_{i_{k+1}, j_{k+1}}^v$ ] where  $(i_k, j_k)$  denotes the  $k$ -th elements of  $list^c$  [ $list^v$ ];
- 0.3 Set  $c^* = \infty$ ,  $\alpha_{i,j}^* = 0, i = 1, \dots, n, j = i, \dots, n$ ;

1. for  $k = 1 : k_f + 1$

- 1.1 Determine the values of  $\tilde{\alpha}_{i,j}, i = 1, \dots, n, j = i, \dots, n$  as the optimal solution of the linear programming problem (13);
- 1.2 Substitute  $\alpha_{i,j} = \tilde{\alpha}_{i,j}, i = 1, \dots, n, j = i, \dots, n$  into (10), solve the resulting convex optimization problem and obtain the cost  $\tilde{c}$ ;
- 1.3 if  $\tilde{c} < c^*$ , set  $c^* = \tilde{c}$  and  $\alpha_{i,j}^* = \tilde{\alpha}_{i,j}, i = 1, \dots, n, j = i, \dots, n$ ;
- 1.4 Eliminate the couple  $(i_k, j_k)$  from the set  $\Omega$ , i.e.  $\Omega = \Omega \setminus (i_k, j_k)$

Note that, being  $k_f \leq n(n+1)/2$ , Heuristics 2 and 3 have a computational complexity time at most  $n(n+1)/2$  times the one of Heuristic 1.

**Numerical Results -** In order to evaluate the performance of the proposed heuristics we considered as a case study a carrier vehicle with maximum velocity  $V_c = 1$  and a carried vehicle with  $V_v = 5$  and normalized operating range  $\bar{a} = 1$ , both starting from an initial position  $p_0 = (0, 0)$ , having to visit 10 target points  $q_i$ , with  $i = 1, \dots, 10$ , randomly generated in a  $5 \times 5$  box and then coming back to  $p_f = p_0$ . A total number of 3000 randomly generated instances of the problem have been considered. The results have synthetically be reported in Table I. As expected the performance of Heuristic 1 is poor, while evident improvements may be

	Heuristic 1	Heuristic 2	Heuristic 3
Average Degradation	17, 53%	1, 08%	0, 26%
Maximum Degradation	124, 29%	22, 23%	6, 37%

TABLE I  
HEURISTICS 1,2,3 VS. OPTIMAL SOLUTION

obtained when using Heuristic 2 and Heuristic 3. Moreover, it is worth remarking that by always picking the smaller cost obtained by the two heuristics the Average Degradation reduces to 0.22% and the Maximal Degradation to 5.16%.

### B. Traveling Salesman Problem with prescribed takeoff-landing sequence

The last mission planning problem considered in this paper is a version of the Traveling Salesman Problem (TSP) for the multi-vehicle system under consideration. Let us assume that the initial positions of both vehicle and carrier are the same  $p_c(0) = p_v(0) = p_0$  and that an unordered set of  $n$  points  $q_{set} = \{q_1, \dots, q_n\}$  to be visited are given. The goal is to determine the optimal trajectory for the vehicle that, from its initial position, touches all  $n$  points one and only one time and returns to the initial point  $p_0$  (i.e.  $p_f \equiv p_0$ ) for landing on the carrier vehicle and concluding the cycle. As well known, TSP problems are typically NP-Hard. For such a reason, in most practical cases, heuristics or constant-factor approximations are used. In this paper, in order to deal with the particular TSP problem at hands, hereafter denoted as Carrier/Carried-TSP (CC-TSP), we propose a heuristic algorithm based on the Euclidean TSP. The Euclidean TSP (E-TSP) is a particular case of the general TSP in which, given  $n$  points in the space, we want to determine the optimal sequence that minimizes the sum of the Euclidean distances amongst consecutive points. One of the main feature of this class of TSP problems is that, although still NP-Hard, it admits a polynomial-time approximation scheme (see [15]). This means that, for any scalar  $e > 0$ , it is possible to find in a polynomial time a tour whose length is at most  $(1 + 1/e)$  times larger than the optimal length. Then, in practice, for any instance of the  $E - TSP$  we can obtain an *almost-optimal* solution in a reasonable time. The CC-TSP heuristic here proposed consists of the following two steps:

- 1) determine the visiting order of the *almost-optimal* E-TSP tour for the set of points  $\{p_0\} \cup q_{set}$ ;
- 2) use the above visiting order and solve the resulting ordered visit of  $n$  points by means of the convex optimization procedure detailed in [13]

The idea behind this approach is that, as it will be clear soon, the completion time of the CC-TSP is connected to the sum of the distances between points, and then, the minimization of E-TSP leads usually to achieve a reasonably good CC-TSP solution. In particular, it is possible to prove that:

**Lemma 4** - Let  $p_0$  and the set of  $n$  points  $q_{list}$  to be visited be given. Let  $\ell_{ETSP}^{opt}$  be the length of the optimal E-TSP tour for the given set of points. Then, a lower bound to CC-TSP is given by  $t_L(\ell_{ETSP}^{opt}, n)$ .

*Proof:* By recalling the definition of the lower bound (3) and assuming the number of takeoff points  $n$  given (according

to the cardinality of the set  $q_{list}$ ), it follows that  $\ell_1 \leq \ell_2 \Rightarrow t_L(\ell_1, n) \leq t_L(\ell_2, n)$ . Let  $\ell$  be the length of a generic hamiltonian cycle for the points  $p_0, q_{list}$  starting at the point  $p_0$ . From the definition of E-TSP, it results that  $\ell_{ETSP}^{opt} \leq \ell$ . Then,  $t_L(\ell_{ETSP}^{opt}, n) \leq t_L(\ell, n)$ . This proves that the lower-bound computed considering the optimal E-TSP tour is also a lower-bound for the CC-TSP because no other choice for the visiting order can obtain a lower value. ■

Moreover, by recalling the upper bound introduced in Section III it is possible to bound the maximal error achieved by the proposed heuristic :

**Lemma 5** - Let the set  $\{p_0\} \cup q_{list}$  be given and  $\ell_{ETSP}$  denote the length of the  $(1 + 1/e)$ -approximated optimal E-TSP tour with  $e \geq 0$ . Then, the completion time  $t_{CC-TSP}^{heu}$  obtained with the CC-TSP heuristic has a cost which is at most  $\varepsilon$  times the optimal one with  $\varepsilon$  given by

$$\varepsilon := \frac{t'_U(\ell_{ETSP}, n, \theta_{list})}{t_L\left(\frac{\ell_{ETSP}}{1+1/e}, n\right)}$$

where  $t'_U(\cdot)$  is defined in (4).

*Proof:* Because  $\ell_{ETSP}$  denotes the length of the *quasi-optimal* E-TSP tour we have that the optimal length  $\ell_{ETSP}^{opt}$  is bounded from below by a function of the scalar parameter  $e$ ,  $\ell_{ETSP}^{opt} \geq \ell_{ETSP}/(1 + 1/e)$ . Applying Lemma 4 we have that the optimal solution of the CC-TSP is then greater than or equal to  $t_L\left(\frac{\ell_{ETSP}}{1+1/e}, n\right)$ . Moreover, by following Section III arguments, it is possible to build up an upper bound considering the sequence of points obtained with the *almost-optimal* E-TSP and the angles  $\theta_{list}$  which corresponds to this solution. ■

Please note that in the particular for the case in which the points to be visited are *sufficiently far* each others (i.e.  $d_{min}/V_v > \bar{a}$ ) and the angles  $\theta_i$  formed by the segments connecting the points in the order given by the ETSP algorithm satisfy  $\theta_i \leq 2 \arcsin(V_c/V_v)$  then  $\varepsilon = t_L(\ell_{ETSP}, n)/t_L\left(\frac{\ell_{ETSP}}{1+1/e}, n\right)$  and thus the optimal ETSP sequence of points is also optimal for CC-TSP.

**Numerical Results** - Results from numerical simulations have been analyzed to compare the optimal solution of the CC-TSP with the one obtained by minimizing the ETSP cost. Again, we use as a case study a carrier with maximum velocity  $V_c = 1$  and a vehicle with  $V_v = 5$  with normalized operating range  $\bar{a} = 1$  which, starting from an initial position  $p_0 = (0, 0)$ , has to visit 5 randomly generated points  $q_i, i = 1, \dots, 5$  and come back to  $p_f = p_0$ . We consider three different scenarios according to how the points are generated: first we consider the ND (Normal Distance) case where the points are randomly generated in a box  $50 \times 50$ , than, to evaluate what happens when the points are confined in more restricted areas, we considered the SD (Short Distance) and the VSD (Very Short Distance) cases where the points are generated within boxes of dimension  $20 \times 10$  and  $10 \times 10$  respectively. Results are reported in Table II where *Cases #* denotes the number of samples considered, *Opt. Sol.* the percentage of cases in which the optimal sequence generated

