ISSN 2281-4299

DIPARTIMENTO DI INGEGNERIA INFORMATICA
AUTOMATICA E GESTIONALE ANTONIO RUBERTI

SAPIENZA
UNIVERSITÀ DI ROMA

An adaptive Cooperative Receding Horizon controller for the multivehicle routing problem

Giorgia Chini
Guido Oddi
Antonio Pietrabissa

Technical Report n. 8, 2012

An adaptive Cooperative Receding Horizon controller for the multivehicle routing problem

December 2012

Giorgia Chini, Guido Oddi, Antonio Pietrabissa

Department of Computer, Control, and Management Engineering "Antonio Ruberti"
University of Rome "Sapienza"
via Ariosto 25, 00185, Rome (IT)

Abstract: The objective of the Vehicle Routing Problem (VRP), in the meaning of this paper, is to find the best path for a vehicle, or the best paths for a fleet of vehicles, with the aim of visiting a set of targets. Possible applications of the vehicle routing problem include surveillance, exploration, logistic, transportation, relief systems, etc. A lot of research has been carried out so far, but the VRP remains a complex and computationally expensive combinatorial problem, leading to the difficulty to actually solve the problem on-line. This paper presents a technique based on the Cooperative Receding Horizon (CRH) approach proposed in [Li06], in which a sequence of optimization problems are computed over a planning horizon and the decisions are applied only over a shorter action horizon, in order to rapidly adapt to possible configuration changes (e.g., new targets appearance). Moreover, the proposed algorithm is able to dynamically adapt to the time-variable configuration of both vehicles and targets as well as to handle the discovery of unknown targets. Several proof of concept simulations show the enhancements of the proposed technique in comparison to the one in [Li06].

Keywords: Vehicle Routing Problem, Receding Horizon Control.

Index

1 Introduction

Recent developments in computer technology and wireless communications opened numerous possibilities for lining up networks of mobile vehicles capable of interacting with the environment to accomplish specific objectives. The possible applications to exploit such new capabilities cover several fields, such as surveillance, exploration, logistic, transportation, relief systems, etc. The vehicles should also be capable of cooperating to perform a mission with a common goal.

The model used in this paper is the one defined in [Li06]. *M* vehicles operate in a limited two-dimensional space and travel to cover a set of *N* targets. *N* and *M* can vary in time (e.g., new targets may appear or some vehicles may become unavailable). Each target is assigned with a value, representing the *reward* of covering the target, which decreases with time. The *mission* is characterized by the process to cover all the targets with the available set of vehicles. The objective is to accomplish the mission collecting the maximum reward.

The algorithms solving this problem belong to the *cooperative control* framework. Receding Horizon Control (RHC) may be used to solve these problems. [Li06] proposed a control technique, in which a sequence of optimization problems is solved over a *planning* horizon and it is executed over a shorter *action* horizon: in this way, the vehicles are able to detect new events (e.g., the appearance of new targets) and, accordingly, to rapidly modify their trajectories on-line (*edge-and-react* method).Vehicles cooperate in the sense that the mission space is dynamically divided among them; a time-variable responsibility region of each vehicle is defined, which depends on the current targets and vehicles positions. As explained in [Li06], the optimization problem carried out at each time step does not attempt to make *any explicit vehicle-to-target assignment*, but only to determine headings which, at the end of the current planning horizon, will place vehicles at positions such that the total expected reward is maximized. The simulation results presented by the authors of [Li06] are very encouraging, since the algorithm is capable (i) to rapidly adapt to incoming events, and (ii) to divide the mission space among the vehicles achieving approximated Voronoi partitions [AURE91]. The main drawback reported by such algorithm is the inefficiency in case of clustered configurations (relatively both to vehicles and targets) and the lack of convergence in certain configurations (see also [CASS03] of the same authors).

To overcome the intrinsic issues of the approach followed by [Li06] and to increase the efficiency of the algorithm, this work proposes an extension of the approach followed in [Li06].

1.1 State of the art and proposed innovation

The Vehicle Routing Problem (VRP) is a generalization of the classical Travel Salesman Problem (TSP), and it consists in finding the optimal (or near-optimal) path (according to specific objective functions – e.g., the shortest path) for a vehicle to visit a set of locations. If the distance between two locations is the Euclidean distance, the problem is referred as Euclidean VRP. The VRP has been firstly studied by Dantzig et al. [DANT59], and it is still a topic of research because of intrinsic hard combinatorial nature. Transportation, logistics, manufacturing, mobile robots systems, military and relief systems are examples of applications involving the solution of this particular kind of problem. Because of the magnitude of the fields of applications, during the years numerous variants of the classic VRP arose. Specific features and constraints have been added during the time to take into account: (i) the complexity of the new systems, e.g., fleets of vehicles covering a mission space; (ii) the requirements of the customers, e.g., time windows or multiple visits; (iii) the decision context, e.g., the change of the routes because of traffic congestion, etc. According to the taxonomy defined in [PILL11], VRP with all modifications can be grouped into four categories:

- *Static and deterministic problems*: this category groups the problems characterized by the complete knowledge of all the inputs and by the invariance of vehicle routes. There is a wide literature on this topic: recent reviews of exact and approximate methods are those of [CORD07] in the case of a single vehicle, [BALD07], [LAPO07], [LAPO09] and [TOTH02] in the case of multiple vehicles.
- *Static and stochastic problems*: this category groups the problems characterized by the complete knowledge of all the inputs, as in the former case, but some or all of them are random variables, in the sense that they are realization of stochastic processes. This lead to the fact that, even if routes are still a-priori assigned, small changes in the routes are allowed. Uncertainty may concerns any of the input data: (i) stochastic customers, which need to be served with a given probability (e.g., [BERT88] for one or two vehicles and [WAT89] in the case of one vehicle); (ii) stochastic times, if the service or travel times are modeled by random variables (e.g.,[VERW03], for a single vehicle and [KENY03] and [LAPO92]f or multiple vehicles); stochastic demands (e.g., [LAPO02], [MEND10] and [MEND11] for a single vehicle case, [CHRI07], [DROR89], [SECO00] and [SECO09] for multiple vehicles case).
- *Dynamic and deterministic problems*, in these problems, vehicles routes are redefined during their fulfillments, since new information may be available during the time (e.g. in case of new targets appearance). Technological support for real-time communication among the vehicles and with the control centers (e.g., through mobile phones and using the GPS) is obviously needed, because new input data are dynamically revealed during the planning or the execution of the routes.
- *Dynamic and stochastic problems*: in these problems the variable nature of the dynamically revealed information is stochastic.

This paper deals with dynamic Euclidean VRP in both deterministic and stochastic scenarios. For the time being, dynamic and deterministic routing problems have been proposed to be solved using exact or approximated methods (e.g. through specific heuristics).

In the context of dynamic VRP in deterministic scenarios, solved with exact methods, [CHEN06] used a linear programming method to solve a dynamic Euclidean VRP with time windows (VRPTW) in a deterministic scenario. This problem is a generalization of the VRP in which each customer must be visited within a specified time interval, using a fleet of vehicles. In such cited work, a dynamic column generation algorithm (DYCOL) is developed, consisting of an iterative optimization which, at each time step, dynamically generates new feasible vehicle trips (new *columns*) starting from a reduced column set and the decision is applied for a fixed *implementation epoch*. During the epoch new events may verify and the algorithm is re-executed, potentially adding new columns to the problem. [BALD08] presented an exact algorithm to solve the Capacitated Vehicle Routing Problem (CVRP), which is the VRP with limited vehicles capacity, for a set of m identical vehicles. The algorithm of [BALD08] consists in the sequence of three heuristics to increase the quality of the solution: the first heuristic is based on the q-route relaxation of the set partitioning formulation of the CVRP; the second one combines Lagrangean relaxation, pricing and cut generation and the third attempts to reduce the gap left by the first two procedures using a classical pricing and cut generation technique. [ROBE12] proposed an exact method for solving the CVRP and the VRPTW, which generates a reduced problem by replacing the original route set with a smaller route set and solving the reduced problem with an integer programming (IP) solver.

With respect to heuristic approaches to solve the dynamic VRP in deterministic scenarios, several techniques were developed so far. Tabu search technique was applied to a dynamic deterministic Euclidean routing problem with multiple vehicles in [GEND99]. The main idea of their approach is to maintain a set of 'good' solutions (that constitutes a sort of adaptive memory), which is used to generate initial solutions for a parallel tabu search. The parallel search is done by partitioning the routes of the current solution, and by

optimizing them independently. Whenever a new customer request arrives, it is checked against all the solutions from the adaptive memory to decide if it should be accepted or rejected. A two-phase heuristic algorithm was developed in [HSUE08] to solve the dynamic VRP in a deterministic scenario for relief logistic in natural disasters with fleets of vehicles: the first phase is characterized by routes construction, and in the second phase routes are improved. Another heuristic to solve the dynamic, multi-vehicle VRP in a deterministic scenario was proposed in [NALL09]. In this algorithm, the cities are clustered in subsets, using the *k*-means clustering algorithm, and each vehicle is assigned to a subset: a single VRP is then solved for each vehicle by a genetic algorithm.

Dynamic problems in stochastic scenarios are solved with methods that analytically integrate the stochastic knowledge of input data in the used mathematical model: practically, the mathematical formulation of the problem explicitly considers the stochastic nature of some inputs data, such as stochastic times of demands, stochastic service times, etc. In [BERT92], for example, a policy able to serve demands over an infinite horizon minimizing the expected system time of the demands is presented for *m* identical vehicles with unlimited capacity. Other strategies used to solve dynamic VRP in stochastic scenarios are the so-called waiting strategies, in which a controller decides if a vehicle should wait for an amount of time after serving a request before heading towards the next customer: this strategy was implemented in various frameworks for dynamic VRP [BRAN05], and dynamic VRPTW [BRAN09], in both cases for a fleet of vehicles. Otherwise, a vehicle can be relocated to a strategic position, where new request are likely to arrive ([LARS01]). This latter strategy is noteworthy since several heuristics, for instance serving emergency fleet deployment problems or emergency vehicle dispatching (or redeployment) problems, are based on this strategy (see for example [GEND01], and [HAGH07], both in the case of multiple vehicles). More recently, *cooperative control* arose in the context of team theory and was used for coordinating a team of vehicles that has a common objective (e. g., a set of vehicles aiming at visiting a set of targets [CASS10], or destination points in post-disaster response [CASS11]). These approaches enable on-line problem solving and avail of the amount of information which can be shared among vehicles to coordinate their activity.

For the time being, the VRP remains a complex combinatorial problem and all the reported solution methods are computationally expensive, leading to the difficulty to actually solve the problem on-line. To develop fast controllers, suited also to uncertain environments, a new methodology was introduced in [LI06], referred to as Cooperative Receding Horizon control (CRH). This approach dynamically determines vehicles trajectories and at the same time takes into account possible uncertainties in the configuration scenario, such as new targets appearance or the elimination of one or more vehicles. As detailed in Section 2, the CRH control dynamically determines the vehicle trajectories by solving a sequence of optimization problems over a *planning horizon* and executing them over a shorter *action horizon*, thus managing uncertainty and reacting to future events as soon as they appear. In this respect, there is a difference with the other controller strategies, which re-plan the whole routes after the new information is arrived (both in the deterministic and in the stochastic cases). That is why this approach is called "*hedge-and-react*" as opposed to "*estimate-and-plan*" methods ([CASS11]). In practice, this control scheme consists in an event-driven route planning, and it is particularly suited for stochastic scenarios. In its basic formulation, it is a centralized scheme: there is a controller with relevant computation and communication ability, which solves an optimization problem as a result of the detection of an event by a vehicle; in [LI05] a distributed version is proposed. Vehicles cooperate, in the sense that, based on vehicles actual position, for each vehicle a responsibility region can be identified (that is an area of competence of that vehicle). As analyzed in Section 2.3 of this paper, such approach presents severe limitations when particular scenarios are considered, for example if the targets are in some way clustered and symmetrically placed over the mission space.

In this paper, the CRH approach was modified by changing the point of view: the above mentioned responsibility regions are defined based on the target positions, leading to the definition of regions of attraction of the targets. The regions borders are dynamics due to the target births and deaths occurrences. This new algorithm, detailed in Section 3.1, is then referred to as target-oriented CRH (tCRH). The obtained routing algorithm has a sort of dual behavior with respect to the original one, and, as analyzed in Section 3.1.3, its performance degrades as the targets become sparser. From the CRH and tCRH analyses, it turns out that they may complement each other. Therefore, in Section 3.2, a mixed strategy is proposed and in Section 3.3 an adaptive strategy, which behaves as the CRH algorithm or as the tCRH algorithm depending on the current targets and vehicles distribution on the mission space. Comprehensive simulation results are presented in Section 4, whereas in Section 5 the conclusion are drawn and some future work is outlined.

2 Cooperative Receding-Horizon (CRH) approach

This Section defines the vehicle routing problem described in [LI06], which, as already mentioned, is the basis of the algorithm proposed in Section 3. The interested reader is referred to [LI06].

2.1 Problem statement

The mission duration is T and the current mission time is denoted with $t \in [0,T]$. The mission space S is a two-dimensional area, with N target points, denoted with $y_i(t)$, $i = 1,2,\ldots,N(t)$, and collected in the set T, and $M(t)$ vehicles, denoted with j, $j = 1,2,\ldots,M(t)$, and collected in the set A. The vehicle position at time t is denoted with $x_j(t)$, and the initial position at time $t_0 = 0$ with x_{j0}. For the sake of simplicity, the vehicles velocities are constant and equal to V_j. The position and velocity of vehicle j then defined by the following equation:

$$\dot{x}_j(t) = V_j \begin{bmatrix} \cos u_j(t) \\ \sin u_j(t) \end{bmatrix}, \; x_j(0) = x_{j0}, \tag{2.1}$$

where $u_j(t) \in [0,2\pi]$ is the vehicle heading at time t.

The target y_i has an associated reward function $R_i\Phi_i(t)$, where R_i is the reward and $\Phi_i(t)$ is a discounting function, which describes how the reward decreases over time. An example of $\Phi_i(t)$ is given by the following equation:

$$\Phi_i(t) = 1 - \frac{\alpha_i}{T} t, \;\; \alpha_i \in (0,1]. \tag{2.2}$$

By properly modifying it, the discounting function may describe also deadlines and obstacles (e.g., with negative rewards).

To capture possible differences in the vehicles capabilities, the capability factor of vehicle j with respect to target i is defined by the function $p_{ij}(t)$, which expresses the probability that vehicle j collects the reward $R_i\Phi_i(t)$ when it visits target i; the capability function may also decrease after the vehicle visit to a target. Target y_i is considered as visited by vehicle x_j if their relative distance is less than a threshold s_i: $\|y_i - x_j\| \leq s_i$.

The cooperation among vehicles is achieved by dynamic partitioning of the mission space. Let $B_i^2(t)$ be the set of the 2 nearest vehicles to target i at time t, and let the *relative distance function* $\delta_{ij}(t)$ be defined as follows:

$$\delta_{ij}(t)=\begin{cases}\dfrac{\|y_i-x_j(t)\|}{\sum_{k\in B_i^2(t)}\|y_i-x_k(t)\|}, & \text{if } j\in B_i^2(t)\\ 1, & \text{otherwise}\end{cases}. \tag{2.3}$$

The *relative proximity function* $q_{ij}(t)$ is then defined as follows:

$$q_{ij}(t)=\begin{cases}1, & \text{if } \delta_{ij}(t)\le\Delta\\ \dfrac{1}{1-2\Delta}\left[(1-\Delta)-\delta_{ij}(t)\right], & \text{if } \Delta<\delta_{ij}(t)\le 1-\Delta\\ 0, & \text{if } \delta_{ij}(t)>1-\Delta\end{cases}, \tag{2.4}$$

where $\Delta \in [0,0.5)$ is the *capture radius*. By observing that $\sum_{j\in B_i^2(t)} q_{ij}(t)=\sum_{j\in \mathrm{A}} q_{ij}(t)=1$, $q_{ij}(t)$ is interpreted as the probability that target y_i is assigned to vehicle j; note that if the *relative distance* $\delta_{ij}(t)$ is less than Δ, $q_{ij}(t) = 1$ and target i is assigned to the sole vehicle j. The functions $q_{ij}(t)$ determine areas of the mission space, named *full responsibility regions* S_j, where a subset of targets is assigned to a vehicle j only; if a target is outside of these areas (in the so-called *cooperative regions* C_j), it is assigned to two vehicles, with a 'strength' which is proportional to the value of the relative distances. Note that if $\Delta = 0.5$, the Voronoi tessellation is achieved, and $S=\bigcup_{j\in A} S_j$.

The CRH algorithm works by assigning the trajectories to the vehicles with the objective of maximizing the total reward. The trajectories are assigned by an event-based on-line controller, which assigns the vehicles headings at time instant t_k by solving an optimization problem $\mathbf{P}_k$, $k = 0,1,\ldots$.

The problem variable at time t_k is then the vector $\mathbf{u}_k = [u_1(t_k) \ldots u_M(t_k)]$. Let H_k be the planning horizon. Recalling equation (2.1), the planned positions of the vehicles at time $t_k + H_k$ associated to the solution $\mathbf{u}_k$ is then:

$$x_j(t_k+H_k)=x_j(t_k)+\dot{x}_j(t_k)H_k. \tag{2.5}$$

The planned horizon H_k is chosen as the minimum time occurring to a vehicle to reach a target:

$$H_k=\min_{i\in T(t_k),\, j\in A(t_k),\, p_{ij}(t_k)}\left\{\frac{\|y_i-x_j(t_k)\|}{V_j}\right\}. \tag{2.6}$$

Define $\tau_{ij}(\mathbf{u}_k,t_k)$ as the earliest time that vehicle can reach target i, starting from $x_j(t_k + H_k)$:

$$\tau_{ij}(\mathbf{u}_k,t_k)=t_k+H_k+\frac{\|y_i-x_j(t_k+H_k)\|}{V_j}. \tag{2.7}$$

If the vehicle reaches the target at time $\tau_{ij}(\mathbf{u}_k,t_k)$, the reward will be equal to $R_i\Phi_i[\tau_{ij}(\mathbf{u}_k,t_k)]$. Accordingly, let us define the following quantities:

$$\widetilde{\Phi}_{ij}(\mathbf{u}_k,t_k)=\Phi_i[\tau_{ij}(\mathbf{u}_k,t_k)], \tag{2.8}$$

$$\widetilde{p}_{ij}(\mathbf{u}_k,t_k)=p_i[\tau_{ij}(\mathbf{u}_k,t_k)], \tag{2.9}$$

$$\tilde{q}_{ij}(\mathbf{u}_k, t_k) = q_{ij}[\delta_{ij}(t_k + H_k)]. \tag{2.10}$$

The optimization problem $\mathbf{P}_k$ is then defined by introducing the following objective function J_{CRH}:

$$\max_{\mathbf{u}_k} J_{CRH}(\mathbf{u}_k, t_k) = \max_{\mathbf{u}_k} \sum_{i=1}^{N(t_k)} \sum_{j=1}^{M(t_k)} R_i \tilde{\Phi}_{ij}(\mathbf{u}_k, t_k) \tilde{p}_{ij}(\mathbf{u}_k, t_k) \tilde{q}_{ij}(\mathbf{u}_k, t_k). \tag{2.11}$$

The control action is implemented over an *action horizon* h_k, determined either by an event occurrence (e.g., a vehicle reaches a target, a new target appears, …), by a constant value or by the following rule:

$$h_k = \begin{cases} 0.5H_k, & \text{if } H_k > r \\ H_k, & \text{if } H_k \le r \end{cases}, \tag{2.12}$$

where r is such that r/V_j is a small distance from the target points, for all $j \in A$.

2.2 CRH properties

In [LI06], some properties of the CRH algorithm are presented, as well as general conditions for the convergence of the trajectories. A vehicle trajectory is a stationary trajectory if the vehicle converges to a target in finite time. In the *M*-vehicles, *N*-targets case, a sufficient condition for the stationarity of the trajectories is given. Moreover, in the 1-vehicle, *N*-target case, a sufficient condition for stationary trajectory is given, as well as a necessary condition for non-stationary trajectory. Unfortunately, there are cases in which such conditions are not met, and the CRH controller generates trajectories exhibiting oscillatory behavior. Also, in some cases, the trajectory may be unstable (i.e., the convergence of the trajectories to the targets is not assured). Those conclusions were already achieved in [LI06] and in [CASS03]. Specifically, Theorem 4 of [LI06] defined conditions of stationary and non-stationary trajectories, in the particular scenario with one vehicle and *N* targets. In order to solve this problem, the same authors proposed in [CASS03] a *direction change cost function* which penalizes the change of direction caused by the oscillations. However, this function heavily depends on the choice of the *C* parameter (see [CASS03] for further details), which must be tuned according to the particular scenario. The following section shows an example of scenario configuration which proves the instability of the CRH and the corresponding lack of convergence.

2.3 Analysis of the limitations of the CRH approach

The simple scenario depicted in Figure 1 shows the oscillatory behavior and the lack of convergence of the trajectories. Figure 1 a) shows a scenario with three targets at points $\{y_1, y_2, y_3\} = \{(-3,0),(0,3),(3,0)\}$, with $R_1 = R_2 = R_3 = 1$ and $\alpha_1 = \alpha_2 = \alpha_3 = 1$, and two vehicles $\{v_1, v_2\}$, with $p_{ij} = e^{-c\tau_{ij}(u_j(t_k))}$, $c = 0.01$, and which, at time t_k, are in points $\{x_1(t_k), x_2(t_k)\} = \{(0,1.5),(-8,8)\}$. The second vehicle is not shown in the figure since it is distant from the targets and from v_1; in particular, it is sufficiently distant to let $q_{i1}(t) = 1$ and $q_{i2}(t) = 0$, $i = 1,2,3$, $t = t_k, t_{k+1}, t_{k+2}$. The target y_1 is the closest to the first vehicle v_1, which is evenly distant from the other targets y_2 and y_3: $\|y_1 - x_1(t_k)\| < \|y_2 - x_1(t_k)\| = \|y_3 - x_1(t_k)\|$. Since only two vehicles are present, they appear in all the sets $B_i^2(t_k)$, $i = 1,2,3$, but, due to the distance of v_2 from the targets, the optimal value of u_1 is independent of u_2. In conclusion, the cost function (2.11) is written as $J[u_1(t_k), t_k] = \sum_{i=1}^{3} R_i \tilde{\Phi}_{i1}[u_1(t_k), t_k] \tilde{p}_{i1}[u_1(t_k), t_k] = \sum_{i=1}^{3} \tilde{\Phi}_{i1}[u_1(t_k), t_k]$. The other simulation parameters were $\Delta = 0.4$, $s = 0.2$, $r = 2$ and $V_1 = V_2 = 0.5$.

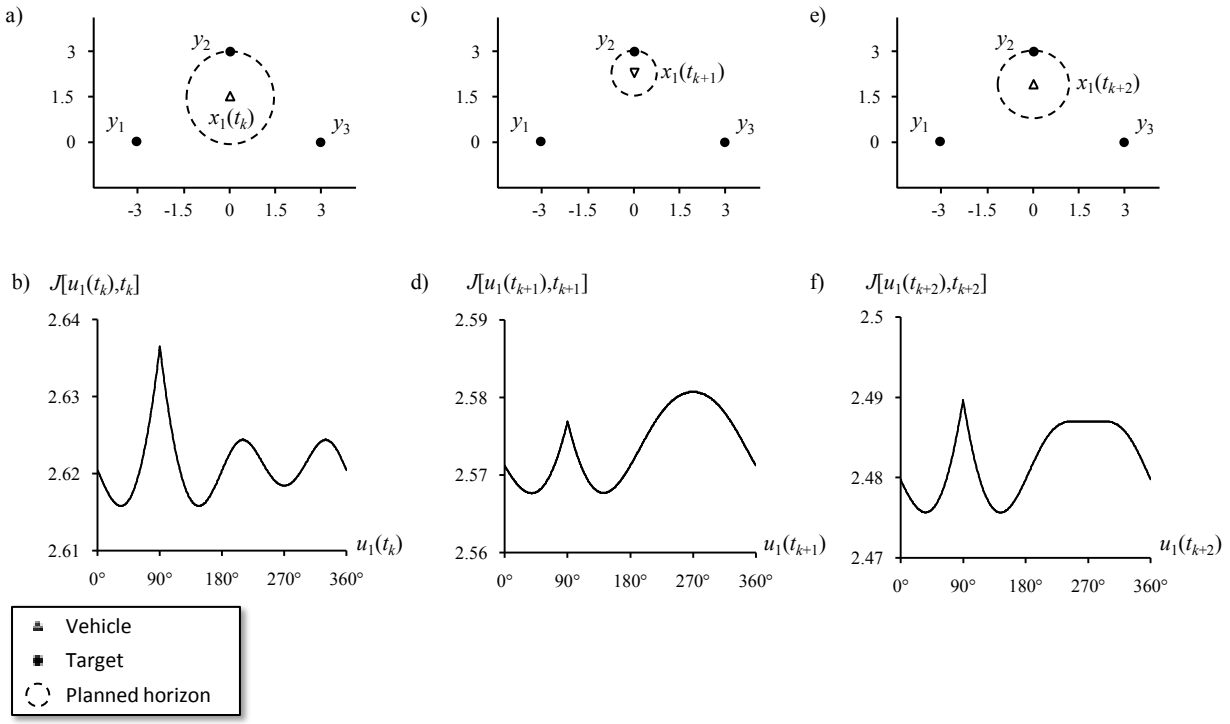


Figure 1: Example of a scenario leading to oscillatory trajectories.

At time t_k, H_k is chosen according to equation (2.6), and is such that, at time $t_k + H_k$, v_1 would move to a point on the circle with radius $\|y_1 - x_1(t_k)\|$. The circle is highlighted in Figure 1 a), and the value of the cost function computed on the circle versus the values of u_1 is shown in Figure 1 b). The figure clearly shows that there are two relative maximum values corresponding to the directions toward the targets y_2 and y_3, and that the absolute maximum is toward the nearest target y_1, i.e., for $u_1 = 90°$. By assuming that no events occur in the meantime, h_k is then set equal to $0.5H_k$ (see equation (2.12)). The direction maximizing the problem $\mathbf{P}_k$ is towards y_1, and is such that v_1 moves to position $x_1(t_{k+1}) = x_1(t_k + h_k) = (0,2.25)$, as shown in Figure 1 c).

As before, at time t_{k+1}, H_{k+1} is such that, at time $t_{k+1} + H_{k+1}$, v_1 would move to a point on the circle with radius $\|y_1 - x_1(t_{k+1})\|$, shown in Figure 1 c). This time, however, Figure 1 c) shows that the cost function over this circle is maximum when $u_1 = 270°$: even if there is a relative maximum towards y_1, the combined attraction of the other two targets is such that the vehicle will move towards them. Accordingly, considering $h_{k+1} = 0.5H_{k+1}$, v_1 will move to position $x_1(t_{k+2}) = (0,1.875)$: as shown in Figure 1 e), the vehicle now moved in the opposite direction with respect to the target 1.At time t_{k+2}, the control action will lead the vehicle towards y_1 again (as shown in Figure 1 f)).

Another scenario which is problematic for the CRH algorithm is when the targets are somehow clustered and few new targets are generated in time. The following figure shows the initial configuration of 15 vehicles and 30 clustered targets (Figure 2 a)) and the vehicle trajectories performed in 30 steps of simulation (Figure 2 b)). The simulation parameters are $\Delta = 0.4$, $s = 0.25$, $r = 2$, $R_i = 1$, for $i \in \{1,2,\ldots,30\}$ and $V_j = 0.5$, for $j \in \{1,2,\ldots,15\}$. The figures show that only one vehicle is attracted by the targets, whereas the other vehicles randomly look for other targets in the mission space. In fact, the sets $B_i^2(t)$ of all the targets $i \in T$ contain only two vehicles, and the targets are on the full responsibility region of only one vehicle, as approximated by the Voronoi partitions highlighted in Figure 2 a) (the Voronoi partitions are obtained with

$\Delta = 0.5$ in (2.4)). Clearly, if the target cluster is conspicuous and few targets are present in the other areas of the mission space, the CRH algorithm is not efficient. In the example, only one vehicle is exploited, with a consequent mission completion delay, while the other vehicles explore the environment. Moreover, the one vehicle in question extremely suffers from the oscillations issue (as explained above), when entering into the cluster (in fact, after 30 steps, only few targets of the cluster are reached). This issue obviously delays the mission completion time and sometimes it even affects the algorithm convergence. See Section 4 for quantitative results showing the inefficiency of such approach.

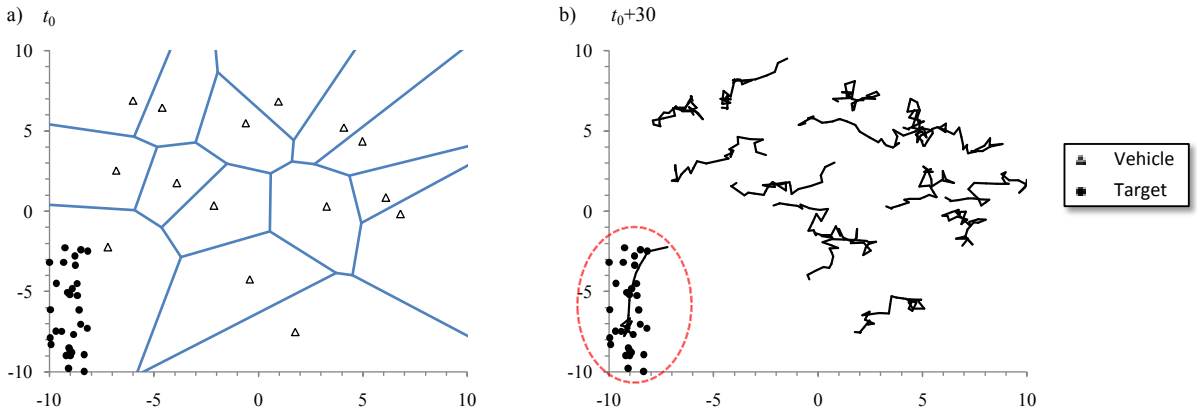


Figure 2: Example of scenario with clustered targets: CRH produces poor performances.

Moreover, by exploring the values of J_{CRH} in the mission plane with different target spatial distributions, it is easy to verify that the stationarity conditions are easier met the more the targets are separated; as soon as a vehicle is surrounded by a cluster of targets, oscillatory phenomena arises with a high probability.

3 Proposed Approach

The basic idea is that the partitioning of the mission space is now performed from the target point of view, instead of the vehicle point of view of [LI06]. For the sake of simplicity, this algorithm is referred to as *target-oriented* CRH, and denoted with tCRH. In fact, with the original CRH, the mission space is dynamically partitioned among the vehicles based on the vehicle positions; over-simplifying, each target is assigned to the nearest vehicle. In contrast, in the tCRH the mission space is partitioned among the targets; over-simplifying, each vehicle is assigned to the nearest target. The partitioning induced by the tCRH is still dynamic due to the fact that target may appear and disappear in time.

As shown in Section 3.1, the two algorithms achieve different vehicle behavior: the basic difference is that, in practice, the CRH algorithm assures that each target is assigned to at least one vehicle, whereas the tCRH algorithm assures that each vehicle is assigned to at least one target. As shown in Sections 2.3 and 3.1.3, depending on the scenario, either the former behavior or the latter is advantageous. On the ground of this consideration, in Section 3.1.3 an adaptive algorithm is conceived, which dynamically favors one behavior of the other one based on current mission developments.

3.1 Target-oriented CRH (tCRH)

3.1.1 Problem statement

The problem statement is very similar to the CRH one defined in Section 2.1. Let $\overline{B}_j^2(t)$ be the set of the two nearest targets to vehicle j at time t, and let the *relative distance function* $\overline{\delta}_{ij}(t)$ be defined now as follows:

$$\overline{\delta}_{ij}(t) = \begin{cases} \dfrac{\left\| y_i - x_j(t) \right\|}{\sum_{k \in \overline{B}_j^2(t)} \left\| y_k - x_j(t) \right\|}, & \text{if } i \in \overline{B}_j^2(t) \\ 1, & \text{otherwise} \end{cases}. \tag{3.1}$$

The *relative proximity function* $\overline{q}_{ij}(t)$ is defined as in equation (2.4). By observing that $\sum_{i \in B_j^2(t)} \overline{q}_{ij}(t) = \sum_{i \in \mathbf{N}} \overline{q}_{ij}(t) = 1$, $\overline{q}_{ij}(t)$ is interpreted as the probability that vehicle j is assigned to target i; note that if the *relative distance* $\overline{\delta}_{ij}(t)$ is less than Δ, $\overline{q}_{ij}(t) = 1$ and vehicle j is assigned to the only target i. The functions $\overline{q}_{ij}(t)$ determine areas of the mission space (named *full responsibility regions*) where the vehicles are assigned to one target only; if a vehicle is outside of these areas (in the so-called *cooperative regions*), it is assigned to two targets, with a 'strength' which is proportional to the value of the relative distances.

As with the CRH algorithm, the tCRH algorithm works by assigning the trajectories to the vehicles with the objective of maximizing the total reward. The trajectories are assigned by an event-based on-line controller, which assign the vehicles headings at time instant t_k by solving an optimization problem $\mathbf{P}_k$, $k = 0,1,\ldots$.

The problem variable at time t_k is again the vector $\mathbf{u}_k = [u_1(t_k) \ \ldots \ u_M(t_k)]$, and the planned positions of the vehicles at time $t_k + H_k$ associated to the solution $\mathbf{u}_k$ is given by equation (2.5). The planned horizon H_k is again chosen as the minimum time occurring to a vehicle to reach a target (see equation (2.6)). The optimization problem is still given by equation (2.11), with the difference that the quantity $\widetilde{q}_{ij}(\mathbf{u}_k, t_k)$ is substituted by the following quantity:

$$\widetilde{q}_{ij}^{tCRH}(\mathbf{u}_k, t_k) = \overline{q}_{ij}[\overline{\delta}_{ij}(t_k + H_k)]. \tag{3.2}$$

and the optimization problem $\mathbf{P}_k$ becomes (introducing the objective function J_{tCRH}) as follows:

$$\max_{\mathbf{u}_k} J_{tCRH}(\mathbf{u}_k, t_k) = \max_{\mathbf{u}_k} \sum_{i=1}^{N(t_k)} \sum_{j=1}^{M(t_k)} R_l \widetilde{\Phi}_{ij}(\mathbf{u}_k, t_k) \widetilde{p}_{ij}(\mathbf{u}_k, t_k) \widetilde{q}_{ij}^{tCRH}(\mathbf{u}_k, t_k) \tag{3.3}$$

3.1.2 tCRH properties

This section introduces theoretical convergence results of the tCRH algorithm, in the particular case of $R_1 = R_2 = \ldots = R_N = R$. Specifically, this section is articulated in three main theorems. If $N = 2$ and the admissible solution space is the bi-dimensional space $\mathbf{R}^2$, *Theorem* 1 demonstrates a) that each minimization problem J_j, with $j \in V$, admits only global minima, b) that all global minima assume the same value and c) that each of them corresponds to the positioning of each vehicle $j \in V$ over one of the targets $i \in T$. *Theorem* 2 extends the results of *Theorem* 1 to the generic case $N > 2$. *Theorem* 3 proves that, if the control horizon H_k is chosen as in equation (2.6) (therefore the minimization problem related to vehicle j is computed over the

admissible solution space (3.1'')), then the algorithm converges in a finite number of steps and guarantees that all the targets will be eventually visited. The following *Lemma* 0, *Lemma* 1 and *Lemma* 2 are used to provide intermediate results required to demonstrate the three main theorems. In the following, for the sake of clearness, the t_k and $\mathbf{u}_k$ arguments of (3.3) are omitted and we will write $\widetilde{q}_{ij}^{tCRH} = q_{ij}$, $\widetilde{\Phi}_{ij}(\mathbf{u}_k, t_k) = \phi_{ij}$ and $\widetilde{p}_{ij}(\mathbf{u}_k, t_k) = p_{ij}$.

The following lemma transforms the tCRH maximization problem (3.1) in a more tractable formulation, useful for the subsequent demonstrations.

Lemma0: Under the assumptions p_{ij}=1 and V_j=V, for all $i \in T$ and $j \in A$, the maximization problem (3.3) at instant t_k is equivalent to the following minimization problem:

$$\min_{\mathbf{u}_k} \sum_{i=1}^{N}\sum_{j=1}^{M} q_{ij} \left\| x_j(t_k + H_k) - y_i \right\|, \tag{3.4}$$

Proof: The result is achieved by direct substitutions of the definition (2.8) of $\widetilde{\Phi}_{ij}$:

$$\begin{aligned}
&\max_{\mathbf{u}_k} \sum_{i=1}^{N}\sum_{j=1}^{M} R\phi_{ij} p_{ij} q_{ij} = \max_{\mathbf{u}_k} \sum_{i=1}^{N} R \sum_{j=1}^{M} \phi_{ij} q_{ij} = \\
&\max_{\mathbf{u}_k} \sum_{i=1}^{N} R \sum_{j=1}^{M} \left[1 - \frac{\alpha}{T} \left(t_k + H_k + \frac{\left\| x_j(t_k + H_k) - y_i \right\|}{V} \right) \right] q_{ij} = \\
&= \max_{\mathbf{u}_k} \sum_{j=1}^{M} R \left(1 - \frac{\alpha}{T} t_k - \frac{\alpha}{T} H_k \right) \sum_{i=1}^{N} q_{ij} + \max_{\mathbf{u}_k} \sum_{j=1}^{M} \left(-R \frac{\alpha}{T} \frac{1}{V_j} \sum_{i=1}^{N} \left\| x_j(t_k + H_k) - y_i \right\| q_{ij} \right) = \\
&= \sum_{j=1}^{M} R \left(1 - \frac{\alpha}{T} t_k - \frac{\alpha}{T} H_k \right) - R \frac{\alpha}{T} \frac{1}{V} \min_{\mathbf{u}_k} \sum_{j=1}^{M}\sum_{i=1}^{N} q_{ij} \left\| x_j(t_k + H_k) - y_i \right\|.
\end{aligned} \tag{3.4'}$$

u_k

Minimizing (3.4) is therefore equivalent to problem (3.3). ■

The admissible solution space of the problem (3.4) is the circumference:

$$F_j(k) = \left\{ w : \left\| w_j - x_j(t_k) \right\| = VH_k \right\}, \ \forall j = 1, \ldots, M. \tag{3.4''}$$

This solution space is such that, at the instant t_{k+H_K}, each vehicle j must move of exactly VH_k meters, starting from the position $x_j(t_k)$ at the previous instant t_k. Since q_{ij} is function of the position x_j of the only vehicle j and of all the targets $i \in T$ (and it does not depend on the position of the other vehicles $k \in V$, $k \neq j$), the minimization problem (3.3) can be rewritten as:

$$\min_{\mathbf{x} \in \bigcup_{j=1,\ldots,M} F_j(k)} \sum_{i=1}^{N}\sum_{j=1}^{M} q_{ij} \left\| x_j - y_i \right\| = \sum_{j=1}^{M} \min_{x_j \in F_j(k)} \sum_{i=1}^{N} q_{ij} \left\| x_j - y_i \right\| = \sum_{j=1}^{M} \min_{x_j \in F_j(k)} J_j, \tag{3.4'''}$$

where:

$$J_j = \sum_{i=1}^{N} q_{ij} \left\| x_j - y_i \right\|. \tag{3.5}$$

Equation (3.5) shows that the minimization (3.4) can be achieved by separately minimizing each objective function J_j, each one over its admissible solution space F_j (3.4''), and the optimum value is the sum of the optimum values computed for all vehicles $j \in V$.

Lemma 1: If the number of vehicles is $M = 1$ and the number of targets is $N = 2$, if $q_{11} = 1$ for target 1 and $q_{21} = 0$ for target 2 and if the admissible solution space for the vehicle is $\mathbf{R}^2$, then the minimization problem admits a global minimum which corresponds to the positioning of the vehicle over the target 1 (i.e. $x_1 = y_1$).

Proof: Since $M = 1$ and $N = 2$, and since $q_{11} = 1$ and $q_{21} = 0$, the minimization problem (3.5) in $\mathbf{R}^2$ becomes:

$$\min_{x_1 \in \mathbf{R}^2} J_1 = \min_{x_1 \in \mathbf{R}^2} \sum_{i=1}^{2} q_{i1} \left\| x_1 - y_i \right\| = \min_{x_1 \in \mathbf{R}^2} \left(q_{11} \left\| x_1 - y_1 \right\| + q_{21} \left\| x_1 - y_2 \right\| \right) = \min_{x_1 \in \mathbf{R}^2} \left\| x_1 - y_1 \right\| \tag{3.6}$$

Since the objective function (3.5) is linear with the distance between x_1 and y_1, the (global) minimum is $x_1 = y_1$. The minimum is global because, in the other region (corresponding to $q_{11} > 0$ for target 1 and $q_{21} > 0$ for target 2), is impossible to obtain the value 0 in (3.5). ■

Lemma 1 shows that, if a vehicle is sufficiently close to a target, then the minimization problem drives the vehicle to the target. The following *Lemma* 2 handles the case in which both q_{11} and q_{21} are positive, and proves that no local minima are present in the corresponding region.

Lemma 2: If the number of vehicles is $M = 1$ and the number of targets is $N = 2$, if $q_{11} > 0$ for target 1 and $q_{21} > 0$ for target 2, and if the admissible solution space for the vehicle is $\mathbf{R}^2$, then the cost function (3.5) admits one stationary point, corresponding to the midpoint $\bar{x} = (y_1 + y_2)/2$ of the segment between y_1 and y_2, and the point $\bar{x}$ is a *saddle* point.

Proof: See the Appendix. ■

Theorem 1: If the number of targets is $N = 2$ and the admissible solution space for each vehicle j is $\mathbf{R}^2$, then the minimization problem of each vehicle j admits two global minima (with the same objective function value) corresponding to the positioning of each vehicle j either over the target 1or over the target 2 (i.e., $x_j = y_1$ or $x_j = y_2$).

Proof: The proof can be easily obtained by combining *Lemma* 1 and *Lemma* 2. The global minimum can be reached only under the conditions of *Lemma* 1. Under the conditions of *Lemma* 2 no local minima are present (only one saddle point in the midpoint between the two targets) and the value of the J_j function (defined in (3.5)) over all x_j belonging to the corresponding admissible solution space is positive. Therefore the minimization problem, over all the $\mathbf{R}^2$ admissible solution space, finds the global minimum under the conditions of *Lemma* 1, i.e., vehicle over the target 1 or, equivalently, over the target 2. The objective function value of (3.5) is 0 in both cases. ■

Theorem2: If the number of targets is $N > 2$ and the admissible solution space for each vehicle j is $\mathbf{R}^2$, then the minimization problem of each vehicle j admits N global minima, each with the same objective function value, corresponding to the positioning of the vehicle j over one of the N targets.

Proof: The main difference with the assumption of *Theorem* 1 ($N = 2$) is that the specific two closest targets of each vehicle j change depending on the position x_j of the vehicle. Then *Theorem* 1 cannot be directly applied for $N > 2$, because the targets 1 and 2 change in function of x_j. In order to exploit the results of *Theorem* 1, fixed the vehicle j, the admissible space $\mathbf{R}^2$ can be divided into a finite set of $m = \binom{N}{2}$ partitions $C_j^1, C_j^2, \ldots, C_j^m$ (s.t. $C_j^1 \cup C_j^2 \cup \ldots \cup C_j^m = \mathbf{R}^2$ and $C_j^l \cap C_j^e = \emptyset, \ \forall l \le m, e \le m, l \ne e$). Each couple of targets (h,k) induces one specific partition $C_j^{(h,k)}$ defined as the continuous region of $\mathbf{R}^2$ such that, for all $x_j \in C_j^{(h,k)}$, the two closest targets to vehicle j are the targets with positions y_k and y_h:

$$C_j^{(h,k)} = \left\{ x_j \in R^n \,\middle|\, \|x_j - y_k\| < \|x_j - y_n\| \ \textbf{and} \ \|x_j - y_h\| < \|x_j - y_n\|, y_n \in \mathbf{T} \setminus \{y_k, y_h\} \right\} \tag{3.7}$$

Notice that $C_j^{(h,k)} = C_j^{(k,h)}$. The minimization of the cost function over $\mathbf{R}^n$, relatively to vehicle j, is then decomposed in m sub-problems, each one over the specific admissible set $C_j^{(h,k)}$ and the final result is the minimum among the m minimization problems.

$$\min_{x_j \in \mathbf{R}^2} J_j = \min_{(h,k) \in T \times T} \min_{x_j \in C_j^{(h,k)}} \left(\|x - y_h\| q_{hj} + \|x - y_k\| q_{kj} \right) \tag{3.8}$$

Since each of the m minimization problems admits, as a consequence of *Theorem* 1, two global minima (in correspondence of the two targets) with objective function value 0, then the final result admit N global minima (with the same value 0), each one in correspondence of a specific target y_i. ■

Theorem3: If the control horizon H_k is chosen as in equation (2.6), and the minimization problems are computed over the admissible solution space $F_j(k) = \{w : \|w_j - x_j(t_k)\| = VH_k\}$, the tCRH algorithm converges in a finite time to a feasible solution and the solution guarantees that all targets are eventually visited by at least one vehicle.

Proof: *Theorem* 2 assures that, if each vehicle j could move to a whatever position $x_j \in \mathbf{R}^2$, then, minimizing J_j of equation (3.5), it would choose a target (indifferently among the N available targets). The condition for vehicle j to move to a target is that at least one target belongs to the vehicle j admissible solution space (as it occurs if the admissible space is $\mathbf{R}^2$). If at each timestep t_k at least one vehicle minimizes J_j of equation (3.5) over an admissible space which contains at least one target, then at each timestep at least one target is guaranteed to be visited. The definition of $F_j(k)$ satisfies this condition: since H_k, at each timestep t_k, is chosen as the minimum time, among all vehicles, to visit a target, then at least one vehicle is able to visit a target during the H_k period of time (i.e., at each timestep). As the number of targets is finite, then the algorithm guarantees that all targets are visited in a finite amount of time. ■

3.1.3 *Analysis of the limitations of the tCRH approach*

Section 2.3 illustrated the limitations of the CRH approach, i.e., (i) the oscillatory behavior in some scenario configurations, and (ii) the poor performances in presence of clusters of targets.

The tCRH approach gives better results in presence of clusters of targets, since each target is capable of attracting more than one vehicle. This behavior is well illustrated in Figure 3, which shows the initial configuration of 10 randomly positioned vehicles and 40 clustered targets (Figure 3a)) and the vehicle trajectories performed in 25 steps of simulation (Figure 3 b)). The simulation parameters are $\Delta = 0.49$, $s = 0.25$, $r = 2$, h_k as defined in equation (4.1), $R_i = 100$, for $i \in \{1,2,\ldots,30\}$ and $V_j = 2$, for $j \in \{1,2,\ldots,10\}$. The cluster will be reached by all the targets and the algorithm efficiency is increased, also in case of few vehicles, compared with the CRH approach (see Section 4 for quantitative simulation results).

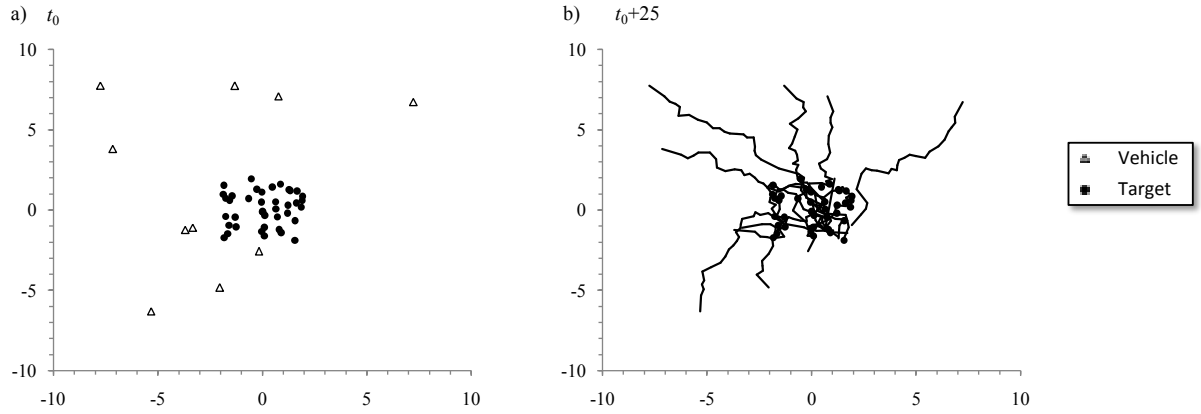


Figure 3: Example of scenario with clustered targets: tCRH produces good performances.

The tCRH approach, moreover, may provide scarce results if the vehicles are clustered and the targets are relatively sparse. Figure 4 a) shows a configuration in which 5 vehicles are grouped in one cluster and 20 targets are grouped in two clusters **A** and **B**: A is the closest cluster of targets to the set of vehicles. The simulation parameters are the same of the previous example. As illustrated in Figure 4 b) and Figure 4 c), the vehicles show a "greedy" behavior, i.e., all the vehicles firstly head to the first cluster of targets and then, after that all those targets are covered, head to the second cluster of targets. This behavior could be inefficient in particular configurations characterized by concentrated vehicles and, moreover, the algorithm does not provide an almost uniform coverage of the mission space, differently from the CRH case. See Section 4 for complete simulation results, showing the comparison among the different approaches.

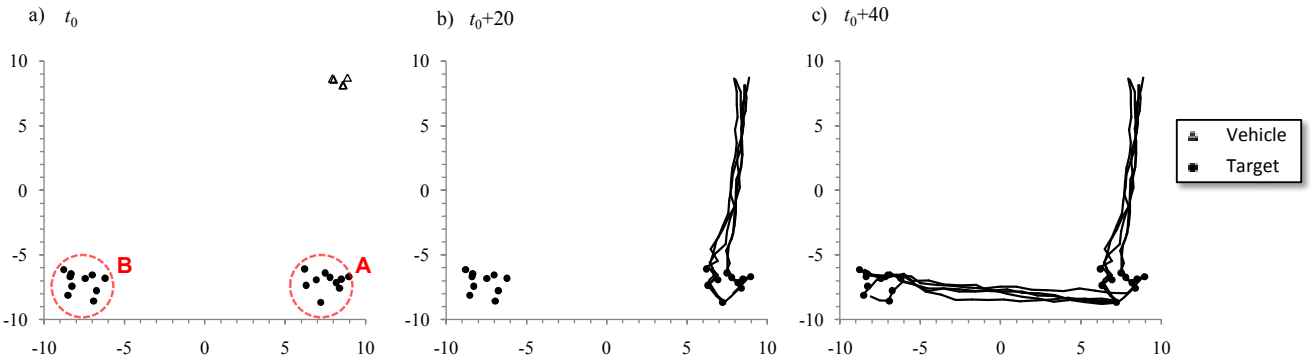


Figure 4: Example of scenario with clustered vehicles and targets: tCRH greedy behavior.

3.2 MixedCRH (mCRH)

As previously analysed, the performances of the two approaches to the dynamic vehicle routing are efficient/inefficient in different scenario configurations:

- the CRH algorithm gives better results if the targets and vehicles are sparse and in stochastic scenarios (i.e., the targets appear stochastically), where the vehicles which are not assigned to any targets are free to perform some exploration of the mission space;
- if the targets are someway clustered, and in deterministic scenarios (the targets are already known at the mission start), the CRH algorithm performances are low since it may happen that only few vehicles are assigned to a large amount of target, while the other vehicles uselessly explore the mission space;
- the tCRH algorithm gives better results if the targets are clustered and in deterministic scenarios, since each target is capable of attracting a vehicle, and the exploration of the mission space is not important;
- if the targets are sparse and in stochastic scenarios, the tCRH algorithm performances are low since it may happen that a single target attracts more than one vehicle and only few vehicles explore the mission space.

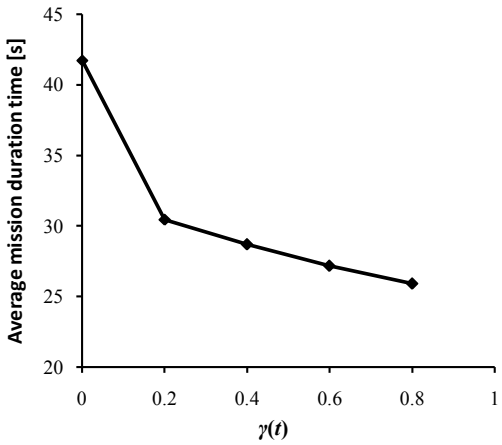
Thus, the idea is to use the CRH algorithm or the tCRH algorithm depending on current mission conditions, i.e., to adapt the algorithm behavior to the mission current situation. To this extent, the objective functions (2.11) and (3.3) of the two problems are simultaneously used in a multi-objective optimization fashion:

$$J_{mCRH}(\mathbf{u}_k, t_k) = \gamma(t_k) J_{CRH}(\mathbf{u}_k, t_k) + [1 - \gamma(t_k)] J_{tCRH}(\mathbf{u}_k, t_k) \tag{3.9}$$

where the real-valued parameter $\gamma(t_k) \in [0,1]$ determines whether the current algorithm behavior is more like the CRH ($\gamma(t_k) = 1$) or the tCRH ($\gamma(t_k) = 0$).

Simulations were performed in two particular scenarios, aimed at highlighting the different impact of the two behaviors. In the first scenario, 10 vehicles, are positioned in a cluster in the center of the mission space (a square with sides of length 20m), and 20 targets are randomly positioned on the mission space. The second scenario is characterized by the presence of three randomly positioned vehicles and 30 clustered targets, in the center of the mission space. The simulation parameters are $\Delta = 0.4$, $s = 0.25$, $r = 2$, $R_i = 1$, $\beta_i = 5$, $D_i = 100$, for $i \in \{1,2,\ldots,30\}$ and $V_j = 0.5$, for $j \in \{1,2,3\}$ or $j \in \{1,2,..,10\}$. As illustrated in Figure 5, large values of $\gamma(t_k)$ (i.e., the CRH behavior) are preferable in scenarios in which the vehicles are clustered, in order to increase the exploration of the mission space; on the contrary, small values of $\gamma(t_k)$ (i.e., the tCRH behavior) are preferable in scenarios in which the targets are clustered, to make the vehicles jointly head to the targets. The results related to the value $\gamma(t_k) = 1$ are not presented since oscillations verify and, thus, the average duration time is the highest in both scenarios.

a) *Clustered vehicles, random targets*



b) *Clustered targets, random vehicles*

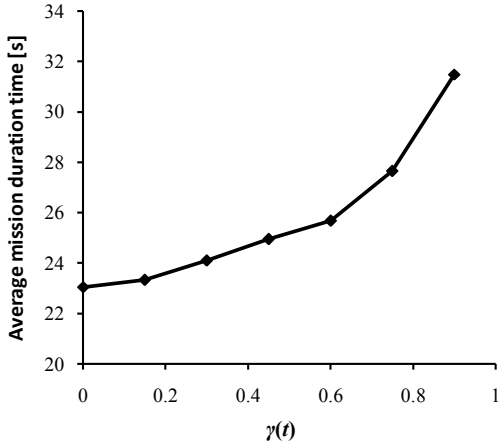


Figure 5: Mixed CRH (mCRH): mission duration time against variable weight.

On the ground of the shown example and of many other simulation results, the *mixed CRH* algorithm, hereafter referred to as mCRH, was defined by setting $\gamma(t_k) = \gamma = 0.5$, for each t_k.

3.3 Adaptive CRH (aCRH)

As shown in the previous Section, the two CRH and tCRH approaches seem to be complementary, in the sense that the CRH algorithm makes the vehicles explore the mission space and the tCRH makes the vehicles head directly to the targets. The aim of this Section is then to introduce an adaptive approach able to exploit both the approaches, according to the current targets and vehicles configurations.

Section 2.3 showed that the CRH approach strongly suffers the presence of a conspicuous number of targets surrounding a vehicle. As clearly exemplified by Figure 1 with reference to a triangle targets configuration, the (weighted) center of gravity of the targets included in the responsibility region of a vehicle, formally defined as:

$$c_j = \frac{\sum_{i \in S_j} R_i y_i}{\sum_{i \in S_j} R_i}, \tag{3.10}$$

appears to be a critical point of the mission space: if a vehicle is situated close to this point, it starts oscillating. The first idea is to make the algorithm be aware of the existence of this critical point and act as a consequence. The idea of taking into account the critical point allows the algorithm to detect a possible point which could generate oscillations and to act to handle this situation, by switching to the tCRH approach. The algorithm is then aimed at setting $\gamma(t_k)$ to a small value γ_0 if the distance between a vehicle position x_j and its center of gravity c_j is below a given threshold c, corresponding to favouring the tCRH 'greedy' behaviour, which allows vehicles to head directly to the targets, and at setting $\gamma(t_k)$ to a large value γ_1 otherwise, corresponding to favouring the CRH behaviour.

The second idea aims at increasing the performances of the algorithm, especially in the case of clustered vehicles, when the tCRH produces poor results, by designing a vehicle routing control strategy based on two principal steps: (i) if the vehicles are grouped, use a large value of $\gamma(t_k)$, i.e., give more importance to the CRH behavior, in order to make the vehicles explore the mission space and distribute onto the mission space; (ii) when the vehicles are sparse on the mission space, use a small value of $\gamma(t_k)$, i.e., give more importance to

the tCRH and let the vehicles directly head to the closest targets. This approach is a sort of *divide et impera* strategy: once the vehicles are correctly positioned onto the mission space, let the vehicles directly head to the targets. The main information that must be known, in the most precise way possible, is when to declare that the vehicles are sufficiently sparse.

In conclusion, the adaptive algorithm behavior is as follows:

$$\gamma(t_k) = \begin{cases} \gamma_0, \textbf{ if } \forall j \in A(t_k), \exists i \in T(t_k) : j \in B_i^2(t_k) \wedge i \in \overline{B}_j^2(t_k) \textbf{ or } \exists j : \left\| x_j - c_j \right\| < c \\ \gamma_1, \qquad \textbf{otherwise} \end{cases}, \tag{3.11}$$

Equation (3.11) states that:

1. $\gamma(t_k)$ is set to a small value γ_0, at time t_k, if one of the following conditions hold:
   - the Euclidean distance between a vehicle j and the related center of gravity c_j is less than c;
   - for all vehicles j, there exists a target i for which the vehicle j is the closest vehicle in $B_i^2(t_k)$ and the target i is the closest target in $\overline{B}_j^2(t_k)$;
2. otherwise $\gamma(t_k) = \gamma_1$, where γ_1 is close to 1, to favour the exploring behaviour of the CRH algorithm.

The parameter $\gamma(t_k)$ is a dynamic factor, since it is computed at each time step t_k. The algorithm is adaptive in the sense that it is able to dynamically analyze the current vehicles positions to switch between the "exploring" CRH and the "greedy" tCRH strategies.

Section 4 evaluates this *adaptive CRH* algorithm, referred to as aCRH.

4 Simulations

The algorithms illustrated in Section 2 and Section 3 were implemented in MATLAB using the *pattern search* algorithm for the solution of non-linear optimization problems (see [HOOK61] for further details). The algorithms illustrated in Section 2 and Section 3 were tested indifferent configurations, characterized by different placements of targets and vehicles in the mission space. The mission space is a square with sides of length 20m. The simulation parameters are constant, unless differently specified, and shown in the following table:

Attribute	Value
R_i, $i \in \{1,2,...,20\}$	100
α_i, $i \in \{1,2,...,20\}$	0.5
β_i, $i \in \{1,2,...,20\}$	0.1
D_i, $i \in \{1,2,...,20\}$	100
s_i, $i \in \{1,2,...,20\}$	0.25
V_j, $j \in \{1,2,...,10\}$	2
b	2
Δ	0.49
r	2
γ_0	0
γ_1	0.9
c	1

Table 1: Simulations parameters

All the vehicles $j \in \{1,2,...,10\}$ are moving at constant velocity $V_j = V = 1$ ms^{-1}. All targets are characterized by the same fixed parameters, as defined in Table 1. s_i (equal, in the simulations, to a fixed value s for each target i) represents the i-th target size (the target i-th is considered captured by j-th vehicle at time t if $\|x_j(t) - y_i(t)\| \leq s_i$). The planning horizon H_k is given by (2.6) and the action horizon h_k is set as follows:

$$h_k = \begin{cases} H_k, \text{ if } H_k \leq s \\ 0.5, \text{ otherwise} \end{cases} \tag{4.1}$$

The action horizon is set, if the vehicles are not close to any target, to a small value in order (i) to rapidly adapt to changes in the scenario (e.g. in the dynamic of the unknown scenarios) and (ii) to be reliable against eventual errors committed by the non-linear optimization solver used in the simulation.

The algorithm was simulated in the following scenarios, characterized by different targets and vehicles positions in the mission space:

1. Targets and vehicles randomly distributed throughout the mission space (Section 4.1).
2. Targets randomly distributed throughout the mission space and clustered vehicles (Section 4.2).
3. Targets grouped in two clusters and vehicles grouped in one cluster (Section 4.3).
4. Targets placed on a circle of radius 8m and vehicles concentrated in its centre (Section 4.4).
5. Targets and vehicles randomly distributed in the mission space, with 5 targets whose positions are unknown to the vehicles (Section 4.5).
6. Dynamic scenario (Section 4.6)

Each scenario was simulated by means of 25 simulation runs with different seeds. The algorithms proposed in this paper were simulated: the CRH (see Section 2), the tCRH (see Section 3.1), the mCRH (see Section 3.2) and the aCRH (see Section 3.3). In some of the simulations, the results of the CRH algorithm are not shown, since the amount of oscillations during the simulation was so high that the mission was rarely successfully accomplished. For each simulated scenario and algorithm, the average and the standard deviation values of mission duration time are computed and reported to evaluate the performances of the proposed algorithms.

4.1 Random distribution of targets and vehicles

The scenario of this simulation is characterized by a random distribution of 20 targets and 10 vehicles throughout the mission space (a square with sides of length 20m). The following figure shows an example of random distribution of both vehicles and targets.

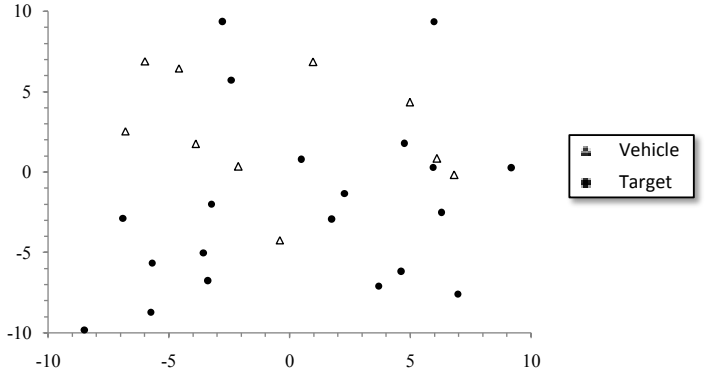


Figure 6: Random distribution of vehicles and targets: example scenario.

The average values of the mission duration time are reported in the following figure, for each algorithm presented in this paper.

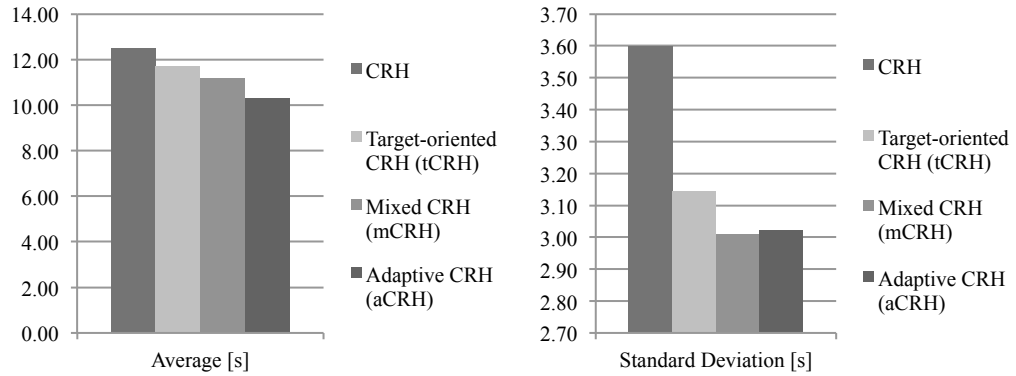


Figure 7: Random distribution of vehicles and targets: mission duration time (average and standard deviation), in seconds.

The figure shows that similar results are obtained by all the algorithms. In this scenario, the CRH does not produce oscillations in the vehicles trajectories in the majority of simulation runs, since the *number of vehicles is comparable with the number of targets* and everything is fairly distributed in the mission space. The tCRH makes the vehicles head to the closest targets as both are randomly distributed. The mCRH and the aCRH exploit the advantages of both the algorithms and the aCRH achieves the best results (the decrease in mission duration time, compared to the CRH approach, is about 17%). Regarding the standard deviation results, similar considerations hold: the standard deviation of the CRH is the largest since, sometimes, small oscillations happen, depending on the particular configuration scenario.

4.2 Random distribution of targets and clustered vehicles

The scenario of this simulation is characterized by a random distribution of 30 targets on the mission space and 3 vehicles starts from the top left of the mission space (they are clustered). The following figure shows an example of random distribution of both vehicles and targets.

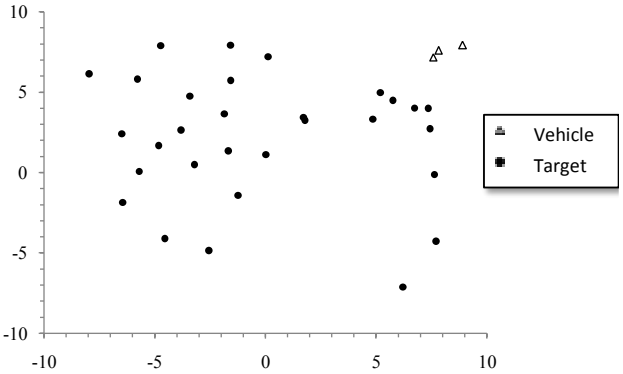


Figure 8: Random distribution of targets and clustered vehicles: example scenario.

The average values of the mission duration time are reported in the following figure, for each algorithm presented in this paper.

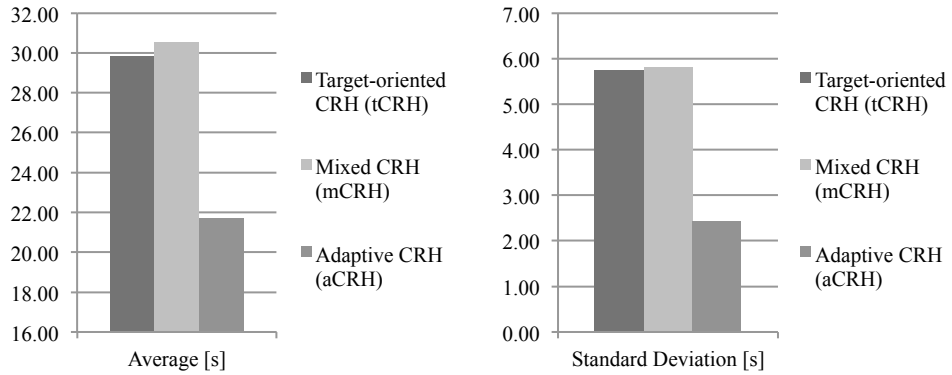


Figure 9: Random distribution of targets and clustered vehicles: mission duration time (average and standard deviation), in seconds.

The CRH results are not shown because in most of the simulation runs the CRH algorithm fails to accomplish the targets covering within the mission time. Figure 9 confirms, as expected, that the tCRH average mission duration time is large. In fact, the tCRH is characterized by a "greedy" behaviour and, since the vehicles are initially clustered, they follow almost the same trajectory. An exploration behaviour must be introduced in the algorithm to achieve better results. The mCRH is not able to achieve better results since oscillations verify when the algorithm sets $\gamma = \gamma_1$, with the effect of delaying the mission accomplishment. Much better results are instead obtained by using the aCRH. With the aCRH algorithm, the vehicles initially explore the mission space, since the algorithms sets $\gamma = \gamma_1$, and they aim at the closest targets with a 'greedy' behaviour as the aCRH sets $\gamma = \gamma_0$. The average mission duration time is decreased, compared to the tCRH, of about 27%. Moreover, the standard deviation results of Figure 9 show that the adaptive strategy is able to produce more consistent results, compared to the tCRH and the mCRH.

4.3 One cluster of targets and random vehicles

The scenario of this simulation is characterized by (i) a set of 30 targets concentrated within a 4m-by-4m rectangle positioned in the center of the mission space and (ii) 3 randomly positioned vehicles. Figure 10 shows an instance of the simulated scenario:

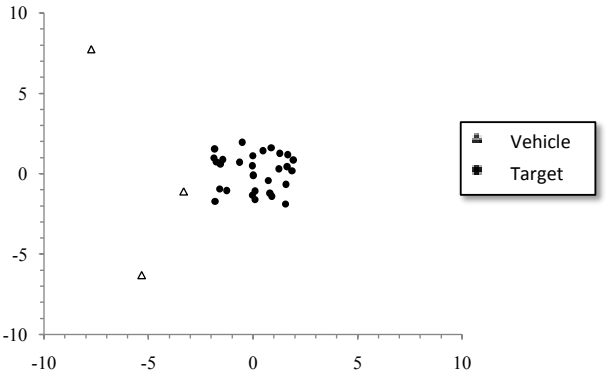


Figure 10: One cluster of targets and randomly positioned vehicles: example scenario.

The average values of the mission duration time are reported in Figure 11, for each algorithm presented in this paper.

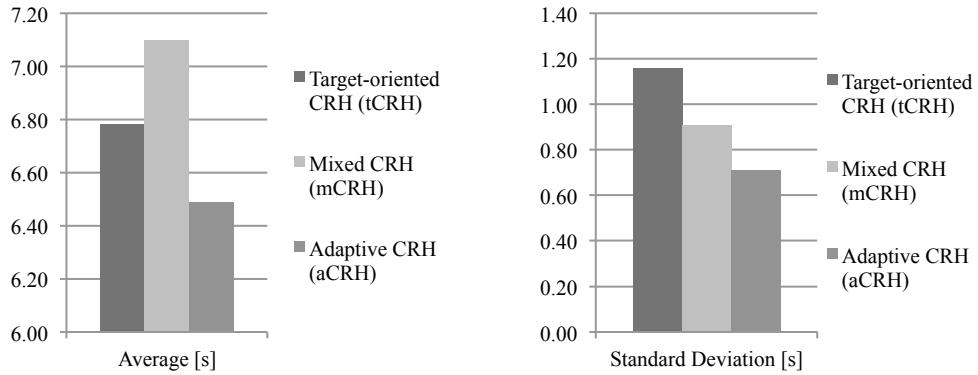


Figure 11: One cluster of targets and randomly positioned vehicles: mission duration time (average and standard deviation), in seconds.

Figure 11 clearly illustrates that, in the presence of clustered targets but random vehicles, the adaptive aCRH strategy achieves better results with respect to the tCRH one, and that the mCRH algorithm is the worst one. Since the targets are clustered, no exploration is needed and the greedy component of the three approaches guarantees a fast convergence. The mCRH suffered from small oscillations occurring as the vehicle reach the target cluster.

4.4 Two clusters of targets and one cluster of vehicles

The scenario of this simulation is characterized by (i) a set of 3 vehicles placed in the top right of the mission space and (ii) two clusters of 15 targets, located in the bottom right and the bottom left of the mission space. The clusters of targets are distributed in 2m-by-2m squares. The following figure shows an instance of the simulated scenario:

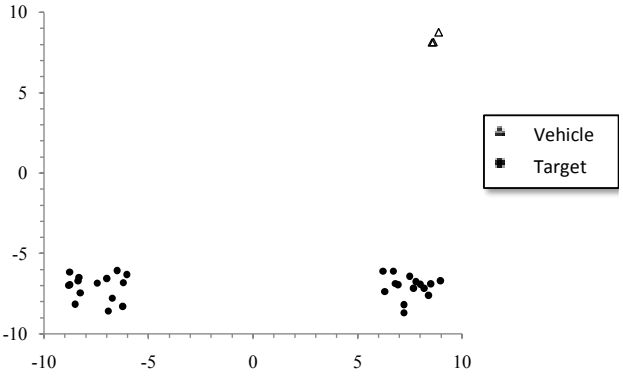


Figure 12: Two clusters of targets and one cluster of vehicles: example scenario.

The average values of the mission duration time are reported in Figure 13, for each algorithm presented in this paper.

In presence of two clusters of targets and one cluster of vehicles, the best result is again obtained by the aCRH (the average mission duration time decrease is of about 11% with respect to the tCRH): in this simulations, initially the algorithm sets a exploring behaviour achieving a repartition of the vehicles among the targets, and it sets a greedy behaviour which lead the vehicles to the targets. As predictable, the worst results are achieved by the tCRH, which makes the vehicles jointly head to the first cluster of targets and then to the second one. The mixed strategy produces intermediate results, again due to small oscillations occurring as the vehicles approach the target clusters.

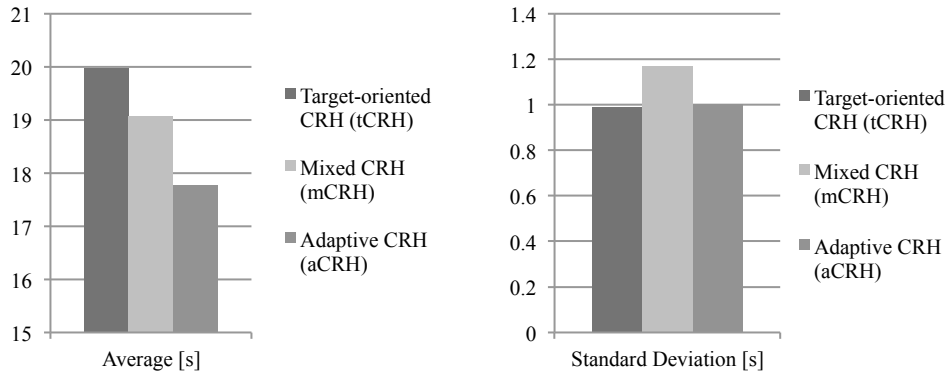


Figure 13: Two clusters of targets and one cluster of vehicles: mission duration time (average and standard deviation), in seconds.

4.5 Targets in circle

The scenario of this simulation is characterized by (i) a set of 30 targets placed on a circle of radius 8m, and (ii) M vehicles concentrated in a region close to its center. The following figure shows an instance of the simulated scenario with $M = 3$.

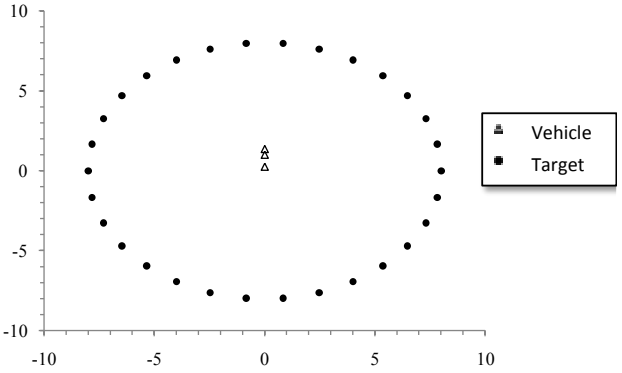


Figure 14: Targets in circle, vehicles close to the center: example scenario with $M = 3$.

The average values of the mission duration time are reported in Figure 15, for each algorithm presented in this paper, for $M = \{3,6\}$. Figure 15 (a) shows that, in case of a small set of vehicles, the best results are produced by the tCRH approach: the algorithm is often able to lead each vehicle to a different part of the circle of targets. However, the standard deviation of the tCRH is the highest because, sometimes, all the vehicles follow the same path, without any mission space division, whereas the aCRH performance are close to the tCRH one in terms of average mission time and better in terms of standard deviation. The mCRH performance is the worst, since the symmetry of the scenario generates several oscillations. In case of six vehicles, the tCRH becomes the worst (the average mission duration time is almost the same of Figure 15(a)), since it is unable to take advantage to the larger set of vehicles and chooses almost the same path for all the vehicles. In this case, the mCRH is the best one since the larger number of vehicles reduces oscillations: the aCRH, anyway, is able to produce results very similar to the mCRH, showing the capability of the algorithm to adapt to both configurations.

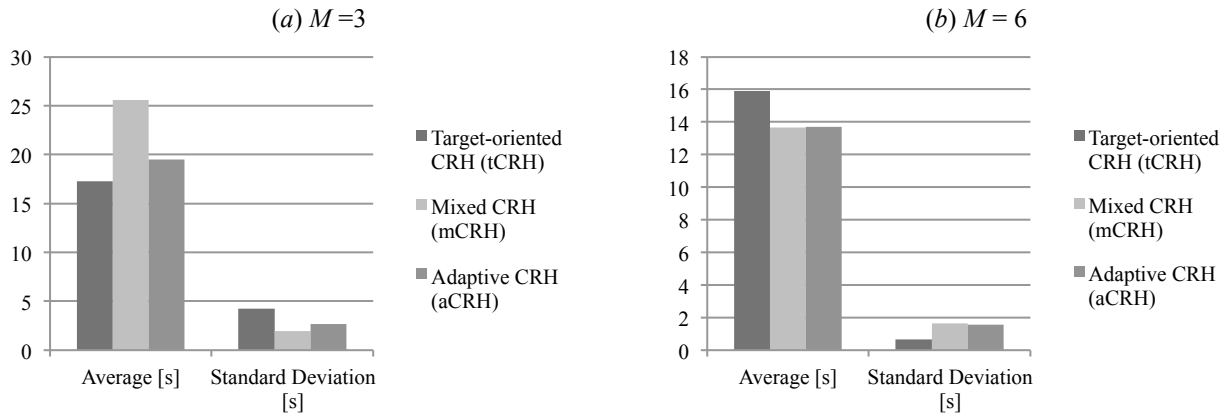


Figure 15: Targets in circle, vehicles close to the center: mission duration time (average and standard deviation), in seconds.

4.6 Unknown targets

The scenario of this simulation is characterized by 20 targets and 10 vehicles which are randomly distributed in the mission space, with 10 targets in unknown positions: the targets may be detected by a vehicle if it is close enough, in the so-called "sensor detection area". The sensor detection area was chosen as a circle of radius 3.333m. Figure 16 shows an instance of the simulated scenario.

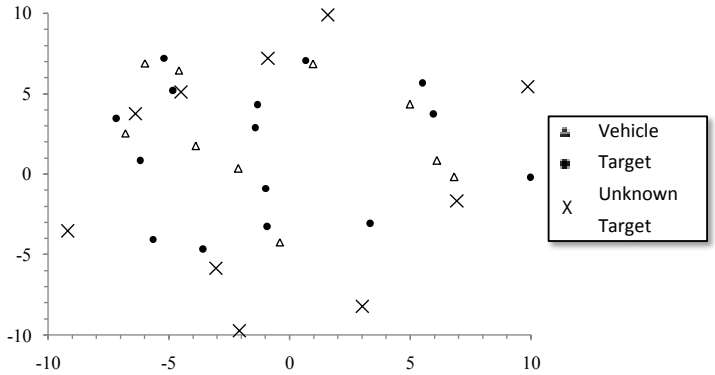


Figure 16: Unknown targets: example scenario.

The average values of the mission duration time are reported in the following figure, for each algorithm presented in this paper.

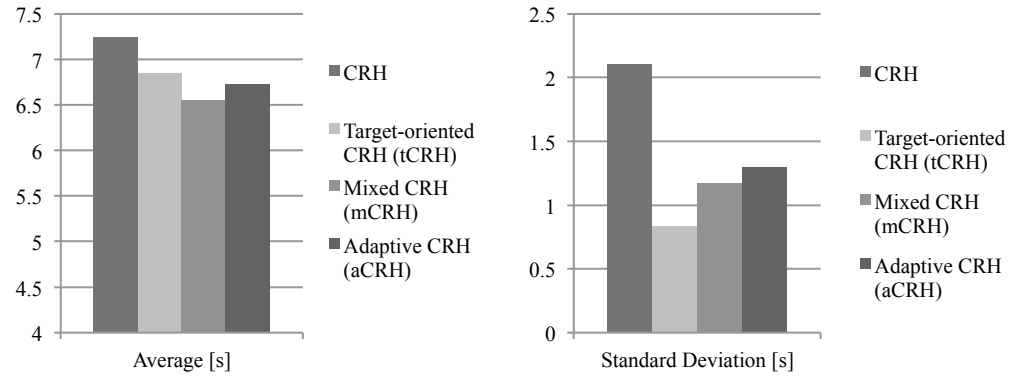


Figure 17: Unknown targets: mission duration time (average and standard deviation), in seconds.

Similarly to the simulation of Section 4.1, the results of the different algorithms are similar (between 6.5s and 7.2s), since the vehicles are already distributed on the mission space and the number of the vehicles is high (at least comparable to the number of targets, which reduces the occurrences of the oscillating behaviour). The 10 unknown targets are discovered and covered in the 100% of the mCRH and aCRH algorithms simulations and in the 96% of the CRH and the tCRH simulations.

4.7 Dynamic scenario

This final scenario aims at demonstrating that the adaptive approach is able to outperform the tCRH and the mCRH in a time-varying environment. The variability consists in the appearance of new sub-sets of targets, appearing in different configurations. The transition times and related configurations are detailed in the following table. The number of vehicles is fixed, $M = 3$, and they are initially randomly deployed on the mission space.

Time step	Number of new targets	Configuration of new targets
$t_0 = 0$s	15	Random targets
$t_1 = 6.5$s	10	One cluster of targets in the top right of the mission space.
$t_2 = 15$s	10	Two clusters of targets, similarly to Figure 12

Table 2: Dynamic scenario: list of events.

The average values of the mission duration time are reported in the following figure, for each algorithm presented in this paper.

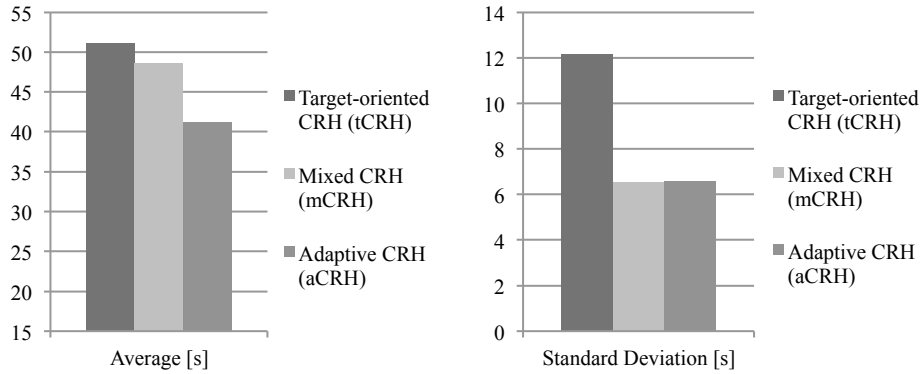


Figure 18: Dynamic scenario: mission duration time (average and standard deviation), in seconds.

The previous figure shows that the aCRH produces very the best results, especially regarding the average mission duration time (the performances enhancement, compared to tCRH, is of about 15%). The mCRH is still affected by oscillation paths. Finally, the standard deviation results are the worst for the tCRH, since the "greedy" approach could favour certain random configuration compared to other ones.

5 Conclusions and Future Work

This paper introduced an innovative cooperative vehicle routing algorithm with the aim of covering a set of targets distributed onto the mission space. The approach is to dynamically re-compute the routing by means

of a receding horizon strategy, able to adapt to changes in the environment configuration (for instance in case of unknown targets and possible new targets appearance). The work developed in [LI06] has been studied: it guarantees the coordination of the agents without any ad-hoc vehicle-target assignment and is capable to uniformly explore the mission space in order to cover the regions belonging to the mission space and rapidly cover eventual unknown targets. The weaknesses of this approach were identified, in particular concerning vehicle oscillations and poor performance in specific, but common, scenario configurations (e.g. clustered). A new strategy has been proposed, which is able to guarantee convergence in the sense that all targets will be eventually covered, but it produces poor performances in sparse or un-clustered environments. The two strategies have been combined, in an adaptive way, to take advantage of the good aspects of both and simulation results shown the goodness of the proposed approach.

The authors are willing to develop further and interesting extensions or integrations of the proposed approach. research directions under investigation are i) dynamic clustering of the targets to better distribute the workload among the vehicles; ii) different metrics for the dynamic partitioning to account for the spatial density of the targets on the mission space; iii) distributed approach with possible partial communications among vehicles (e.g., information about the positions can be shared only among dynamic sub-sets of vehicles); iv) new 'local' rules for the adaptive algorithm, i.e., the value of the parameter gamma in equation (3.11) may be different for the different vehicles, depending on their position in the mission space. On another research perspective, the authors are working on a discrete version of the adaptive algorithm presented in this paper, deployable in real urban environments, using a proper distance function different from the Euclidean distance, and simulated using city maps; a distributed discrete algorithm may be envisaged for a solution to be used in real urban and communication-constrained environments.

References

[AURE91] Aurenhammer F. (1991).“*Voronoi diagrams – a survey of a fundamental geometric data structure*”. ACM Comput. Surv. 23, 3 (September 1991), 345-405. DOI=10.1145/116873.116880 http://doi.acm.org/10.1145/116873.116880.

[BALD07] Baldacci R., Toth P., and Vigo D., (2007). “*Recent advances in vehicle routing exact algorithms*”. 4OR: A Quarterly Journal of Operations Research, 5 (4): 269-298.

[BALD08] Baldacci R., Christofides N., Mingozzi A., (2008). “*An exact algorithm for the vehicle routing problem based on the set partitioning formulation with additional cuts*”. Math. Program., Ser. A (2008) 115: 351-385, DOI: 10.1007/s10107-007-0178-5.

[BERT88] Bertsimas D., (1988). “*Probabilistic combinatorial optimization problems*”. PhD thesis, Massachussetts Institute of Technology, Dept. ofMathematics.

[BERT92] Bertsimas D. J., Van Ryzin G., (1992). “*Stochastic and dynamic vehicle routing in the Euclidean plane with multiple capacitated vehicles*”, Operations Research 41 (1): 60-76.

[BRAN05] BrankeJ., MiddendorfM., Noeth G., Dessouky M., (2005). “*Waiting strategies for dynamic vehicle routing*”. Transportation Science, 39 (3): 298-312.

[BRAN09] Branchini R., Armentano V. A., Lokketangen A., (2009). “*Adaptive granular local search heuristic for a dynamic vehicle routing problem*”. Computer&Operations Research, 36 (11): 2955-2968.

[CASS03] G. Cassandras, Wei Li, “*A receding horizon approach for solving some cooperative control problems*”, Proceedings of the 41st IEEE Conference on Decision and Control, Las Vegas, Nevada USA, December 2002.

[CASS10] Yao C., Ding C., Cassandras C. G., (2010). “*Cooperative Receding Horizon Control for Multi-agent Rendezvous Problems in Uncertain Environments*”, 49th IEEE Conference on Decision and Control, December 15-17, 2010.

[CASS11] Cassandras, C.G., and Paschalidis, I.C., (2011). “*Optimizing the Transportation System' Response Capabilities*”, DHS 2011 Science Conference - Fifth Annual University Network Summit, March 2011.

[CHEN06] Chen Z., Xu H., (2006). "*Dynamic column generation for dynamic vehicle routing with time windows*". Transportation Science, 40 (1): 74-88.

[CHRI07] Christiansen C., Lysgaard J., (2007). "*A branch-and-price algorithm for the capacitated vehicle routing problem with stochastic demands*". Operations Research Letters, 35 (6): 773-781.

[CORD07] Cordeau J. F., Laporte G., Savelsbergh M. W., Vigo D., (2007). "*Vehicle routing*". In Barnhart, C., and Laporte G., editors, Transportation, volume 14 of handbooks in Operations Research and Management Science, chapter 6, pages 367-428. Elsevier.

[DANT59] Dantzig G., Ramser J., (1959). "*The truck dispatching problem*", Management Science, 6 (1): 80-91.

[DROR89] Dror M., Laporte G., Trudeau P., (1989). "*Vehicle Routing with stochastic demands: Properties and solution frameworks*". Transportation Science, 23 (3): 166-176.

[GEND01] Gendreau M., Laporte G., Semet F., (2001). "*A dynamic model and parallel tabu search heuristic for real-time ambulance relocation*". Parallel Computing, 27 (12): 1641-1653.

[GEND99] Gendreau M., Guertin F., Potvin J.-Y., Taillard E., (1999). "*Parallel tabu-search for real-time vehicle routing and dispatching*", Transportation Science, 33 (4): 381-390.

[HAGH07] Haghani A., Yang S., (2007). "*Real-time emergency response fleet deployment: Concepts, systems, simulation & case studies*". In Zeimpekis V., Tarantilis C. D., Giaglis G. M., Minis I. editors, Dynamic Fleet Management, volume 38 of Operations Research/Computer Science Interfaces, pages 133-162. Springer US.

[HOOK61] Hooke, R., Jeeves, T.A. (1961). *"Direct search" solution of numerical and statistical problems"*. Journal of the Association for Computing Machinery(ACM)8 (2): 212–229.

[HSUE08] Hsueh C. F., Chen H. K., Chou H. W., (2008). "*Dynamic Vehicle Routing for Relief Logistics in Natural Disasters*". Vehicle Routing Problem, edited by TonciCaric and Hrvoje Gold, pp. 71-84, September 2008.

[KENY03] Kenyon A. S., and Morton D., (2003). "*Stochastic vehicle routing with random travel times*". Transportation Science, 37 (1): 69.

[LAPO02] Laporte G., Louveaux F. Mercure H., (2002). "*An integer L-shaped algorithm for the capacitated vehicle routing problem with stochastic demands*". Operation Research, 50 (3): 415-423.

[LAPO07]Laporte G., (2007). "*What you should know about the vehicle routingproblem*". Naval Research Logistics, 54 (8):811-819.

[LAPO09] Laporte G., (2009). "*Fifty years of vehicle routing*". Transportation Science, 43 (4): 408-416.

[LAPO92] Laporte G., Louveaux F., Mercure H., (1992). "*The vehicle routing problem with stochastic travel times*". Transportation Science, 26 (3): 161-170.

[LARS01] Larsen A., (2001). "*The Dynamic Vehicle Routing Problem*". PhD thesis, Technical University of Denmark (DTU).

[LI05] Li W., Cassandras C. G., (2005). "*Cooperative Distributed Control of Networks with Mobile Nodes: Theory and Practice*". Proceedings of 2005 Allerton Conference, September 2005.

[LI06] Li W., Cassandras C. G., (2006). "*A Cooperative Receding Horizon Controller for Multivehicle Uncertain Environments*". IEEE Transactions on Automatic Control, 51(2): 242-257.

[MEND10] Mendoza J. E., Castanier B., Guèret C., Medaglia A. L., Velasco N., (2010). "*A memetic algorithm for the multi-compartment vehicle routing problem with stochastic demands*". Computers & Operations Research, 37 (11): 1886-1898.

[MEND11] Mendoza J. E., Castanier B., Guèret C., Medaglia A. L., Velasco N., (2011). "*Constructive Heuristics for the multi-compartment vehicle routing problem with stochastic demands*". Transportation Science, 45 (3): 346-363.

[Nall09] Nallusamy R., Duraiswamy K., Dhanalaksmi R., Parthiban P., (2009)."*Optimization of multiple vehicle routing problems using approximation algorithms*". International Journal of Engineering Science and Technology, vol. 1 (3), 2009,129-135.

[PILL11] Pillac V., Gendreau M., Christelle G., Medaglia A. L., (2011)."*A Review of dynamic Vehicle Routing Problems*", CIRRELT-2011-62.

[ROBE12] Roberti R.,(2012). "*Exact Algorithms for Different Classes of Vehicle Routing Problems*". PhD Thesis, Alma Mater Studiorum- University of Bologna.

[SECO00] Secomandi N., (2000). "*Comparing neuro-dynamic programming algorithms for the vehicle routing problem with stochastic demands*". Computers & Operations Research, 27 (11-12): 1201-1225.

[SECO09] Secomandi N., (2009). "*Reoptimization approaches for the vehicle-routing problem with stochastic demands*". Operation Research, 57 (1): 214-230.

[TOTH02] Toth P., Vigo D., (2002). "*The vehicle routing problem*", volume 9 of SIAM Monograph on Discrete Mathematics. SIAM Philadelphia.

[VanH06] Van Hentenryck P., Bent R., (2006). "*Online stochastic combinatorial optimization*". MIT Press.

[VERW03] Verweij B., Ahmed S., Kleywegt A., Nemhauser G., Shapiro A., (2003). "*The sample average approximation method applied to stochastic routing problems: a computational study*". Computational Optimization and Applications, 24 (2): 289-333.

[WAT89] Waters C., (1989). "*Vehicle-scheduling problems with uncertainty and omitted customer*". The Journal of the Operational Research Society, 40 (12riz): 1099-1108.

Appendix

Proof of Lemma 2: The proof is divided in two parts: (1) the first part demonstrates that only one stationary point exists for $J_j = q_{1j}\|x_j - y_1\| + q_{2j}\|x_j - y_2\|$ and the point is the midpoint $\bar{x}_j = (y_1 + y_2)/2$ of the segment between y_1 and y_2; (2) the second part proves that the point is a saddle point. The objective function J_j can be written in the following way, by using the explicit expression of q_{ij} in equation (3.5).

$$J_j = \frac{1}{1-2\Delta}\left[(1-\Delta)\|x_j - y_1\| - \frac{\|x_j - y_1\|^2}{\|x_j - y_1\| + \|x_j - y_2\|}\right] + \frac{1}{1-2\Delta}\left[(1-\Delta)\|x_j - y_2\| - \frac{\|x_j - y_2\|^2}{\|x_j - y_1\| + \|x_j - y_2\|}\right], \quad \text{(A.1)}$$

(1) A point x_j is a stationary point of J_j in (A.1), if $\frac{\partial J_j}{\partial x_j} = 0$. To compute $\frac{\partial J_j}{\partial x_j}$, the easiest way is to use the chain rule: let define two functions of the x_j variable:

$$\begin{cases} f(x_j) = \|x_j - y_1\| \\ g(x_j) = \|x_j - y_2\| \end{cases}$$

Using those functions, the stationary point condition can be written as:

$$\frac{\partial J_j}{\partial x_j} = \frac{\partial J_j}{\partial f}\frac{\partial f}{\partial x_j} + \frac{\partial J_j}{\partial g}\frac{\partial g}{\partial x_j} = 0, \quad \text{(A.2)}$$

With such a variable substitution in (A.1), the objective function J_j can be rewritten as:

$$J_j = \frac{1}{1-2\Delta}\left[(1-\Delta)f - \frac{f^2}{f+g}\right] + \frac{1}{1-2\Delta}\left[(1-\Delta)g - \frac{g^2}{f+g}\right]$$

Let compute the $\frac{\partial J_j}{\partial f}$ and $\frac{\partial J_j}{\partial g}$ terms:

$$\frac{\partial J_j}{\partial f} = \frac{1}{1-2\Delta}\left[(1-\Delta) - \frac{2f(f+g) - f^2}{(f+g)^2} + \frac{g^2}{(f+g)^2}\right]$$

$$\frac{\partial J_j}{\partial g} = \frac{1}{1-2\Delta}\left[\frac{f^2}{(f+g)^2} + (1-\Delta) - \frac{2g(f+g) - g^2}{(f+g)^2}\right]$$

Moreover, $\frac{\partial f}{\partial x_j} = \frac{x_j - y_1}{\|x_j - y_1\|}$ and $\frac{\partial g}{\partial x_j} = \frac{x_j - y_2}{\|x_j - y_2\|}$ represent the versors (with module 1) connecting x_j and y_1, y_2 respectively. From equation (A.2), it follows that:

$$\frac{\partial J_j}{\partial f}\frac{\partial f}{\partial x_j} = -\frac{\partial J_j}{\partial g}\frac{\partial g}{\partial x_j}, \quad \text{(A.3)}$$

which represents an equality relationship between two vectors. Two vectors are equal if and only if they have the same module and the same direction. Regarding the module, since the $\frac{\partial f}{\partial x_j} = \frac{x_j - y_1}{\|x_j - y_1\|}$ and

$\frac{\partial g}{\partial x_j} = \frac{x_j - y_2}{\|x_j - y_2\|}$ factors are versors, then $\frac{\partial J_j}{\partial f}$ and $\frac{\partial J_j}{\partial g}$ are the modules of the two vectors and, from (A.3), they must be equal:

$$\begin{aligned}
&\frac{1}{1-2\Delta}\left[(1-\Delta) - \frac{2f(f+g)-f^2}{(f+g)^2} + \frac{g^2}{(f+g)^2}\right] = \frac{1}{1-2\Delta}\left[\frac{f^2}{(f+g)^2} + (1-\Delta) - \frac{2g(f+g)-g^2}{(f+g)^2}\right] \\
&(1-\Delta)(f+g)^2 - 2f(f+g) + f^2 + g^2 = f^2 + (1-\Delta)(f+g)^2 - 2g(f+g) + g^2 \\
&-2f(f+g) = -2g(f+g) \\
&f = g \Leftrightarrow \|x_j - y_1\| = \|x_j - y_2\|
\end{aligned} \quad , \tag{A.4}$$

The previous equation imposes distance constraints on the x_j position with respect to targets y_1 and y_2. The direction constraint is given by the equation (A.3), which imposes that the two versors $\frac{\partial f}{\partial x_j} = \frac{x_j - y_1}{\|x_j - y_1\|}$ and $\frac{\partial g}{\partial x_j} = \frac{x_j - y_2}{\|x_j - y_2\|}$ must have the same direction but inverted verses. The only way to satisfy both the constraints is in the midpoint $\bar{x}_j = (y_1 + y_2)/2$ between targets positionsy_1 and y_2. □

(2) To prove that the point $\bar{x}_j = (y_1 + y_2)/2$ is a saddle point, instead of computing the second derivative and evaluate the Hessian matrix, the definition of saddle point is utilized in this paper: a stationary point which admits both ascent and descent directions. Thus, the proof can be translated into finding at least one ascent and one descent direction from the point $\bar{x}_j = (y_1 + y_2)/2$.

Let define $D_1 = \left\|\frac{y_1 + y_2}{2} - y_1\right\|, D_2 = \left\|\frac{y_1 + y_2}{2} - y_2\right\|$ as the two distances between the stationary point and the two targets. It follows that: $D_1 = D_2 = D$. In the stationary point the objective function of equation (A.1) assumes the following value:

$$\begin{aligned}
&J_j = \frac{1}{1-2\Delta}\left[(1-\Delta)D - \frac{D^2}{2D}\right] + \frac{1}{1-2\Delta}\left[(1-\Delta)D - \frac{D^2}{2D}\right] = \\
&\frac{1}{1-2\Delta}\left[D - \Delta D - \frac{D}{2} + D - \Delta D - \frac{D}{2}\right] = \frac{1}{1-2\Delta}\left[(1-2\Delta)D\right] = D
\end{aligned} \quad , \tag{A.5}$$

Regarding the ascent direction, let consider a point $\hat{x}_j$ situated on the equidistant straight line ρ but not situated on the segment connecting y_1 and y_2 (see the following figure).

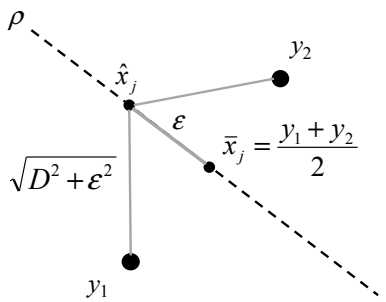


Figure 19: Example of point on the equidistant straight line between target y_1 and y_2.

The objective is to show that moving of a small $\varepsilon > 0$ along the straight line, the objective function J_j increases. It is easily understandable that, if the point moves along such straight line, both distances D_1 and D_2 increase: they become $\hat{D}^{\varepsilon}{}_1 = \hat{D}^{\varepsilon}{}_2 = \sqrt{D^2 + \varepsilon^2} > D$. By substituting $\hat{D}^{\varepsilon}{}_1$ and $\hat{D}^{\varepsilon}{}_2$ into equation (A.1), and noticing that, since the new point belongs to the equidistant straight line, also for $\hat{D}^{\varepsilon}{}_1$ and $\hat{D}^{\varepsilon}{}_2$ the relationship $\hat{D}^{\varepsilon}{}_1 = \hat{D}^{\varepsilon}{}_2 = \hat{D}^{\varepsilon}$ holds:

$$\hat{J}^{\varepsilon}{}_j = \frac{1}{1-2\Delta}\left[(1-\Delta)\hat{D}^{\varepsilon} - \frac{\hat{D}^{\varepsilon 2}}{2\hat{D}^{\varepsilon}}\right] + \frac{1}{1-2\Delta}\left[(1-\Delta)\hat{D}^{\varepsilon} - \frac{\hat{D}^{\varepsilon 2}}{2\hat{D}^{\varepsilon}}\right] = \hat{D}^{\varepsilon} = \sqrt{D^2 + \varepsilon^2} > D = \bar{J}_j$$

Then the direction along the equidistant straight line is an ascent direction. □

Regarding the descent direction, let consider a point situated on the segment connecting the midpoint $\bar{x}_j = (y_1 + y_2)/2$ and the target position y_2, at distance ε from the midpoint. This implies that the distances of the point from the two targets become $\breve{D}^{\varepsilon}{}_1 = D + \varepsilon$ and $\breve{D}^{\varepsilon}{}_2 = D - \varepsilon$, respectively. By substituting the $\breve{D}^{\varepsilon}{}_1$ and $\breve{D}^{\varepsilon}{}_2$ into equation (A.1):

$$\begin{aligned}
\hat{J}^{\varepsilon}{}_j &= \frac{1}{1-2\Delta}\left[(1-\Delta)\breve{D}^{\varepsilon}{}_1 - \frac{\breve{D}^{\varepsilon}{}_1{}^2}{\breve{D}^{\varepsilon}{}_1 + \breve{D}^{\varepsilon}{}_2}\right] + \frac{1}{1-2\Delta}\left[(1-\Delta)\breve{D}^{\varepsilon}{}_2 - \frac{\breve{D}^{\varepsilon}{}_2{}^2}{\breve{D}^{\varepsilon}{}_1 + \breve{D}^{\varepsilon}{}_2}\right] = \\
&= \frac{1}{1-2\Delta}\left[(1-\Delta)(D+\varepsilon) - \frac{(D+\varepsilon)^2}{(D+\varepsilon)+(D-\varepsilon)}\right] + \frac{1}{1-2\Delta}\left[(1-\Delta)(D-\varepsilon) - \frac{(D-\varepsilon)^2}{(D+\varepsilon)+(D-\varepsilon)}\right] = \\
&= \frac{1}{1-2\Delta}\left[2D(1-\Delta) - \frac{D^2+\varepsilon^2+2\varepsilon D}{2D} - \frac{D^2+\varepsilon^2-2\varepsilon D}{2D}\right] = \\
&= \frac{1}{1-2\Delta}\left[2D(1-\Delta) - \frac{D^2}{2D} - \frac{D^2}{2D} - \frac{\varepsilon^2}{2D} - \frac{\varepsilon^2}{2D}\right] = \\
&= \frac{1}{1-2\Delta}\left[2D(1-\Delta) - D - \frac{\varepsilon^2}{D}\right] = D - \frac{1}{1-2\Delta}\frac{\varepsilon^2}{D} < D = \bar{J}_j
\end{aligned} \quad , \qquad \text{(A.6)}$$

Equation (A.6) holds for each $\varepsilon > 0$. Such a direction is a descent direction. ■