A Single Agent Search of a Two Dimensional Space Using Probability Collectives and Convex Optimization

Christopher W. Lum,*
Autonomous Flight Systems Laboratory
University of Washington, Seattle, WA, 98105, USA

March 9, 2006

Abstract

Searching missions are often complicated and difficult operations to undertake due to their dynamic nature. For example, in search and rescue missions, targets are not guaranteed to be stationary and observations become less reliable as time progresses. In addition, searches are often initiated with only a very rough idea of the target location. This work considers the problem of searching a two dimensional space for targets using a single autonomous agent. The system maintains a world model which includes the estimate of possible target states. The issue of compelling the agent to converge on possibly moving targets and continuing to search new regions is formulated as a model predictive control problem. The world model is propagated forward in time and control decisions are made on the predicted future state of the world. Waypoints are assigned by by solving a convex optimization problem using numerical methods.

1 Nomenclature

A_{tgt}, B_{tgt}	Dynamics of estimated target
$\chi(t)$	Particle set at time t
d	Number of time steps to predict ahead
$f_0(x)$	Objective function in $(\wp)$
$f_0(z)$	Objective function in $(\wp_3)$
$f_i(z)$	Inequality constraints in $(\wp_3)$
$h()$	Function which returns $\hat{x}_w(t+p)$ in $(\wp_1)$
I_k	Interval of $[l_k, u_k]$ in $(\wp)$
l_k	Lower limit of interval I_k in $(\wp)$
M	Number of particles in $(\wp)$
$n^{[m]}$	Noise added to particle m in $(\wp)$
$(\wp)$	Problem of minimizing over a box
$(\wp_1)$	Problem of propagating world state
$(\wp_2)$	Problem of finding cell of maximum score
$(\wp_3)$	Problem of finding optimal waypoints
rand(x, y)	Uniformly distributed random number in $[x, y]$
u_k	Upper limit of interval I_k in $(\wp)$
$w^{[m]}(t)$	Weight of particle m at time t
X	A box
$\bar{x}^\star$	Quasi-optimal minimizer in $(\wp)$
$x^{[m]}(t)$	Particle M at time t in $(\wp)$
$\hat{x}_{tgt}(t)$	Estimated state of target at time t
$x_w(t)$	State of world at time t
$\hat{x}_w(t)$	Estimated world state at time t
z	Decision vector
z_0	Agent's state at time t
$\bar{z}^\star$	Quasi-optimal solution in $(\wp_2)$
$\bar{z}$	Optimal solution in $(\wp_3)$
$z_{uav}(t)$	See z_0

2 Introduction

Search and surveillance-type missions typically require heavy human involvement. Tasks of assigning regions to search and coordinating sensor measurements are usually left to human decision making and analysis. In a noisy environment, it becomes difficult for a human operator to classify sensor readings and assign confidence in these readings. Determining regions of high target-location probability and coordinating nearby agents to converge on a particular spot while allowing other vehicles to continue searching is also difficult.

*Research Assistant, Dept. of Aeronautics and Astronautics, lum@u.washington.edu

Therefore, the primary limitation to concurrent operation of multiple vehicles remains lack of autonomy of these vehicles.

Many of these search type missions are initiated with a poor estimate of the target's actual position. To aggravate matters, often the target is moving or evading. Furthermore, visibility and/or lack of computational power eliminates the possibility of identifying targets using a vision based system (either manned or unmanned). A specific example is the detection of a submerged submarine in littoral waters. For a search to be successful, a system must be able to search for and identify a target in an efficient manner. Eventually, these algorithms would be developed to operate with a team of heterogeneous vehicles. This team would be comprised of individual vehicles known as agents. Each agent could have different capabilities and sensors which dictates that algorithms be easily adaptable to accommodate these differences.

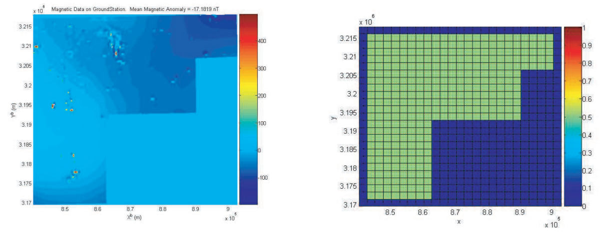
Previous work at the University of Washington has established a framework for the integration of various tasks in an autonomous system. This involves classifying task as either a strategic, tactical, or dynamics and control problem. These correspond to low bandwidth tasks such as path planning[1], medium bandwidth tasks such as target identification, and high bandwidth tasks such as state stabilization, respectively. This works looks at automating the strategic searching task for a single agent in the setting described above.

Other groups such as Durrant-Whyte et al.[2] have studied the problem of searching for a target using a Bayesian probabilistic approach and have investigated some of the communication issues involved in such a search. Polycarpou et al.[3] have applied optimization techniques to generate search patterns over a finite amount of steps. The search strategy presented here follows a similar approach and investigates the effect of incorporating a predictive world estimate to the problem of finding an optimal search pattern.

Section 3 describes the idea of the occupancy based maps and the overall architecture of system. Section 4 describes the predictive world model and how it is used in the overall problem. Finding a quasi-optimal cell to search is described in section 5. Finding a set of waypoints which are feasible in order to place the agent in the quasi-optimal cell is described in section 6. Finally, conclusions and continuing research directions are present in Section 7.

3 Occupancy Based Maps

In order to effectively search a two dimensional domain for a target, the system must keep track of state of the world in terms of possible target locations. To do this, an occupancy based map is employed. In this scheme, the search domain is discretized into rectangular cells. Each cell is assigned a score based on the probability that the target is located in that grid. This is similar to a two dimensional, discretized probability density function [4]. This centralized occupancy based map is shared and updated by all agents involved in the search. At each time step, guidance decisions for each agent are chosen based on this map. An example of this is shown below in Figure 1.



(a) Total magnetic intensity map of region

(b) Initial Occupancy Based Map

Figure 1: Discretization of search region into an occupancy based map.

In Figure 1(b), the blue sections represent scores with zero scores where as the green represents scores of 1/2. This is the initial state of the occupancy based map. It represents having no a priori knowledge of the targets location other than it cannot be in a section where no real data exists (i.e. the sections of uniform blue in Figure 1(a)).

This work does not focus on how the occupancy based map is updated but instead concentrates on how to find a optimal cell to search using this occupancy based map. The overall goal for the agent would be to attempt to converge on regions of high score (a high probability that the target is located there). Of course, the problem of finding a an (x, y) coordinate which maximizes $x_w(t)$ may not be simple. From an optimization standpoint, $x_w(t)$ is in general non-convex, discontinuous, and has gradient equal to either zero or infinity. We now propose a method which yields a quasi-optimal solution which is feasible and is formulated as a convex optimization problem.

The overall flow of the system is shown below in Figure 2.

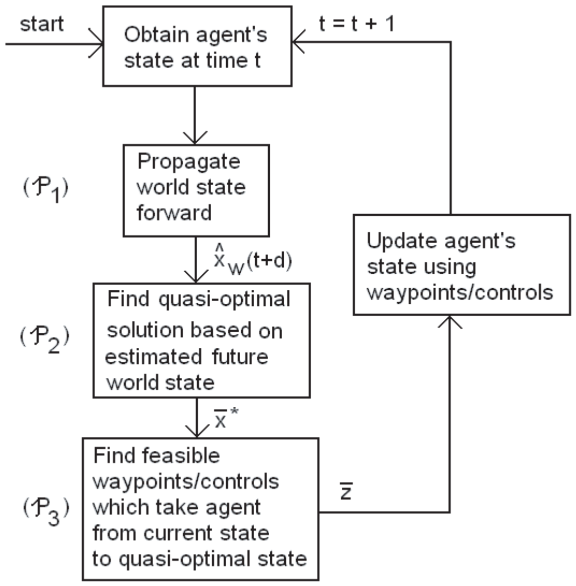


Figure 2: Flow diagram for optimization process

As can be seen, the process is comprised of three main problems which are referred to as $(\wp_1)$, $(\wp_2)$, and $(\wp_3)$.

It starts by finding the agent's state at the current time, $z_{uav}(t)$. Next, $(\wp_1)$ is solved to obtain the estimated world state at time $t + d$. Next, $(\wp_2)$ is solved to find the coordinates of the cell with the maximum score in the reachable cells (these are the cells that the agent can reach in d steps). Once this quasi-optimal solution, $\bar{z}^*$ is found, $(\wp_3)$ consists of finding an optimal set of waypoints/controls, $\bar{z}$ which will take the agent from its current state to the quasi-optimal state. This is formulated as a convex optimization problem.

4 $(\wp_1)$ Predictive World Model

The first problem $(\wp_1)$ involves creating an estimate of the world state at a given time and then projecting this estimate forward in time to obtain the estimated state of the world at time $t + d$. The block diagram for the World Estimator is shown below in Figure 3.

The inputs are the estimated state of the target (position and velocity) of the target at the current time,

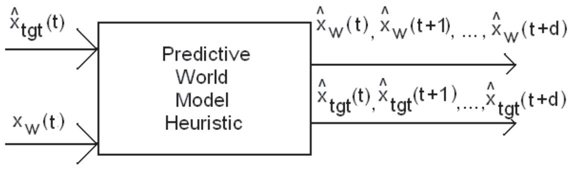


Figure 3: Block diagram for world estimator

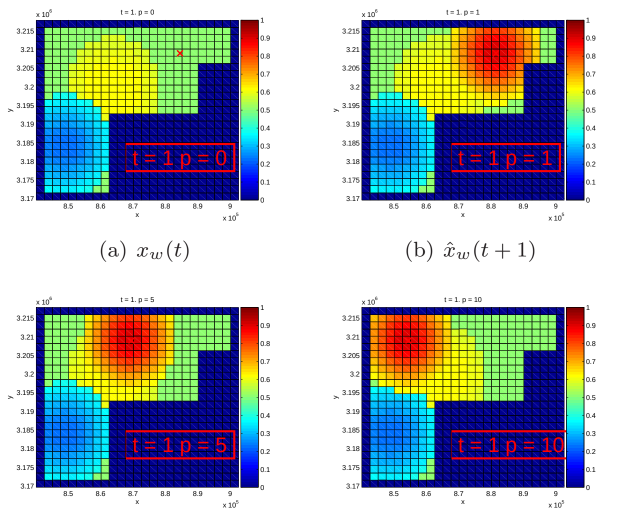
$\hat{x}(t)$ and the current state of the occupancy based map, $x_w(t)$. In order to propagate an estimate of the target state, we assume simple dynamics of the form

$$\hat{x}_{tgt}(t+1) = A_{tgt}\hat{x}_{tgt}(t) + B_{tgt}\hat{u}_{tgt}(t) \qquad (1)$$

The world estimate at time $t + p$ is then a function of the estimated target state at time $t + p$ and the world state at the original time t.

$$\hat{x}_w(t+p) = h(\hat{x}_{tgt}(t+p), x_w(t)) \text{ for } p = 0, ..., d \qquad (2)$$

In our example, the function $h()$ simply adds a two dimensional gaussian centered about $\hat{x}_{tgt}(t + d)$ to $x_w(t)$. An example of this is shown below when the estimated target state is observed to be moving to the left at a constant velocity.



(a) $x_w(t)$ (b) $\hat{x}_w(t+1)$

(c) $\hat{x}_w(t+5)$ (d) $\hat{x}_w(t+10)$

Figure 4: Estimated world states at different times for estimated target moving to the left.

Now that the state of the world can be estimated at time $t + d$, we can attempt to find the coordinates of a cell with a high score in the reachable cells, $\bar{z}^\star$. This is addressed in ($\wp_2$).

5 ($\wp_2$) Probability Collectives

5.1 Theory

In order to find the coordinates of a cell with the maximum score, we use a method which mixes the ideas of probability collectives [5] and particle filters [6]. This provides a method to attempt to find a minimizer to the following problem

$$(\wp) \text{ minimize } f_0(x) \text{ over } x \in X = \text{a box} \tag{3}$$

Recall that a box is defined by each element x_k of $x \in X \subseteq \Re^n$ being in a certain interval $I_k = [l_k, u_k]$.

$$X = \left\{ x \left| \begin{array}{c} x_1 \in I_1 = [l_1, u_1] \\ x_2 \in I_2 = [l_2, u_2] \\ \vdots \\ x_n \in I_n = [l_n, u_n] \end{array} \right. \right\} \tag{4}$$

The difficultly in solving Eq. 3 arises from the fact that the objective function may not be well behaved (i.e. non-convex, non-differentiable, etc.). This is especially true in our case. It may be difficult or impossible to find an optimal solution. An algorithm to find a quasi-optimal, feasible solution is now proposed.

1. Generate M particles (instances of $x \in X$) distributed over X in some fashion.
2. Assign weights to each particle based on its objective function value.
3. Resample the particles proportional to the weights.
4. Repeat step 2 and 3 until some stopping criterion is reached.

Let us examine each step in detail.

5.1.1 Initial Particle Distribution

To find a quasi-optimal minimizer of $f_0(x)$, a finite set of possible minimizers are used. Each individual guess of a minimizer, $x^[m](t)$ is called a particle and together the particles make up the particle set, $\chi(t)$.

$$\chi(t) = \bigcup_M x^{[m]}(t) = \left\{ x^{[1]}(t), x^{[2]}(t), \ldots, x^{[M]}(t) \right\} \tag{5}$$

To initialize the algorithm, we need to assign actual values to the initial particle set. Since there is no a priori knowledge regarding the minimizer of $f_0(x)$, the initial distribution of the particles is chosen as a uniform distribution over the set X

$$x_k^{[m]}(0) = \text{rand}(u_k, l_k) \quad \text{for} \begin{array}{c} m = 1, \ldots, M \\ k = 1, \ldots, n \end{array} \tag{6}$$

5.1.2 Assign Particle Weights

We now assign a weight to each particle.

$$w^{[m]}(t) = -f_0(x^{[m]}(t)) \quad \text{for } m = 1, \ldots, M \tag{7}$$

Note that this assigns a higher weight to particles which yields a smaller objective function value.

5.1.3 Resample Particles

In order to generate the next particle set, we sample from the current particle set proportional to the weights.

$$\tilde{x}^{[m]}(t) = g(\chi(t), w(t)) \quad \text{for } m = 1, \ldots, M \tag{8}$$

Here, $g()$ is a sampling function which samples elements from $\chi(t)$ proportional to the weights $w(t)$. One popular method to do this is to use the roulette wheel method. In this method a roulette wheel with M slots is created. The weights are normalized so that they sum to 1. Each normalized weight then represents the angular percentage that this particle occupies on the roulette wheel. The wheel is spun and depending on where it lands, the corresponding particle $x^{[m]}(t) \in \chi(t)$ is selected as $\tilde{x}^{[m]}(t)$. This process is repeated M times.

As with many genetic algorithms, a mutation process must be included when evolving one population to another. This is true here as well and the mutation operation is represented by simply adding noise to each sample $\tilde{x}^{[m]}$. Recall that we require that each particle $x^{[m]}(t) \in X\ \forall t$. Care must be taken so that the noise added does not "push the particle out of X". The noise must therefore be in the interval

$$n^{[m]}(t) \in [l - \tilde{x}^{[m]}(t), u - \tilde{x}^{[m]}(t)] \quad \text{for } m = 1, \dots, M \tag{9}$$

Finally, the new particle set is determined by

$$x^{[m]}(t+1) = \tilde{x}^{[m]}(t) + n^{[m]}(t) \quad \text{for } m = 1, \dots, M \tag{10}$$

This formulation guarantees that each particle $x^{[m]}(t) \in X\ \forall\, t$ (each particle represents a feasible solution to ($\wp$)). It has the feature that as this evolves from generation to generation, the particles with a higher weight (i.e. lower objective function value) are more likely to continue on to the next population.

5.1.4 Stopping Criterion

Step 2 and 3 are repeated until some stopping criterion is reached. This could be something like the variance of the particles is reduced below some threshold. In this case, we simply repeat it for T steps. The quasi-optimal minimizer is then computed from the average of the final particle set.

$$\bar{x}^\star = \frac{1}{M} \sum_{m=1}^{M} x^{[m]}(T) \tag{11}$$

5.2 Application to Search

We can apply the above method of finding a quasi-optimal solution to ($\wp_2$). We define this problem as

$$(\wp_2) \text{ minimize } f_0(z) \text{ over } z \in Z \tag{12}$$

5.2.1 Parameterizing X

Here, the set Z is all the locations where the agent is able to reach in d steps (reachable states)

$$Z = \left\{ z \,\middle|\, z = z_{uav} + r \begin{pmatrix} cos(\pi/2 - \psi) \\ sin(\pi/2 - \psi) \end{pmatrix}, \begin{matrix} r \in I_r \\ \psi \in I_\psi \end{matrix} \right\} \tag{13}$$

In Eq. 13, the intervals describe the maximum radius and heading angle that the agent can achieve. Since we assume a simple model, we have $I_r = [0, d \cdot \Delta T \cdot V_{max}]$ and $I_\psi = [0, 2\pi]$. An example of this set Z is shown below in Figure 5

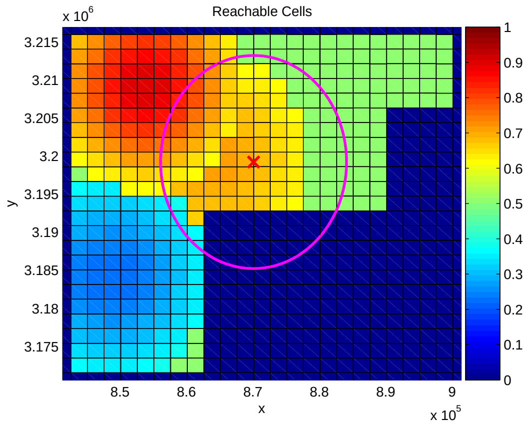


Figure 5: Reachable locations Z shown inside purple circle

Note that the set Z is not a box. So to use the method above, we can parameterize the set using r and ψ, namely

$$x = \begin{pmatrix} r \\ \psi \end{pmatrix} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \tag{14}$$

So the set X is simply

$$X = \left\{ x \,\middle|\, \begin{matrix} x_1 \in I_r = [0, d \cdot \Delta T \cdot V_{max}], \\ x_2 \in I_\psi = [0, 2\pi] \end{matrix} \right\} \tag{15}$$

Using Eq. 13, we see that the conversion between $x \in X$ and $z \in Z$ is simply

$$z = z_{uav} + x_1 \begin{pmatrix} cos(\pi/2 - x_2) \\ sin(\pi/2 - x_2) \end{pmatrix} \tag{16}$$

5.2.2 Define $f_0(x)$

We now need to define the objective function $f_0(x)$. We know that the scores are a function of the position z. We want to find the maximum score, so we actual find the minimizer of the negative scores. Furthermore, in the context of model predictive control, instead of minimizing over the current world state at time t, we actually minimize over the projected world state at time $t+d$. We can use Eq. 16 to convert between z and x, so we can define the objective function as

$$f_0(x) := -\hat{x}_w(t+d) \tag{17}$$

So with Eq. 15 and 17, we see that $(\wp_2)$ is equivalent to $(\wp)$. Therefore the methods described above can be used to find a quasi-optimal minimizer $\bar{x}^\star$ which can be converted into $\bar{z}^\star$. A block diagram showing inputs and outputs for the probability collective minimization routine is shown below in Figure 6.

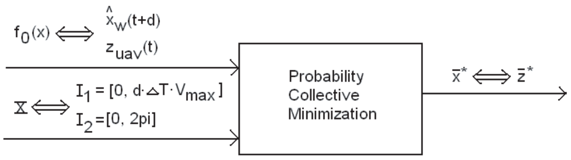
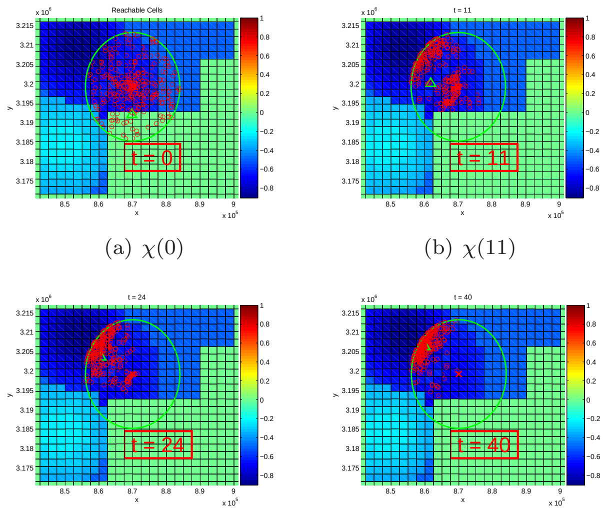


Figure 6: Block diagram for Probability Collective minimization

An example of progression of this process is shown below in Figure 7



(a) $\chi(0)$

(b) $\chi(11)$

(c) $\chi(24)$

(d) $\chi(40)$

Figure 7: Progression of probability collective process. True minimum is located in upper left corner of reachable cells.

The particles are shown at red circles. The centroid of the particles (green triangle) eventually centers near the true optimal solution. Note that it does not achieve the true optimal but it does achieve a feasible solution.

6 $(\wp_3)$ Convex Formulation

We are now consider the final problem $(\wp_3)$ which concerns finding feasible waypoints which take the agent from the current state $z_0 = z_{uav}$ to the quasi-optimal state found in $(\wp_2)$. From an optimization standpoint, the corresponding decision vector $z \in \Re^{2\cdot d}$ is

$$z = \begin{pmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ \vdots \\ z_{2\cdot d-1} \\ z_{2\cdot d} \end{pmatrix} = \begin{pmatrix} z_1(1) \\ z_2(1) \\ z_1(2) \\ z_2(2) \\ \vdots \\ z_1(d) \\ z_2(d) \end{pmatrix} \tag{18}$$

Using Eq. 16, the quasi-optimal solution $\bar{x}^\star$ obtained by the probability collectives method can be converted to an (x, y) coordinate $\bar{z}^\star$. The objective function for $(\wp_3)$ can then be formulated as

$$f_0(z) := \sum_{t=1}^{d} ||z(t) - \bar{z}^\star||_2^2 \tag{19}$$

Here, each coordinate $z(k)$ is a waypoint which dictates where the agent should be located at time k. Obviously, the minimum of this function is zero. This corresponds to all the waypoints being placed at the quasi-optimal solution $\bar{z}^\star$. However this in general would not be feasible. We know that the agent can only travel a distance $r_{max} = \Delta T \cdot V_{max}$ in a single step. So in order for the waypoints to be feasible, we introduce constraints of the form

$$f_1(z) := ||z(1) - z_0||_2^2 - r_{max}^2 \leq 0 \tag{20}$$

$$f_i(z) := ||z(i) - z(i-1)||_2^2 - r_{max}^2 \leq 0 \text{ for } i = 2, \ldots, d \tag{21}$$

Physically, these constraints say that each waypoint must be within a distance r_max of the previous waypoint. This ensures that flight path generated is a feasible one.

The problem can now be formally stated as

$$(\wp_3) \text{ minimize } f_0(z) \text{ over } z \in \Re^{2\cdot d} \qquad (22)$$
$$\text{subject to} f_i(z) \le 0 \text{ for } i = 1, \dots, d$$

In order to analyze what type of optimization problem this is, a closer look at the objective function and constraint functions is required. The objective function can be written as

$$f_0(z) = \frac{1}{2} z^T H z + f^T z + r \qquad (23)$$

$$\begin{array}{llll} \text{where} & H & = & 2I_{d\times d} \\ & f^T & = & 2\begin{pmatrix} \bar{z}_1^\star & \bar{z}_2^\star & \bar{z}_1^\star & \bar{z}_2^\star & \dots \end{pmatrix} \\ & r & = & d\|\bar{z}^\star\|_2^2 \end{array}$$

This is a strictly convex function since it is in a quadratic form and the Hessian is equal to H which is positive definite (all eigenvalues are equal to 2).

The constraint functions can be analyzed in a similar fashion. The first constraint $f_1(z)$ can be written as

$$f_1(z) = \frac{1}{2} z^T H_1 z + f_1^T z + r_1 \qquad (24)$$

$$\begin{array}{llll} \text{where} & H_1 & = & \text{diag}(2I_{2\times 2}, 0_{d-2\times d-2}) \\ & f_1^T & = & \begin{pmatrix} -2z_{1,0} & -2z_{2,0} & 0 & \dots & 0 \end{pmatrix} \\ & r_1 & = & -r_{max}^2 + \|z_0\|_2^2 \end{array}$$

And the constraint functions $f_2(z)$ through $f_d(z)$ can be written as

$$f_i(z) = \frac{1}{2} z^T H_i z + f_i^T z + r_i \text{ for } i = 2, \dots, d \qquad (25)$$

$$\begin{array}{llll} \text{where} & H_i & = & \text{diag}(N_i, A, M_i) \\ & f_i^T & = & \begin{pmatrix} 0 & \dots & 0 \end{pmatrix} \\ & r_i & = & -r_{max}^2 \\ & N_i & = & \text{zeros}(2(i-2)) \\ & M_i & = & \text{zeros}(2d-4-2(i-2)) \\ & A & = & \begin{pmatrix} 2 & 0 & -2 & 0 \\ 0 & 2 & 0 & -2 \\ -2 & 0 & 2 & 0 \\ 0 & -2 & 0 & 2 \end{pmatrix} \end{array}$$

In Eq. 24 and 25, $\text{diag}(x, y)$ represents a block diagonal matrix with submatrix x in the upper left block and submatrix y in the lower right corner. Similarly, $\text{zeros}(p)$ represents a square zero matrix of size $p \times p$. Although this looks like a complicated formulation, note that only H_i is changes with each i. The formulation just describes that H_2 is a block diagonal matrix with A in the upper left corner and zeros elsewhere. H_3 is a block diagonal matrix where the submatrix A moves two columns to the right and two rows down. This process of moving the A matrix by 2 rows and columns with each i is described by the N_i and M_i submatrices.

One can now see that the constraint function $f_i(z)$ for $i = 1, \dots, d$ are convex functions because they are in quadratic forms and their respective Hessians are all positive semi-definite.

Therefore, we see that $(\wp_3)$ consists of a convex objective function over a convex set, so this is a convex programming problem. We can show that it is well posed and the feasible set is non-empty, so a unique optimal solution exists. In a similar fashion to the previous two problems, $(\wp_3)$ can be packaged nicely into a system shown below

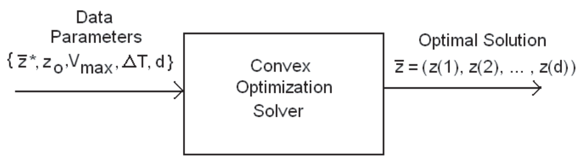


Figure 8: Block diagram for convex optimization solver

An example of the solution for the example situation (with $d = 10$) is shown below in Figure 9.

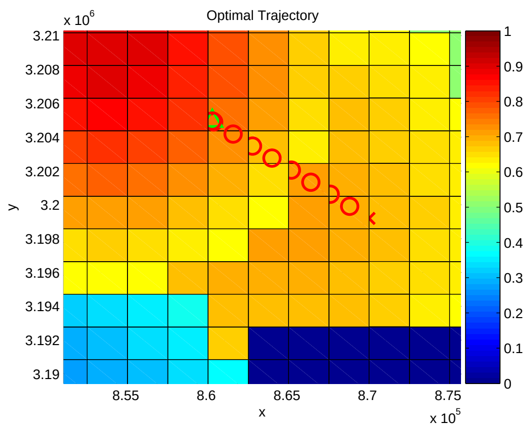


Figure 9: Optimal solution $\bar{z}$ to $(\wp_3)$ zoomed into area of interest with $d = 10$

Figure 9, the red x represents the location of the agent, the green triangle is the desired destination $\bar{z}^{\star}$, and the red circles represent waypoints $\bar{z}$. The thing to notice is although $d = 10$, there are only 8 visible waypoints. This is because waypoints 8, 9, and 10 are overlapping and all are equal to $\bar{z}^{\star}$. Furthermore, constraints $f_1(z)$ through $f_7(z)$ are active. This shows that the formulation of the objective function yields waypoints which place the agent at the optimal solution in the shortest possible time. This is the desired behavior during a searching type application where it is desired that the agent location and verify the target as soon as possible.

7 Conclusions

In conclusion, the method proposed gives a way to formulate the difficult problem of finding and optimal coordinate to search as a convex optimization problem with a guaranteed unique optimal solution. Furthermore, the formulation yields a solution which places the agent at $\bar{z}^{\star}$ in minimum time.

The main approximation that was made is that $\bar{z}^{\star}$ is only a quasi-optimal coordinate. However, if the procedure is executed for each time step as shown in Figure 2, then $\bar{z}^{\star}$ is only needed to obtain the desired heading of the agent. Only the first two elements of $\bar{z}$ are actually used as they are the first waypoint.

Future research will be directed towards incorporating this process for multiple agents to work in a coordinated fashion. Also method to update the occupancy map will be investigated.

References

[1] Pongpunwattana, A. and Rysdyk, R., "Real-Time Planning for Multiple Autonomous Vehicles in Dynamic Uncertain Environments," *AIAA JACIC*, December 2004, pp. 580–604.

[2] Bourgault, F. and Durrant-Whyte, H. F., "Communication in General Decentralized Filters and the Coordinated Search Strategy," *Proceedings of the 7th International Conference on Information Fusion*, Australian Centre for Field Robotics, Stockholm, Sweden, 2004.

[3] Flint, M., Polycarpou, M., and Fernandez-Gaucherand, E., "Cooperative Control for Multiple Autonomous UAV's Searching for Targets," *Proceedings of the 41st IEEE Conference on Decision and Control*, University of Cincinnati, Las Vegas, NV, 2004.

[4] Bourgault, F., Furukawa, T., and Durrant-Whyte, H., "Coordinated Decentralized Search for a Lost Target in a Bayesian World," *Proceedings of the 2003 IEEE/RSJ Intl. Conference on Intelligent Robots and Systems*, Australian Centre for Field Robotics, Las Vegas, NV, October 2003.

[5] Huang, C.-F., Wolpert, D. H., Bieniawski, S., and Strauss, C. E., "A Comparative Study of Probability Collectives Based Multi-agent Systems and Genetic Algorithms," *Proceedings of the GECCO 2005 Conference*, 2005.

[6] Fox, D., Hightower, J., Liao, L., and Schulz, D., "Bayesian Filtering for Location Estimation," *IEEE Pervasive Computing*, July 2003, pp. 23–33.