Role Assignment in Multi-Agent Coordination *

Meng Ji† Shun-ichi Azuma‡ and Magnus Egerstedt†

***Abstract*— In this paper we study the problem of parameterized assignment. This problem arises when a team of mobile robots must decide what role to take on in a given planar formation, where the parameters are the rotation and translation of the formation. A suite of suboptimal, but computationally tractable (polynomial time) algorithms are given, based on a solution to the problem of finding the optimal translation and rotation given a fixed assignment. Numerical examples show the viability of the proposed, suboptimal solutions.**

***Index Terms*— Coordination; Multi-agent control; Assignment problem**

I. Introduction

A standard issue in multi-agent coordination is the problem of formation control, where a "formation" is understood to be a rotationally and translationally invariant configuration. In particular, once a desired translation and rotation has been established, the *dispatch* problem concerns the issue of sending a collection of agents from their initial positions to the target positions, e.g. while avoiding collisions or preserving energy. In this paper, we consider the *pre-dispatch* problem, namely the problem of determining the following three arguments:

1) *Rotation*: How should the target formation be rotated?
2) *Translation*: How should the target formation be translated?
3) *Assignment*: What roles in the target formation should the individual agents be assigned to?

In other words, given a collection of N planar agents, whose positions are $x_i \in \mathbb{R}^2,\ i = 1, \ldots, N$, and a representation of the target formation through $y_i \in \mathbb{R}^2,\ i = 1, \ldots, N$, what we are interested in is determining the rotation angle θ and translation $v \in \mathbb{R}^2$ of the target formation. Moreover, we also need to answer the question of "Who goes where?", i.e. find an appropriate permutation $p : \{1, \ldots, N\} \to \{1, \ldots, N\}$ that assigns target position $y_{p(i)}$ to agent i, located at x_i.

In this paper, we will show how the simultaneous rotation, translation, and assignment optimization problem can be cast as a parameterized assignment problem. And, it should be noted already at this point that the solution to this problem will result in a centralized algorithm in the sense that the computation of our solutions will require complete information about all agents in the team. As such, the algorithms can not be used in decentralized decision making scenarios. Instead, they should be thought of as "locker-room agreements" in the sense that before the team is actually deployed, the solution will have to have been obtained in a centralized manner.

One would like to be able to implement the proposed algorithms for teams of arbitrary size (possibly very large). As such, we need to know if the solutions scale polynomially in the problem size. In this paper, only heuristic but computational tractable, suboptimal approaches will be proposed and the computational complexity issue will be left to the future.

The outline of this paper is as follows: Section II presents the background to the problem while, in Section III, we define the problem and present a theorem which is essential to the numerical solutions proposed in Section IV. In the next section numerical examples are shown and the results of the different methods are compared. The last section contains a discussion of further issues, such as computational complexity and decentralization issues.

II. Background

During the last five years, several problems have attracted a lot of attention in the field of multi-agent, distributed control, such as rendezvous problems, agreement problems, formation control problems and so forth.

The rendezvous problem i.e. the problem of driving all agents to a single point, has for instance been addressed in [12], [17], [18]. A variety of algorithms have been proposed such that a group of agents can reach some agreement, such as direction, position, etc., with only local information. The local information limitation is an embodiment of the limited range of sensors or communication channels available to the individual agents.

Another problem in multi-agent robot control that has received considerable attention is that of formation control [1], [2], [3], [4], [5], [7], [8], [9], [11], [13]. Here the task is to drive a given formation error to zero. This formation error has to be defined in such a way that it reflects how well the formation is being achieved. Examples include deviations from desired positions [13], deviations from desired inter-robot distances [11], or as dissimilarities between graphs encoding the desired and actual formations [12].

As mentioned before, formation control is one of the central themes in the multi-agent literature. The com-

* This work was supported by the U.S. Army Research Office through the grant # 99838.

† School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30032, USA. mengji,magnus@ece.gatech.edu.

‡ Graduate School of Informatics, Kyoto University , Uji, Kyoto 611-0011, Japan sazuma@i.kyoto-u.ac.jp.

plete formation procedure can be roughly divided into three phases as will be the case in the paper:

1) *Gathering (Rendezvous)*
The goal of the first phase is to build a complete graph. Given an arbitrary initial configuration of the multi-agent group, apply a rendezvous algorithm to drive the agents sufficiently close to one another in order to facilitate locker-room agreement.
2) *Role Assignment*
In the second phase, a position must be selected in the for each agent in such a way that the displacement of the whole group is minimized.
3) *Dispatching*
After an assignment is chosen, the group adopts a formation control scheme to reach the target formation. (E.g. [16], [6].)

The main focus in this paper is on the second phase, i.e. the agent-target matching problem, or reconfiguration problem. The problem has in fact been studied in both 2D [2] and 3D [3] cases. However the novelty with this paper lies in a formulation and analysis of the combination of rotation and translation. In the following section, we lay out the mathematical foundation of the problem and present a theorem that will serve as a basis for the subsequent suboptimal solution.

III. Problem Formulation

The problem considered in this paper is to simultaneously optimize (i) the translation and rotation of the target formation, and (ii) the assignment (or matching) of agents to targets. This is mathematically formulated as follows:

Let $N \in \mathbb{N}$ denote the number of the agents, and let $x_1, x_2, \ldots, x_N \in \mathbb{R}^2$ be the planar positions of the agents and $y_1, y_2, \ldots, y_N \in \mathbb{R}^2$ be that of the targets, which will be expressed as $x \triangleq (x_1, x_2, \ldots, x_N)$ and $y \triangleq (y_1, y_2, \ldots, y_N)$. We denote by $\theta \in [0, 2\pi)$ and $v \in \mathbb{R}^2$ the angular rotation and the translation of the target formation. The assignment of the agents to the targets is described by p, which is an element of the set P_N, i.e., the set of all possible permutations over N elements. For example, $p = \{2, 3, 1\}$, with $N = 3$, means that the agents at x_1, x_2, x_3 are assigned to the targets at y_2, y_3, y_1, respectively. Furthermore, the i-th element of p is represented by $p(i)$.

Now, consider the following problem:

$$\Sigma_c(x, y) : \min_{(v,\theta,p)\in\mathbb{R}^2\times[0,2\pi)\times P_N} J_c(x, y, v, \theta, p), \tag{1}$$

where $J_c(x, y, v, \theta, p)$ is the cost

$$J_c(x, y, v, \theta, p) \triangleq \sum_{i=1}^{N} c(x_i, R(\theta)(y_{p(i)} + v)). \tag{2}$$

Here, $R(\theta)$ is the rotation matrix, i.e.,

$$R(\theta) \triangleq \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}, \tag{3}$$

and c is a performance measure. The interpretation is that $c : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}$ gives the cost of assigning the agent i at x_i to the target located at $R(\theta)(y_{p(i)} + v)$.

For example, if c is the square of the l_2-norm (Euclidean norm) of the difference between x_i and $R(\theta)(y_{p(i)} + v)$, we get

$$\Sigma_{l_2^2}(x, y) : \min_{(v,\theta,p)\in\mathbb{R}^2\times[0,2\pi)\times P_N} J_{l_2^2}(x, y, v, \theta, p), \tag{4}$$

$$\text{s.t. } J_{l_2^2}(x, y, v, \theta, p) = \sum_{i=1}^{N} \|x_i - R(\theta)(y_{p(i)} + v)\|_2^2. \tag{5}$$

The corresponding optimization problem denoted as $\Sigma_{l_2^2}(x, y)$ is the problem under consideration in this paper.

Theorem 3.1: Suppose that $N \in \mathbb{N}$, $x_1, x_2, \ldots, x_N \in \mathbb{R}^2$, and $y_1, y_2, \ldots, y_N \in \mathbb{R}^2$ are given. Let (v^*, θ^*, p^*) denote a globally optimal solution to $\Sigma_{l_2^2}(x, y)$. Then the following holds.
(i) The optimal translation is

$$v^* = R(\theta^*)^T x_c - y_c, \tag{6}$$

where $x_c \triangleq \frac{1}{N}\sum_{i=1}^{N} x_i$ and $y_c \triangleq \frac{1}{N}\sum_{i=1}^{N} y_i$ are the centers of mass of the agent and target positions, respectively.
(ii) The optimal solution to the problem

$$\min_{\theta\in[0,2\pi)} J_{l_2^2}(x, y, v^*, \theta, p^*) \tag{7}$$

is

$$\theta^* = \tan^{-1}\left(\frac{W_2(v^*, p^*)}{W_1(v^*, p^*)}\right), \tag{8}$$

where

$$\begin{aligned} W_1(v, p) &\triangleq \sum_{i=1}^{N} x_i^T (y_{p(i)} + v), \\ W_2(v, p) &\triangleq \sum_{i=1}^{N} x_i^T \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} (y_{p(i)} + v). \end{aligned}$$

(iii) The optimal (possibly not unique) assignment satisfies

$$p^* = \operatorname*{argmin}_{p\in P_N} J_{l_2^2}(x, y, v^*, \theta^*, p). \tag{9}$$

Moreover, the problem

$$\min_{p\in P_N} J_{l_2^2}(x, y, v^*, \theta^*, p) \tag{10}$$

corresponds to the well-known linear assignment problem.

Proof: (i) Since

$$\begin{aligned}
J_{l_2^2}(x,y,v,\theta,p) &= \sum_{i=1}^{N} \|x_i - R(\theta)(y_{p(i)}+v)\|_2^2 \\
&= \sum_{i=1}^{N} \big[(R(\theta)^T x_i - y_{p(i)} - v)^T R(\theta)^T \\
&\quad R(\theta)(R(\theta)^T x_i - y_{p(i)} - v)\big] \\
&= \sum_{i=1}^{N} \big[\big(v - (R(\theta)^T x_i - y_{p(i)})\big)^T \\
&\quad \big(v - (R(\theta)^T x_i - y_{p(i)})\big)\big],
\end{aligned}$$

the derivative of $J_{l_2^2}(x,y,v,\theta,p)$ with respect to v is given by

$$\begin{aligned}
\frac{\partial J_{l_2^2}}{\partial v} &= \sum_{i=1}^{N} 2\big(v - (R(\theta)^T x_i - y_{p(i)})\big)^T \\
&= 2N\big(v - (R(\theta)^T x_c - y_c)\big)^T.
\end{aligned}$$

It should be noted that this derivative does not depend on the assignment p. Hence, by noting that $v^* = \operatorname{argmin}_{v\in\mathbb{R}^2} J_{l_2^2}(x,y,v,\theta^*,p)$ for any p, and that $J_{l_2^2}$ is convex in v, we obtain

$$v^* - (R(\theta^*)^T x_c - y_c) = 0 \tag{11}$$

as the first order necessary condition for the problem $\min_{v\in\mathbb{R}^2} J_{l_2^2}(x,y,v,\theta^*,p^*)$, which proves (6).
(ii) From

$$\begin{aligned}
\|x_i - R(\theta)(y_{p(i)}+v)\|_2^2 = x_i^T x_i - 2x_i^T R(\theta)(y_{p(i)}+v) \\
+(y_{p(i)}+v)^T(y_{p(i)}+v),
\end{aligned}$$

it follows that

$$\theta^* = \operatorname*{argmin}_{\theta\in[0,2\pi)} \sum_{i=1}^{N} -x_i^T R(\theta)(y_{p^*(i)}+v^*). \tag{12}$$

In addition, we have

$$\begin{aligned}
&\sum_{i=1}^{N} -x_i^T R(\theta)(y_{p(i)}+v) \\
&= -\sum_{i=1}^{N} x_i^T \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} (y_{p(i)}+v)\cos\theta \\
&\quad + x_i^T \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} (y_{p(i)}+v)\sin\theta \\
&= -\left(W_1(v,p)\cos\theta + W_2(v,p)\sin\theta\right),
\end{aligned}$$

and hence (8) follows.
(iii) Per definition, (9) holds. In addition, by defining $\tilde{y}_{p(i)} \triangleq R(\theta^*)(y_{p(i)}+v^*)$ we obtain

$$\min_{p\in P_N} J_{l_2^2}(x,y,v^*,\theta^*,p) = \min_{p\in P_N} \sum_{i=1}^{N} \|x_i - \tilde{y}_{p(i)}\|^2,$$

which in turn implies that (10) corresponds to the linear assignment problem. (See [14] for further details of the linear assignment problem.) In this way, This completes the proof. ∎

Note that $J_{l_2^2}$ is convex in θ on $[\frac{-\pi}{2}, \frac{\pi}{2}]$, but not in p, since it is a discrete decision variable over a finite set and convexity is only well-defined over topological spaces.

What Theorem 3.1 means is that we can solve the problem $\Sigma_{l_2^2}$ if two of the three optimal parameters v^*, θ^*, or p^* are provided. In fact, (i) implies that v^* does not depend upon p^* and can be obtained by (6) if θ^* is provided. What (ii) shows is that θ^* is given by (8) if v^* and p^* are given. Moreover (iii) implies that since (10) is a linear assignment problem, p^* is easily computed e.g., using the Hungarian method, which is a polynomial time algorithm whose computational complexity is $O(N^3)$ [14]. However, the problem of solving for these three parameters simultaneously is not trivial and initial work by the authors and coworkers suggest that this problem is in fact $\mathcal{NP}$-hard. Instead, three heuristic suboptimal solutions are given in next section.

IV. Suboptimal Solutions

As the problem becomes more complex when we want to optimize over the three parameters simultaneously, we seek some feasible way which can lead us to suboptimal, yet reasonably good solutions to the problem (4).

Note that in Theorem 3.1, (10) and (7) correspond to (4) with fixed θ and with fixed p, respectively, which implies that if either θ or p is fixed, (4) can be solved. More precisely, (7) is explicitly solved and (10) can be efficiently solved using well-known methods for the linear assignment problem, e.g., using the Hungarian method.

First, observe that since v^* is independent of p, and in fact given by

$$v^*(\theta) = R^T(\theta)x_c - y_c,$$

we can express the cost $J_{l_2^2}(x,y,v,\theta,p)$ without reference to v, through

$$J_{l_2^2}(x,y,\theta,p) = \sum_{i=1}^{N} \|x_i - R(\theta)(y_{p(i)} + R^T(\theta)x_c - y_c)\|_2^2.$$

Now, note that

$$\begin{aligned}
&x_i - R(\theta)(y_{p(i)} + R^T(\theta)x_c - y_c) \\
&= x_i - R(\theta)y_{p(i)} - x_c + R(\theta)y_c) \\
&= x_i - x_c - R(\theta)(y_{p(i)} - y_c).
\end{aligned}$$

Hence, with a slight abuse of notation, we can assume that x_c and y_c have already been absorbed by the state variables. In other words, we let $x_i \triangleq x_i - x_c$ and $y_i \triangleq y_{p(i)} - y_c$, which corresponds to original and target formations whose center of mass is equal to the origin. Since the decision variables here are p and θ, for simplicity reason, we will denote the cost function as $J(p,\theta)$ for given x and y, if it is clear from context.

Based on Theorem 3.1, together with the above observation we propose four methods. In what follows, $(\theta_i^{\#}, p_i^{\#})$ $(i = 1,2,3,4)$ represent the corresponding suboptimal solutions.

Method A: Arbitrary Initial Rotation

In this method, we start from a target formation with 0 initial rotation, i.e. $\theta_{initial} = 0$, and find the resulting optimal assignment of $p_1^{\#}$. Then, based on that assignment, we find the optimal rotation angle $\theta_1^{\#}$, i.e.

$$\begin{cases} p_1^{\#} & \triangleq & p^*(0), \\ \theta_1^{\#} & \triangleq & \theta^*(p_1^{\#}). \end{cases} \quad (13)$$

Rather than producing a particularly good solution, this simple method gives us a basic building block from which we can construct more sophisticated methods, leading to better results. One way to compose a better method is to repeat *Method A*, which leads to the next method.

Method B: Iterative Method

Another possible approach for obtaining a practical solution is to mutually and iteratively apply Theorem 3.1 (i) and (ii). In other words, repeat *Method A* until its solution converges. The solution is thus given by

$$\begin{cases} p_2^{\#} & \triangleq & p_2^{\#}(N_{iter}) \\ \theta_2^{\#} & \triangleq & \theta_2^{\#}(N_{iter}), \end{cases} \quad (14)$$

where N_{iter} is the total number of the iterations, $\theta_2^{\#}(i)$ and $p_2^{\#}(i)$, for $i = 1, 2, \ldots, N_{iter}$, are defined as $p_2^{\#}(0) \triangleq p^*(0)$, $\theta_2^{\#}(0) \triangleq \theta^*(p_2^{\#}(0))$, and

$$\begin{cases} p_2^{\#}(i) & \triangleq & p^*(\theta_2^{\#}(i-1)) \\ \theta_2^{\#}(i) & \triangleq & \theta^*(p_2^{\#}(i)). \end{cases} \quad (15)$$

Using this method, we expect substantial improvements in the solution. At the same time, the computational cost increases linearly in N_{iter}. However, because of the discrete and finite set P over which p takes values, it is unclear if this method converges, and if so, to what accumulation point. Note that the notion of a local minimum is ill-defined since P is not a topological space.

Proposition 4.1: Let $J_{i,j} \triangleq J(p_2^{\#}(i), \theta_2^{\#}(j))$. For any arbitrary robot and target positions x and y, there exists a finite positive integer $k \leq \mathrm{card}(P) = N!$, where $\mathrm{card}(\cdot)$ denotes cardinality, such that $\forall\ 1 \leq i \leq k$

$$J_{i-1,i-1} \geq J_{i,i-1} \geq J_{i,i}. \quad (16)$$

Furthermore, $\forall\ i \geq k+1$

$$\begin{cases} p_2^{\#}(i) = p_2^{\#}(k) \\ \theta_2^{\#}(i) = \theta_2^{\#}(k-1). \end{cases} \quad (17)$$

Proof: From the definition (15), we have

$$p_2^{\#}(i) = \operatorname*{argmin}_{p \in P} J(p, \theta^{\#}(i-1))$$

and hence $J_{i-1,i-1} \geq J_{i,i-1}$. Moreover

$$\theta_2^{\#}(i) = \operatorname*{argmin}_{\forall \theta \in (0, 2\pi]} J(p_2^{\#}(i), \theta),$$

which implies that $J_{i,i-1} \geq J_{i,i}$, and hence (16) follows. Furthermore, since p takes value in a finite set and we know that there exists a global minimum, the iterative sequence in (16) has to terminate in a finite number of steps. ■

Remark 4.2: The solution does not necessarily converge to the global minimum.

Method C: Angular Discretization

The solution $(\theta_3^{\#}, p_3^{\#})$ is given based on the discretization of the rotation angles,

$$\begin{cases} p_3^{\#} & \triangleq & p^*(\theta_3^{\#}) \\ \theta_3^{\#} & \triangleq & \operatorname{argmin}_{\theta \in \{\hat{\theta}_0, \hat{\theta}_1, \ldots, \hat{\theta}_{d-1}\}} J(x, y, \theta, p^*(\theta)), \end{cases} \quad (18)$$

i.e. $(\theta_3^{\#}, p_3^{\#})$ is an optimal solution to the problem

$$\min_{\theta \in \{\hat{\theta}_0, \hat{\theta}_1, \ldots, \hat{\theta}_{d-1}\}} \min_{p \in P_N} \sum_{i=1}^{N} \|x_i - R(\theta) y_{p(i)}\|^2. \quad (19)$$

Note that we calculate an optimal solution to (19) by solving (10) for every $\theta \in \{\hat{\theta}_0, \hat{\theta}_1, \ldots, \hat{\theta}_{d-1}\}$, where $\theta^*(p)$ and $p^*(\theta)$ are defined in Theorem 3.1, and $\hat{\theta}_i \triangleq 2\pi k/d$ $(k = 0, 1, \ldots, d-1)$ for a given positive integer d.

Method D: Improved Angular Discretization

This is an extension of Method C in the sense that after finding the optimal assignment $p^*(\theta_3^{\#})$, we proceed with one more step of optimization in which we find the optimal angle with respect to $p^*(\theta_3^{\#})$. Hence

$$\begin{cases} p_4^{\#} & \triangleq & p^*(\theta_3^{\#}) \\ \theta_4^{\#} & \triangleq & \theta^*(p_4^{\#}). \end{cases} \quad (20)$$

V. Numerical Examples

In the previous section, we proposed several suboptimal solutions that are feasible to implement. Now, we need to examine:

- Whether the results are close enough to the optimal solution?
- How much time dose each method take?

Table I and Fig. 1 show the simulation results for $N = 8$, where we take $d \triangleq 100$ for obtaining $(\theta_3^{\#}, p_3^{\#})$ and $(\theta_4^{\#}, p_4^{\#})$. Moreover the problem (10) is solved by the Hungarian method, where the optimal solution obtained by enumerating all elements of P_N (note $\mathrm{card}(P_N) = 40320$) is also shown.

We can see that $(\theta_3^{\#}, p_3^{\#})$ and $(\theta_4^{\#}, p_4^{\#})$ are better than the other solutions in terms of their costs. On the other hand, $(\theta_1^{\#}, p_1^{\#})$ is better from the viewpoint of the computation time, while $(\theta_2^{\#}, p_2^{\#})$ might be the best compromise between accuracy and computation time. As the number of iterations is set to 30, the evolution of the solution is shown in Fig. 2. It is worth noticing that the solution is not improving after some iterations, for *Method B* does not guarantee a global optimal solution.

Finally, we apply *Method D* to the originally posed three-phase problem of sequential *rendezvous*, *assignment*, and *dispatch*. An example of this is shown in Figures 3, 4 and 5

TABLE I

Result of numerical simulations.

i	1	$2(N_{iter} \triangleq 30)$	3	4	optimal
$\theta_i^{\#}$ [rad]	-0.216	-1.3339	-1.916	-1.931	-1.931
$\{\text{robots} \to p_i^{\#}\}$	$\left\{\begin{array}{c} 1\to 8 \\ 2\to 3 \\ 3\to 7 \\ 4\to 1 \\ 5\to 4 \\ 6\to 2 \\ 7\to 5 \\ 8\to 6 \end{array}\right\}$	$\left\{\begin{array}{c} 1\to 8 \\ 2\to 3 \\ 3\to 6 \\ 4\to 4 \\ 5\to 7 \\ 6\to 5 \\ 7\to 2 \\ 8\to 1 \end{array}\right\}$	$\left\{\begin{array}{c} 1\to 6 \\ 2\to 8 \\ 3\to 5 \\ 4\to 4 \\ 5\to 7 \\ 6\to 1 \\ 7\to 3 \\ 8\to 2 \end{array}\right\}$	$\left\{\begin{array}{c} 1\to 6 \\ 2\to 8 \\ 3\to 5 \\ 4\to 4 \\ 5\to 7 \\ 6\to 1 \\ 7\to 3 \\ 8\to 2 \end{array}\right\}$	$\left\{\begin{array}{c} 1\to 6 \\ 2\to 8 \\ 3\to 5 \\ 4\to 4 \\ 5\to 7 \\ 6\to 1 \\ 7\to 3 \\ 8\to 2 \end{array}\right\}$
$J(\theta_i^{\#}, p_i^{\#})$	63.6973	33.9484	32.3675	32.3529	32.3529
Computation time [sec]	0.016	0.3900	0.922	0.938	260.203

Remark 5.1: *Method D* will lead to the global solution if the discretization of θ is fine enough. In other words, as d becomes large enough the global minimum will be obtained. This observation could be turned into an algorithm (*Method D'*), where d is iteratively increased. However, it is premature to declare that *Method D'* is a numerically tractable algorithm, because the problem of choosing a lower bound on d is not known.

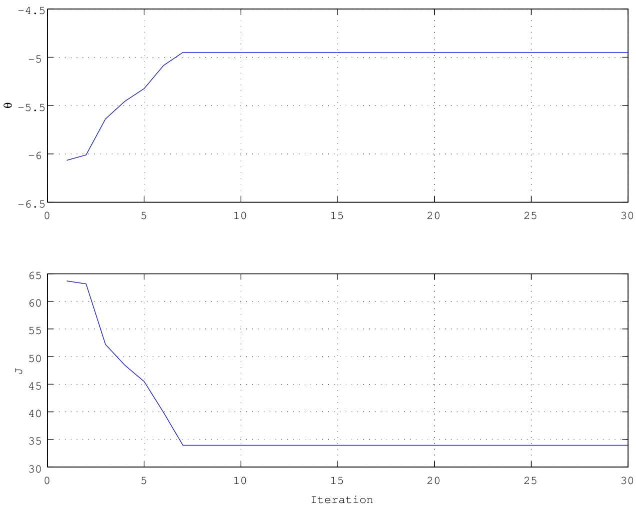

Fig. 2. Changes of $\theta_2^{\#}(i)$ and $J(\theta_2^{\#}(i), p_2^{\#})$ in Method B. Depicted is the rotation angle as a function of the iteration number (upper figure) together with the corresponding cost (lower figure).

VI. Discussions and Conclusion

To conclude, let us discuss some issues pertaining to the computational complexity associated with the different assignment algorithms. As shown in [14], the problem (10) is solved by the Hungarian method using $O(N^3)$ operations, which is higher than the complexity associated with (7). This is the case since in order to solve (7) the number of necessary function evaluations is linear in N.

We now let N_a^i $(i = 1, 2, 3, 4)$ denote the number of assignments that must be solved in order to compute the four suboptimal solutions in equations (13)-(20). It

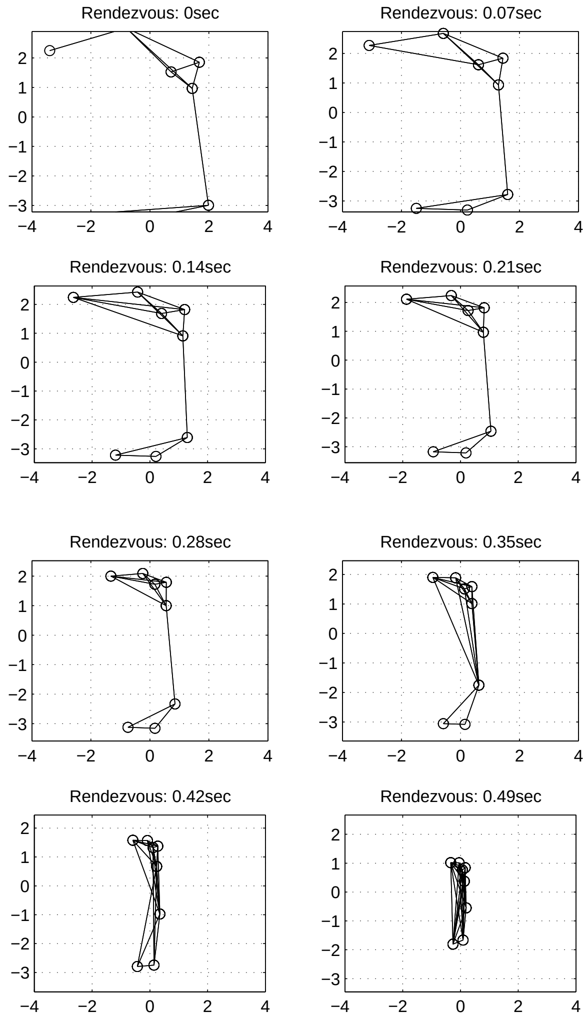

Fig. 3. Phase I: The rendezvous procedure, starting from an arbitrary connected graph, generate a complete graph after 0.45 second.

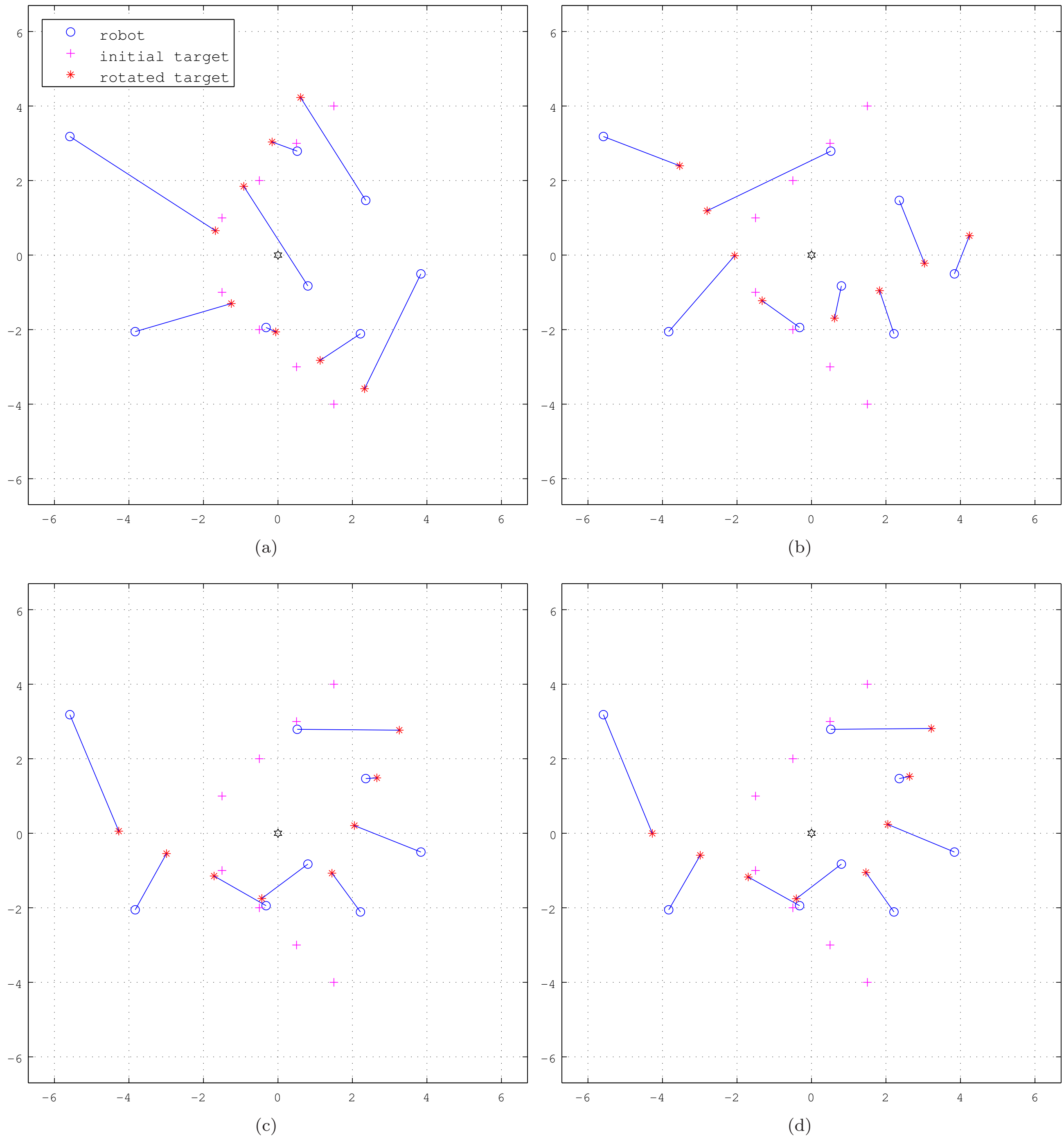

Fig. 1. Result of numerical simulations: (a) Method A: $(\theta_1^\#, p_1^\#)$; (b) Method B: $(\theta_2^\#, p_2^\#)$; (c) Method C: $(\theta_3^\#, p_3^\#)$; (d) Method D: $(\theta_4^\#, p_4^\#)$.

is straightforward to show that

$$N_a^1 = 1, \tag{21}$$
$$N_a^2 = N_{iter}, \tag{22}$$
$$N_a^3 = d, \tag{23}$$
$$N_a^4 = d + 1. \tag{24}$$

Hence, the computational complexity associated with the best of the four methods is $O(N^3 d)$, which is certainly less than the $O(N!)$ obtained through permutation enumeration.

The question now is how to choose d in such a way that the solution to the problem in (20) approaches the solution to the original problem (4) as well as possible. To this end we let n_N be the average number of distinctly different assignments encountered as θ sweeps through d values, as $d \gg 1$. The average is obtained by generating a large number of random formations of N agents. In Figure 6 we have plotted n_N as a function of N and it appears that n_N is linear in N. What this means is that d should be linear in N in order to obtain an adequate

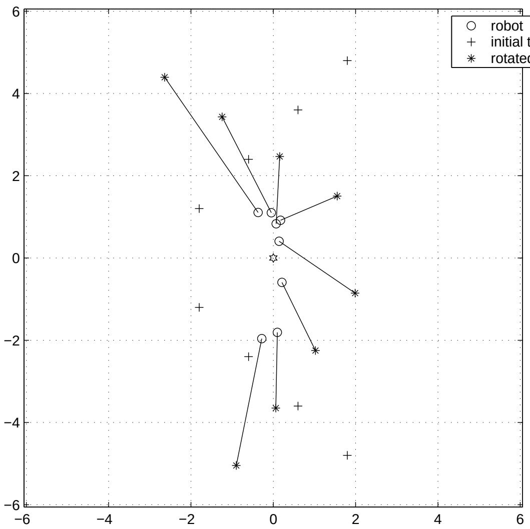

Fig. 4. Phase II: A suboptimal assignment is obtained using Method D.

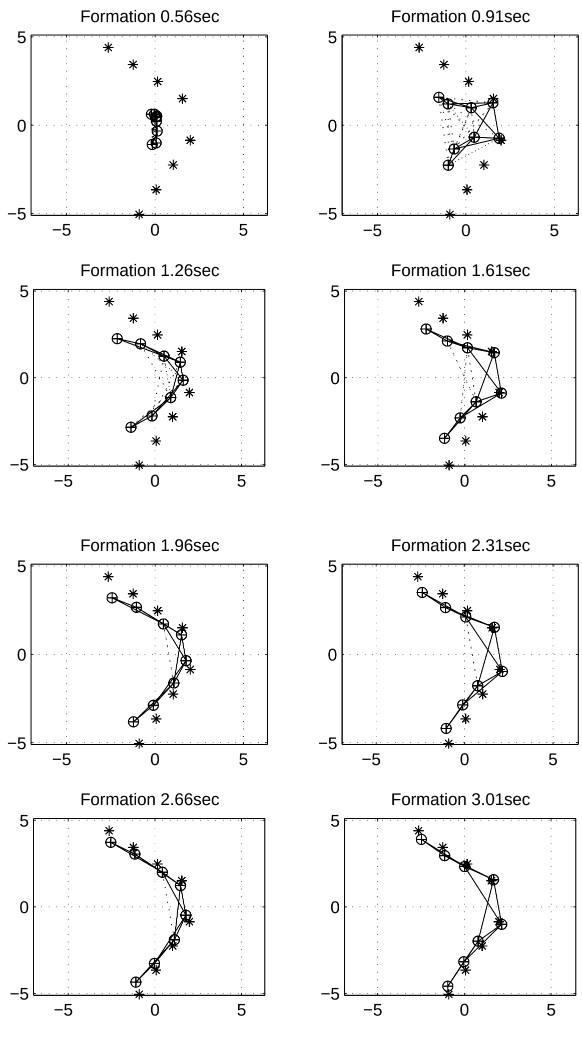

Fig. 5. Phase III: Target formation is achieved after 2.4 seconds. In the graph, circles denote the actual positions of the agents, while asterisks denote the target positions.

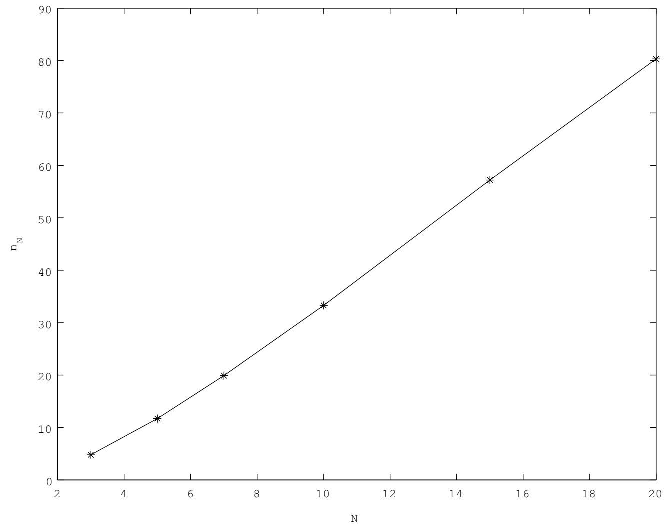

Fig. 6. Number of assignment related to N

solution, which implies that the complexity becomes $O(N^4)$. However this is not sufficient to determine the complexity of the problem since one need to ensure that d is large enough to capture the correct assignments. And to find this d is certainly not an easy task.

Finally, it should be noted that since the computations must be computed across the different agents, further complexity reductions should be possible through decentralization and/or parallelization of these computations. This endeavor is, however, left to the future.

Acknowledgement

The authors would like to thank Oleg Prokopyev and Panos Pardalos at the University of Florida for helpful discussion about the computational complexity issue.

References

[1] J. Lawton, R. Beard, B. Young, "A Decentralized Approach To Formation Maneuvers," *IEEE Transactions on Robotics and Automation*, Dec. 2003, Vol. 19, No. 6, pp. 933-941.

[2] M. E. Broucke, "Disjoint path algorithms for planar reconfiguration of identical vehicles," *Proceedings of the American Control Conference 2003*, Jun. 2003, Vol. 3, pp. 2199-2204.

[3] M. E. Broucke, "Reconfiguration of identical vehicles in 3D," *Proceedings of the 42nd IEEE Conference on Decision and Control 2003*, Dec. 2003, Vol. 2, pp. 1538-1543.

[4] R. E. Burkard, "Selected topics in assignment problems," *Discrete Applied Mathematics*, Nov. 2002, Vol. 123, pp. 257-302.

[5] J. P. Desai, J. P. Ostrowski, and V. Kumar, "Modeling and Control of Formations of Nonholonomic Mobile Robots", *IEEE Transactions on Robotics and Automation*, Dec. 2001, Vol. 17, No. 6, pp. 905-908.

[6] W. B. Dunbar and R. M. Murray, "Receding Horizon Control of Multi-Vehicle Formations: A Distributed Implementation," *Proceeding of 43rd IEEE Conference on Decision and Control 2004*, Dec. 2004, Vol. 2, pp. 1995 - 2002.

[7] M. Egerstedt and X. Hu, "Formation Constrained Multi-Agent Control," *IEEE Transactions on Robotics and Automation*, Dec. 2001, Vol. 17, No. 6, pp. 947-951.

[8] A. Muhammad and M. Egerstedt, "On the Structural Complexity of Multi-Agent Robot Formations," *Proceedings of the American Control Conference 2004*, Jun.-Jul. 2004, Vol. 6, pp. 4957-4962.

[9] T. Eren, P. N. Belhumeur, B.D.O. Anderson, and A.S. Morse, "A framework for maintaining formations based on rigidity," *Proceedings of the 15th IFAC World Congress*, Barcelona, Spain, July 2002, pp. 2752-2757.

[10] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, San Francisco, CA, 1979.

[11] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Transactions on Automatic Control*, Jun. 2003, Vol. 48, No. 6, pp. 988-1001.

[12] M. Ji and M. Egerstedt, "Connectedness Preserving Distributed Coordination Control over Dynamic Graphs," *Proceedings of the American Control Conference 2005*, Jun. 2005, pp. 93-98.

[13] P. Ören, M. Egerstedt, and X. Hu, "A Control Lyapunov Function Approach to Multi-Agent Coordination," *IEEE Transactions on Robotics and Automation*, Oct. 2002, Vol. 18, No. 5, pp. 847-851.

[14] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval Research Logistics Quarterly 2*, 1955, pp. 83-97.

[15] P. M. Pardalos, F. Rendl and H. Wolkowicz, "The Quadratic Assignment Problem", In "The Quadratic Assignment and Related Problems" (P.M. Pardalos and H. Wolkowicz, Ed.), Vol. 16 of DIMACS Series, American Mathematical Society, 1994, pp. 1-41.

[16] R. O. Saber and R. M. Murray, "Distributed Structural Stabilization and Tracking for Formations of Dynamic Multi-Agents," *Proceedings of the 41st IEEE Conference on Conference on Decision and Control 2002*, Dec. 2002, Vol. 1, pp. 209-215.

[17] R. O. Saber and R. M. Murray, "Agreement Problems in Networks with Directed Graphs and Switching Toplogy," *Proceedings of the 42nd IEEE Conference on Decision and Control 2003*, Dec. 2003, Vol. 4 , pp. 4126-4132.

[18] R. O. Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on Automatic Control*, Sept. 2004, Vol. 49, No. 9, pp. 1520-1533.