

## University of Windsor Scholarship at UWindsor

---

Computer Science Publications

Department of Computer Science

---

2007

# Multi-robot-based nanoassembly planning with automated path generation

Xiaobu Yuan

*University of Windsor*

Simon X. Yang

*University of Guelph*

Follow this and additional works at: <http://scholar.uwindsor.ca/computersciencepub>

 Part of the [Artificial Intelligence and Robotics Commons](#)

---

### Recommended Citation

Yuan, Xiaobu and Yang, Simon X.. (2007). Multi-robot-based nanoassembly planning with automated path generation. *IEEE/ASME Transactions on Mechatronics*, 12 (3), 352-356.

<http://scholar.uwindsor.ca/computersciencepub/10>

This Article is brought to you for free and open access by the Department of Computer Science at Scholarship at UWindsor. It has been accepted for inclusion in Computer Science Publications by an authorized administrator of Scholarship at UWindsor. For more information, please contact [scholarship@uwindsor.ca](mailto:scholarship@uwindsor.ca).

# Multi-Robot-Based Nanoassembly Planning with Automated Path Generation

Xiaobu Yuan<sup>1</sup> and Simon X. Yang<sup>2</sup>

<sup>1</sup>School of Computer Science, University of Windsor, Windsor, Ontario, N9B 3P4, Canada

<sup>2</sup>School of Engineering, University of Guelph, Guelph, Ontario, N1G 2W1, Canada

## Abstract

The construction of nanoscale structures is small in size, yet massive in volume. Such a massive construction of nanoscale structures requires a large number of autonomous nanorobots to work together. Presented in this paper is a novel approach of automated multi-robot nanoassembly planning. This approach uses an improved self-organizing map to coordinate assembly tasks of nanorobots while generating optimized motion paths at run time with a modified shunting neural network. It is capable of synchronizing multiple nanorobots working simultaneously and efficiently on the assembly of swarms of objects in the presence of obstacles and environmental uncertainty. Operation of the presented approach is demonstrated with experiments at the end of the paper.

## 1 Introduction

Nanotechnology works on the nanometer scale of molecules and atoms. It offers the promise of making a wide range of products that are nowadays impossible to make, and presents enormous potential in a vast range of new applications. Current research projects on nanotechnology have been mainly in a few areas. They focus respectively on the development of devices and systems to prepare tools on the invention of methods for the manipulation of nanoparti-

cles [24, 10, 31, 14], and on the formation of strategies that allow the automation of assembly and manufacturing at the nanometer scale [6, 11]. Applications of nanotechnology are also under active investigation [7, 4, 8].

Nevertheless, working on the nanometer scale presents new challenges that researchers have never faced before when they work on machinery parts in the macro world. Among the list of features that characterize nanoassembly, a complementary pair stands out as the most obvious and essential, i.e., the extreme smallness in size [29] and the massive volume in construction [27]. Aimed at tackling the problem associated with object manipulation in extremely small sizes, man-machine interfaces have been designed to bridge the macro and nano worlds mostly by means of virtual reality or augmented reality [3, 18].

The massive construction of nanoscale structures, however, requires large numbers of autonomous nanorobots to work together. Despite the remarkable progress in nanoscale manipulation, little work has been done in respect to the planning of multiple nanorobots. Presented in this paper is an investigation on the automation of multi-robot-based nanoassembly. A strategy is developed to efficiently synchronize multiple nanorobots while automatically generating collision-free paths. Simulation results demonstrate that this strategy is capable of coordinating multiple nanorobots with equally distributed

workload and automatically generated motion paths.

## 2 Related Work

Though there is hardly any report on the planning of multi-robot nanoassembly, significant work has been done in the coordination and cooperation of multiple robots in the macro world on the issues of task assignment and path planning. Task assignment is responsible for moving robots to certain locations for them to perform designated assembly tasks. In this regard a number of algorithms have been proposed, including graph matching [17], network simplex [26], distributed auction [32], dynamic Tabu searching [12], and genetic-based or agent-based algorithms [9, 1]. Efficient solutions were achieved without taking motion planning into consideration. As a result, robots have to wait until their destinations are finalized by an algorithm.

Path planning, in comparison, is responsible for the actual movements of robots. It controls a group of mobile robots to move from initial positions, visit intermediate locations, and reach final destinations. The motion paths need to be free of collisions, and the traveling distances should be short. There are quite a large number of studies on path planning, including the global method, potential field method, and genetic algorithm [2, 20, 34]. In addition to their limits to single-robot applications, most of them suffer from high computational complexity due to the search in a high-dimensional space for collision-free paths.

There is also a number of studies focused particularly on path planning of multiple robots. By combining with a genetic algorithm, a centralized technique was proposed to plan the motion paths of multiple cooperative robots [5]. Based upon the observation that the potential solution of each sub-problem is from and evolves only in its own sub-population, this technique introduces fitness functions of sub-population to reflect the interactions among all sub-problems.

Though the developed technique offers improved convergence rates and is suitable for parallel computation, the problem of high computational cost remains unsolved.

Several other studies applied the concept of self-organizing maps (SOMs) to the planning of multiple robots. An earlier work was presented in [16], which worked on path planning from initial to goal positions, but failed to consider the depots that have to be visited by one and only one of the robots. A couple of more recent papers also worked on the problem from the approach of SOM, one in concept [30] and another in practice [21]. The latter used an SOM-based action selection framework to control the action of multi-robots. The focus was on action selection and obstacle avoidance for robots to track targets, and intermediate depots were not considered.

Meanwhile, neural networks have proved to be useful in path planning. Path planning with penetration growth distance, for instance, shows the advantage in searching for collision-free paths [25]. There is a number of models developed for real-time motion planning through learning. For instance, combining an adaptive sensory-motor mapping model and an online visual error correction model may produce the trajectory of robot manipulators at run time [19]; and dynamic navigation of a mobile robot without any collisions is possible through unsupervised learning [23]. Since static or learning-based path planning do not react quickly to changes, they cannot handle the uncertainty in nanoassembly planning caused by disturbances and dust particles [29].

One of the major tasks of nanoassembly is to construct structures that form certain desired patterns [28]. In particular, the planning of nanoassembly involves different tasks, including object assignment, obstacle detection and avoidance, path finding, and path sequencing. An initial work on nanoassembly planning was reported in [22]. It used a number of existing techniques, and developed a practical method for automated nanoassembly planning. This method

works with a single nanorobot, and the motion paths are not optimal. Besides, its algorithm breaks down when the environment changes, for example, in the presence of foreign objects.

The rest of the paper first introduces a modified shunting neural network for automated generation of optimized collision-free paths in Section 3, and a mechanism of multi-robot coordination with an improved self-organizing map in Section 4. Section 5 then presents the problem statement, develops a strategy to links the improved self-organizing map with the modified shunting neural network, and discusses the steps involved in the process of automated nanoassembly planning. The last section before conclusions provides experiment results to examine issues involved in the study.

### 3 Optimized Path Generation

Presented in this section is a biologically inspired neural network that is capable of both dealing with environmental uncertainty of nanoassembly planning and generating optimized collision-free paths at run time.

#### 3.1 Biologically Inspired Neural Network

The original model of a biologically inspired neural network was proposed in [13], which used electrical circuit elements to describe a patch of membrane in biological neural systems. Its simplified model takes the form of (1), in which  $x_i$  is the neural activity of the  $i$ th neuron in the two-dimensional membrane. Parameters  $A$ ,  $B$ , and  $D$  are three nonnegative constants describing the passive decay rate, the upper and lower bounds of  $x_i$  respectively.  $S_i^e$  is the excitatory input and  $S_i^i$  is the inhibitory input to the neuron.

$$\frac{dx_i}{dt} = -Ax_i + (B - x_i)S_i^e(t) - (D + x_i)S_i^i(t) \quad (1)$$

Suppose a neuron  $N_q$  locates at a point  $\mathbf{q}$  in the three-dimensional network,  $\mathbf{q} = \langle q_1, q_2, q_3 \rangle$ .

It connects to all its  $n$  direct neighboring neurons  $N_{p_j}$ ,  $\mathbf{p}_j = \langle p_{j1}, p_{j2}, p_{j3} \rangle$  and  $1 \leq j \leq n$ . Following the notation in (1),  $x_q$  and  $x_p$  denote the neural activities of  $N_q$  and  $N_p$  respectively. A modification to (1) produces the following shunting equation that defines the dynamics of  $N_q$  [33].

$$\begin{aligned} \frac{dx_q}{dt} = & -Ax_q + (B - x_q)([I_q]^+ + \sum_{j=1}^n \omega_{qp_j} [x_{p_j}]^+) \\ & - (D + x_q)([I_q]^- + \sum_{j=1}^n \omega_{qp_j} c [x_{p_j} - s]^-) \quad (2) \end{aligned}$$

In the equation,  $I_q$  is the external inputs to  $N_q$ . The two functions  $[x]^+$  and  $[x]^-$  result in  $\max\{x, 0\}$  and  $\max\{-x, 0\}$ , respectively. Parameters  $A$ ,  $B$ , and  $D$  represent the passive decay rate, the upper and lower bounds of  $N_q$ , respectively. Parameter  $c$  is a constant in the range  $[0, 1]$ , and  $s$  is an adjustable safety factor. Especially, the symmetric weights  $\omega_{qp}$  are determined by a monotonically decreasing function  $f(|\mathbf{q} - \mathbf{p}|)$  of the Euclidean distance between  $\mathbf{p}$  and  $\mathbf{q}$ . For instance,  $f(a) = \mu/a$ , if  $0 < a < r_0$  for two positive constants  $\mu$  and  $r_0$ . Otherwise,  $f(a) = 0$ .

#### 3.2 Dynamic Path Generation

Every of the three-dimensional neural network maps to one of the three dimensions of the physical layout. For an assembly task, the neuron in the neural network that maps to initial position of a nanorobot is a starting neuron  $N_s$ , and the one that maps to the completion of the assembly task by the nanorobot is the target neuron  $N_t$ . All the other neurons classify into two types. One type, denoted by set  $\{N_c\}$ , includes all the neurons whose location maps to a collision between objects; the other,  $\{N_f\}$ , counters in the rest of neurons that lead to the collision-free movements of the nanorobot.

Different external inputs to the neurons then distinguish one type from another. The input  $I_q$  in (2) to neuron  $N_q$  is a large positive constant  $V$ ,  $V \gg B$ , if  $N_q$  happens to be the target  $N_t$ .  $I_q$

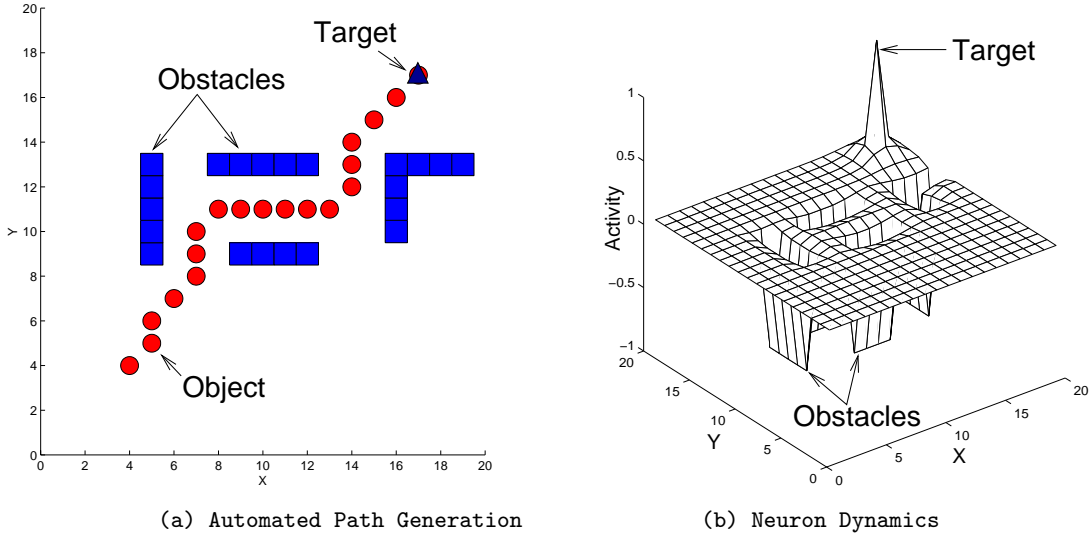


Figure 1: Realtime Generation of Optimized Collision-Free Paths

changes to  $-V$  if  $N_q$  is an element in  $\{N_c\}$ . Otherwise,  $I_q$  is  $0$  for all the neurons in the set of  $\{N_f\}$  (Fig. 1). As the starting neuron must be an element of  $\{N_f\}$ , its external input is always  $0$ . In addition, the stimuli within the receptive field of neuron  $N_q$  also include a sum of the weighted neural activities from its direct neighbors. In such a way, it allows the network to propagate positive neural activity through excitatory connections, and to restrain the negative activities through inhibitory connections.

Let  $N_s$  and  $N_t$  be the starting and target neurons respectively. Suppose  $N_p$  is the neuron whose neural activity  $x_p$  yields the biggest value among all the  $v$  neighboring neurons of  $N_s$ , i.e.,  $x_p = \max\{x_{sj}, j = 1, 2, \dots, v\}$ . The gradient ascent rule requires neuron  $N_s$  to move to the position of neuron  $N_p$ . As shown in Fig. 1(a), this movement maps to the robot space as a step of moving the object around in the environment. By following the gradient ascent rule and adaptively changing the current position of  $N_s$ , the neural network globally guides the nanorobot to push the object towards its goal while avoiding collisions.

The influence of parameters  $c$  and  $s$  in (2) on path generation is the clearance distance from

obstacles. They determine the relative strength and the threshold of the negative neural connections respectively. This neural network does not suffer from local minimum, even in a complicated maze-type environment of many deadlock situations. The neural activity propagation from the target to the starting position always generates smooth, continuous, and optimized paths from starting to destination locations with obstacle clearance. Moreover, path generation reacts at run time to any changes in the environment.

## 4 Multi-Robot Coordination

The construction of nanoscale structures requires a large number of nanorobots working together, and nanoassembly planning needs to efficiently coordinate their work. This section describes the structure of an improved self-organizing map and a mechanism for multi-robot coordination.

### 4.1 An Improved Self-Organizing Map

A self-organizing map (SOM) combines a competitive learning principle with a set of nodes that are topologically structured in a way that

the adjacent nodes tend to have similar weight vectors [15]. In the classical application of pattern recognition, an SOM is a two-layer neural network that consists of an input layer and an output layer. The input layer has  $F$  nodes, where  $F$  is the number of pattern features; and the output layer has  $M$  nodes, where  $M$  is the number of results after classification. Each of the nodes in the input layer connects to all nodes in the output layer.

The connection strengths of an output node  $R_j$  are given by an  $F$ -dimensional weight vector  $\mathbf{r}_j = \langle r_{j1}, \dots, r_{jF} \rangle$ , where  $1 \leq j \leq M$ . The learning algorithm of an SOM neural network ensures that the most highly (or lowly) activated node wins the competition. Meanwhile, the weight vectors of the winner and its neighbors change from their initial values toward the feature values of the input pattern. The neural network is self-organizing in such a sense that the output nodes tend to attain weight vectors that capture the characteristics of the input vector space.

Assume that there are  $K$  nanorobots working simultaneously on the task of pushing  $N$  objects from their initial locations to  $M$  destination locations in the three-dimensional space, where  $K < M < N$ . As shown in Fig. 2(a), the self-organizing map can be improved in such a way that it consists of three nodes  $x_i$ ,  $y_i$ , and  $z_i$  in the input layer, which take in respectively the three Cartesian coordinates  $p_{ix}$ ,  $p_{iy}$ , and  $p_{iz}$  of the  $i$ th destination location  $P_i$ ,  $1 \leq i \leq N$ . In its output layer, there is a set of two-dimensional nodes  $R_{lj}$ , where  $1 \leq l \leq K$  and  $1 \leq j \leq M$ . An weight vector  $\mathbf{r}_{lj}$  connects the three input nodes to an output node  $R_{lj}$  with  $r_{ljx}$ ,  $r_{l jy}$ , and  $r_{ljz}$ .

Fig. 2(b) illustrates the geometrical meaning of the nodes in the two-layer neural network. In the figure, the red circle represents the  $i$ th destination location  $P_i$ ,  $1 \leq i \leq N$ , which is supposedly the current input node under processing. Small green circles represent output nodes  $R_{lj}$ ,  $1 \leq l \leq K$  and  $1 \leq j \leq M$ . Moreover, unfilled circles represent unprocessed destination

locations, and blue blocks stand for obstacles. For any of those non-obstacle items, a projection to the axes identifies the corresponding coordinates. For example, the projected coordinates for  $P_i$  are  $x_i$  and  $y_i$  in the two-dimensional illustration. Similarly,  $R_{lj}$  projects to  $r_{ljx}$  and  $r_{l jy}$  in the figure.

## 4.2 Coordinating Mechanism

A fixed point  $l'$  on the  $l$ -axis of the output layer in Fig. 2(a) groups together the  $M$  nodes on the  $m$ -axis, i.e.,  $R_{l'j}$  for  $j = 1, \dots, M$ . These  $M$  nodes are initialized in such a way that they are distributed on an imaginary line that virtually links the  $M$  output nodes from the starting location to the final location of the  $l$ th nanorobot. When a destination location  $P_i$  is presented to the two-layer neural network,  $1 \leq i \leq N$ , all the  $KM$  nodes in the output layer participate in competition, and a winner  $R'_{lj}$  is determined according to the following criterion.

$$R'_{lj} \Leftarrow S'_{lj} = \min\{S_{lj} \mid l = 1, \dots, K; j = 1, \dots, M; R_{lj} \in \Omega\} \quad (3)$$

In (3),  $R_{lj}$  is the  $j$ th node of the  $l$ th virtual line;  $\Omega$  is the set of output nodes on the  $l$ th virtual line that have not been a winner before; and  $S'_{lj}$  is the minimum of  $S_{lj}$  whose definition is given below,

$$S_{lj} = S(P_i, R_{lj})(1 + T) \quad (4)$$

where  $S(P_i, R_{lj})$  is the distance between  $P_i$  and  $R_{lj}$  along the path generated by the shunting equation (2) in Section 3. The term  $T$  in (4) equalizes the workload distribution of nanorobots. The workload of nanorobots has different emphases in different applications, and a definition of  $T$  is given below that particularly takes the traveling distance of nanorobots into the consideration of workload,

$$T = \frac{L_l - V}{1 + V} \quad (5)$$

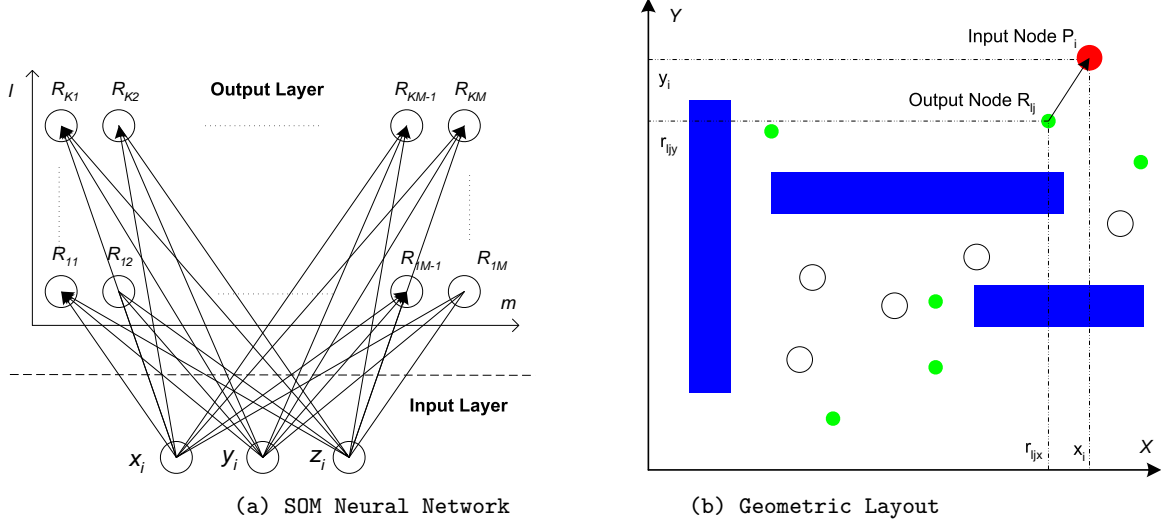


Figure 2: SOM-based Neural Network and Its Geometric Representation

where  $L_l$  is the length of the  $l$ th virtual line,  $1 \leq l \leq K$ ; and  $V$  is the average path length.

$$V = \frac{1}{K} \sum_{l=1}^K L_l \quad (6)$$

In addition, the neural network updates itself by revising its weight vectors  $\mathbf{r}_{lj}$ ,  $1 \leq l \leq K$  and  $1 \leq j \leq M$ , according to the following rule,

$$\mathbf{r}_{lj}(t+1) = \begin{cases} \mathbf{p}_i, & \text{if } S(P_i, R_{lj}) < \eta\rho \\ \mathbf{r}_{lj}(t) + \beta f(d_{lj})(\mathbf{p}_i - \mathbf{r}_{lj}(t)), & \text{otherwise} \end{cases} \quad (7)$$

where  $\eta$  is a small constant normally less than 0.5;  $\beta$  is the learning rate;  $\rho$  is the minimum distance between any two nodes of the target locations; and  $f(d_{lj})$  is a neighborhood function whose definition is given below,

$$f(d_{lj}) = \begin{cases} e^{-d_{lj}^2/G^2}, & \text{if } d_{lj} < \gamma M \\ 0, & \text{otherwise} \end{cases} \quad (8)$$

and,

$$G = (1 - \alpha)^t G \quad (9)$$

In (8),  $d_{lj}$  is the distance measured along the  $l$ th virtual line between the  $j$ th node and the

winner node  $R'_{lj}$ ;  $G$  is the gain constant with an initial value of, for example, 10; and  $\gamma$  is a small constant indicating the range of neighborhood, which is normally less than 0.4. In (9),  $t$  is the number of iterations, and  $\alpha$  is the gain changing rate. The smaller the  $\alpha$  is, the longer time the computation takes, and the shorter distance of the total paths of all the nanorobots will be. The normal range of  $\alpha$  is between 0.001 and 0.05.

## 5 Automated Nanoassembly Planning

When a single nanorobot works on the formation of patterns or construction of particles, nanoassembly planning focuses on object assignment, and the searching and sequencing of collision-free motion paths. With the involvement of multiple nanorobots, the issue of coordination becomes crucial in planning. This section develops a seamless approach of automated multi-robot-based nanoassembly planning.

### 5.1 Problem Statement

*Suppose that the working environment of a nanoassembly planning system*

consists of a set of obstacles, a collection of  $N$  identical objects, and a team of  $K$  nanorobots, all of which locate at known positions in three-dimensional space. Given a cluster of  $M$  locations that defines the pattern of a product, an assembly task instructs a nanorobot to reach an object and push it from its initial position to a destination along a collision-free path with a predetermine distance from obstacles.

With the assumption of  $K < M < N$ , multi-robot nanoassembly planning dynamically coordinates the team of nanorobots to complete the assembly tasks of constructing the product with a minimal or near-minimal total cost. When the cost of nanoassembly tasks is evaluated in terms of traveling distances, the cost for each individual nanorobot is the distance it travels from its initial to final position. The total cost is the sum of costs of all the nanorobots.

## 5.2 Planning Process

Fig. 3 highlights the steps in the presented approach of multi-robot nanoassembly planning. The automated planning starts with an initialization to the modified shunting neuron network (Section 3). According to the location of obstacles in the 3D space, inputs to their corresponding neurons in the network are set to a large negative constant  $-V$ . All other neurons take  $\theta$  as their input, and each of the  $K$  nanorobots takes a position in the neuron network in line with their spatial occupations. Afterwards, the improved SOM neuron network is initialized in the way as described in the previous subsection. In addition to the initialization of parameters  $\eta$ ,  $\beta$ ,  $G$ ,  $\gamma$ , and  $\alpha$ , the connecting weights from the three input nodes to the output nodes are set to form  $K$  virtual lines.

The remaining steps operate in two phases. In

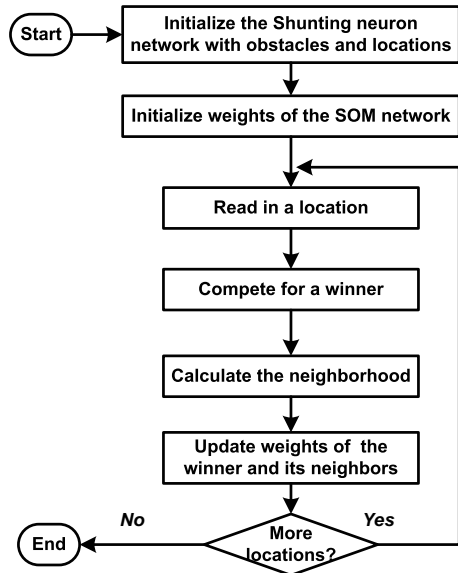


Figure 3: A Process of Multi-Robot Nanoassembly Planning

the first phase, coordinates of the  $N$  objects are fed into the SOM neuron network one set after another, and the  $K$  nanorobots compete to grasp the objects. In the second phase, coordinates of the  $M$  destination locations are fed into the SOM neuron network one by one, and the nanorobots compete to push the objects to the destinations. When the number of nanorobots is less than the number of objects, i.e.,  $K < N$ , this two-phase operation repeat to deal with unprocessed objects.

The same SOM network is used in both phases to manage the nanorobots, but there is a need to initialize the network again before the second phase starts. In either of the two phases, the coordinate input of a position to the SOM neuron network sets a large positive constant  $V$  to the input at the matching neuron in the modified shunting neuron network. This input stimulates the shunting network to generate collision-free paths at realtime from all nanorobots to the position.

The distances along the paths are used in (4) for nanorobot competition as regulated by (3), and the winner follows its path to reach the object at the location or push the object



to a destination. As illustrated in Fig. 3, automated nanoassembly planning then computes the neighborhood with (8), and updates the weights of nanorobots in the SOM neuron network with (7). This process continues in the first phase until all the objects are accompanied by a nanorobot, or in the second phase until all the objects are pushed to a destination.

### 5.3 Discussions

Nanorobots and nanoparticles are so tiny that their sizes are almost negligible in the physical environment. Nanoassembly, therefore, allows path planning to treat them as circular dots, each of which occupies a point in space. This observation greatly simplifies the work when the presented multi-robot nanoassembly planning initializes the modified shunting neural network for the generation of collision-free paths. The input to a neuron in the network can simply be determined by checking the corresponding location in space.

The paths of nanorobot movements are piecewise linear, which is consistent with single-robot nanoassembly planning as characterized in [22]. Parameters  $c$  and  $s$  in (2) determine the relative strength and the threshold of the negative neural connections respectively. Optimality of path generation refers to the ability of (2) to adjust parameters  $c$  and  $s$  for automatic generation of smooth, continuous, and collision-free motion paths, without being too close to or too far from obstacles.

The computational complexity of path generation linearly depends on the size of the modified shunting neural network. This neural network does not suffer from local minimum, even in a complicated maze-type environment of many deadlock situations. In addition, its underlying dynamic neural activity operates without explicitly searching over the free workspace, without explicitly optimizing any cost functions, without any prior knowledge of a dynamic environment, and without any learning procedures. It is therefore capable of dealing with environmental un-

certainty in the presence of dust particles.

The process of nanoassembly planning links nanorobots coordination with real-time path generation. The SOM structure given in Fig. 2 treats nanorobots as nodes in the output layer. This structure allows the number of nanorobots to change during the coordination process, which is a useful feature that helps to handle situations when some nanorobots break down or some more are added. The coordinates of the input layer are also changeable. It further enables the presented approach of nanoassembly planning to deal with environment changes due to shifted or moving targets caused by disturbances.

## 6 Simulation Results

Simulation has been conducted to examine the presented approach of multi-robot nanoassembly planning. Programs were coded with MATLAB on a personal computer running Windows XP, and test cases covered both multi-robot path planning without obstacles and multi-robot nanoassembly planning in the presence of obstacles. Experiment results given below are in 2D cases for the simplicity of illustration, but the presented approach works with higher dimensions as both the modified shunting neural network and the improved self-organizing map are structured for 3D applications.

Between the two neuron networks that work together in automated multi-robot nanoassembly planning, the core is the improved SOM network as it is in charge of coordinating the operations of multiple nanorobots. The first group of experiments, therefore, concentrated on checking the capability of the SOM network in workload equalization when a smaller number of nanorobots are involved in operation. In particular, suppose that there are three nanorobots originally placed at the ‘ $S$ ’ location in Fig. 4(a), and they return to the ‘ $S$ ’ location after all the marked positions have been visited by one and only one nanorobot.

The ‘ $S$ ’ point in Fig. 4(a) is also the loca-

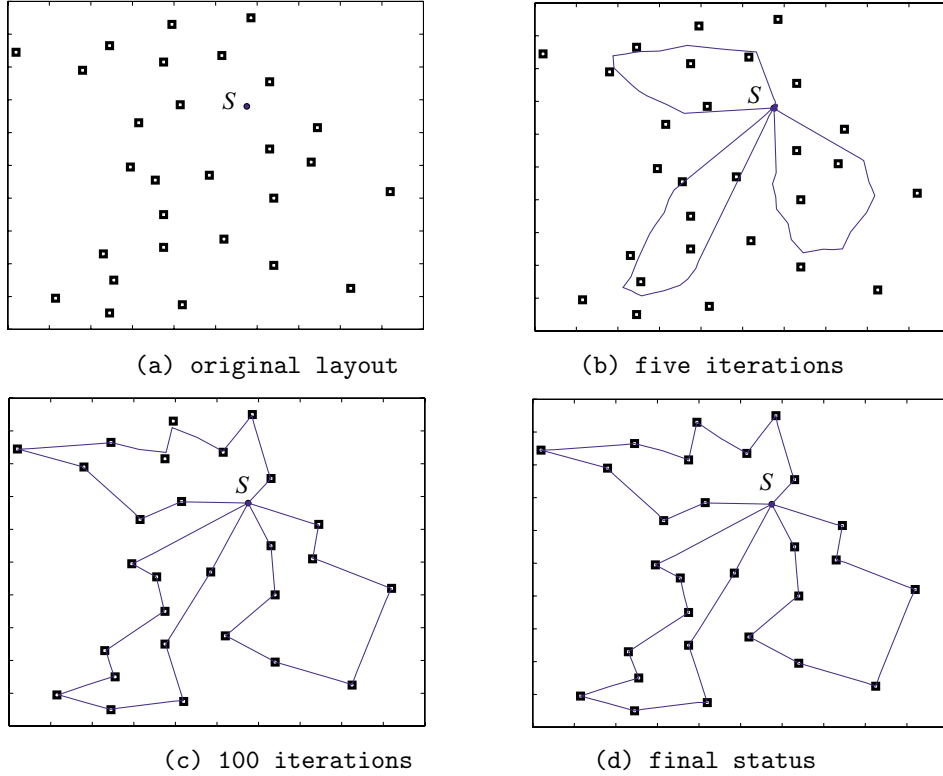


Figure 4: An Experiment of Path Planning

tion where all the three groups of  $M$  output nodes along three virtual lines meet together at the initialization of the SOM neuron network. These three virtual lines then gradually stretch out in the self-organizing process, moving the nodes towards the marked positions. The geometric meaning of virtual lines becomes clear after they start to form three rings in Fig. 4(b) after several iterations of weight adjustment. This process continues to allow one node on one of the virtual lines reaching to one marked position (Fig. 4(c)), and it ends when each of the marked positions is visited by one node (Fig. 4(d)).

When a test considers neither the pushing operation of nanoassembly nor the presence of obstacles as in the case of Fig. 4, it becomes an extremely simplified case of path planning for multiple robots without physical shape. The results demonstrate the SOM network's ability to equally distribute workload. Tests in this group

also indicate that the number of required iterations mainly depends on the parameter of gain change rate  $\alpha$ . A total of 160 iterations are needed in the experiment with  $\alpha$  being set to 0.03. In comparison, the self-organizing process is not much sensitive to the learning rate  $\beta$ , neighborhood parameter  $\gamma$ , and parameter  $\eta$  when they are in the ranges of  $[0.05, 0.1]$ ,  $[0.1, 0.4]$ , and  $[0.01, 0.1]$  respectively.

The second group of experiments tested the presented approach of nanoassembly planning on the setting of Fig. 5(a), which is similar to the one used in [22]. The presence of obstacles puts the modified shunting neural network in work to generate collision-free paths. In addition to the initialization of SOM network parameters, the parameters of the shunting network are initialized so that the passive decay rate  $A$ , upper bound  $B$ , and lower bound  $D$  are set to 10.0, 1.0, and 1.0 respectively. The other parameters are

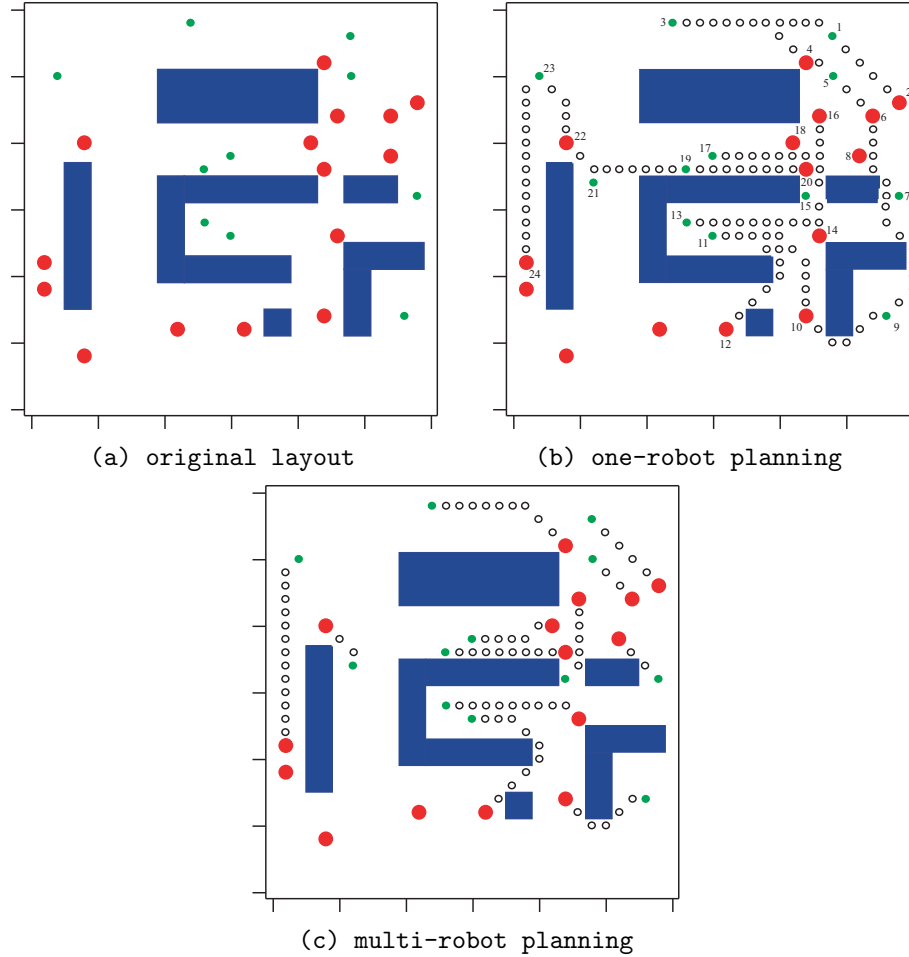


Figure 5: An Experiment of Nanoassembly Planning

set in such a way that  $\mu = 1.0$ ,  $c = 0.9$ ,  $s = -0.7$ , and  $r_0 = 2.0$  for the neighborhood connections, and  $V = 100.0$  for the external inputs.

Fig. 5(b) depicts the operation of a single nanorobot in action. By following the optimized collision-free paths, the nanorobot pushes all objects one after another to the nearest destinations. The numbers indicate the order of pushing operations, and the little circles illustrate the trace of object/nanorobot movements along the paths. Fig. 5(c) finally demonstrate the operation of multi-robot nanoassembly planning. Assuming enough nanorobots and sufficient destinations, all the nanorobots work together pushing the objects to their destinations along collision-free paths.

## 7 Conclusion

This paper presents an approach of automated planning of multi-robot-based nanoassembly. It uses an improved self-organizing map (SOM) to coordinate multiple nanorobots for them to work simultaneously with equally distributed workload, and a modified shunting neuron network to generate optimized collision-free paths at run time with adjustable safety distance to obstacles. This approach fits all the tasks of multi-robot nanoassembly planning into a seamless process, and makes it possible for realtime handling of environmental uncertainty. Continuous research is under active investigation to examine its application in practice for further refinement and

enhancement.

## References

- [1] R. Akkiraju, P. Keskinocak, S. Murthy, and F. Wu. An agent-based approach for scheduling multiple machines. *Applied Intelligence*, 14:135–144, 2001.
- [2] K. Al-Sultan and D. Aliyu. A new potential field-based algorithm for path planning. *Journal of Intelligent and Robotic Systems*, 14(5):657–662, Oct 1996.
- [3] B. Aruk, H. Hashimoto, and M. Sitti. Man-machine interface for micro/nano manipulation with an afm probe. In *Proc. 2001 1st IEEE Conference on Nanotechnology*, pages 151–156, Oct. 2001.
- [4] D. Bell, L. Dong, Y. Sun, L. Zhang, B. Nelson, and D. Grutzmacher. Manipulation of nanocoils for nanoelectromagnets. In *Proc. 5th IEEE Conference on Nanotechnology*, pages 149–152, 2005.
- [5] Z. Cai and Z. Peng. Cooperative co-evolutionary adaptive genetic algorithm in path planning of cooperative multimobile robot systems. *Journal of Intelligent and Robotic Systems: Theory and Applications*, 33(1):61–71, 2002.
- [6] A. Cavalcanti. Neural motion and evolutionary decision in robotic competition applied for molecular machine system design. In *Proc. 2002 IEEE International Symposium on Computer Aided Control System Design*, pages 115–125, Sept. 2002.
- [7] A. Cavalcanti. Assembly automation with evolutionary nanorobots and sensor-based control applied to nanomedicine. *IEEE Transactions on Nanotechnology*, 2(2):82–87, June 2003.
- [8] R. Chau, S. Datta, M. Doczy, B. Doyle, B. Jin, J. Kavalieros, A. Majumdar, M. Metz, and M. Radosavljevic. Benchmarking nanotechnology for high-performance and low-power logic transistor applications. *IEEE Transactions on Nanotechnology*, 4(2):153–158, March 2005.
- [9] P. Chu and J. Beasley. A genetic algorithm for the generalised assignment problem. *Computers and Operations Research*, 24(1):17–23, 1997.
- [10] S. Feng, Y. Lee, and K. Lyons. An open framework for the assembly of micro- and nano-scale artifacts. In *Proc. 5th IEEE International Symposium on Assembly and Task Planning*, pages 72–78, July 2003.
- [11] T. Fukuda and F. Arai. Prototyping design and automation of microhano manipulation system. In *Proc. 2000 IEEE International Conference on Robotics and Automation*, volume 4, pages 192–197, April 2000.
- [12] A. Higgins. A dynamic tabu search for large-scale generalised assignment problems. *Computers and Operations Research*, 28:1039–1048, 2001.
- [13] A. Hodgkin and A. Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *Journal of Physiology London*, 117:500–544, 1952.
- [14] S. Kaya. COSMOS — a novel MOS device paradigm. *IEEE Transactions on Nanotechnology*, 4(5):588–593, Sept. 2005.
- [15] T. Kohonen. *Self-Organizing Maps*. Springer, Berlin, 1995.
- [16] F. Kolushev and A. Bogdanov. Neural algorithms of path planning for mobile robots in transport systems. In *Proc. IEEE International Joint Conference on Neural Networks*, pages 4381–4386, July 1999.

- [17] K. Kwok, B. Driessen, C. Phillips, and C. Tovey. Analyzing the multiple-target-multiple-agent scenario using optimal assignment algorithms. *Journal of Intelligent and Robotic Systems*, 35(111-122), 2002.
- [18] G. Li, N. Xi, H. Chen, C. Pomeroy, and M. Prokos. “videolized” atomic force microscopy for interactive nanomanipulation and nanoassembly. *IEEE Transactions on Nanotechnology*, 4(5):605–615, Sept. 2005.
- [19] L. Li and H. Ogmen. Visually guided motor control: Adaptive sensorimotor mapping with online visual-error correction. In *Proceedings of the World Congress on Neural Networks*, pages 127–134, 1994.
- [20] Z. Li and T. Bui. Robot path planning using fluid model. *Journal of Intelligent and Robotic Systems*, 21:29–50, 1998.
- [21] K. Low, W. Leow, and J. Ang. Action selection for single- and multi-robot tasks using cooperative extended kohonen maps. In *Proc. IEEE 18th International Joint Conference on Artificial Intelligence*, pages 1505–1506, August 2003.
- [22] J. Makaliwe and A. Requicha. Automatic planning of nanoparticle assembly tasks. In *Proc. 4th IEEE International Symposium on Assembly and Task Planning*, pages 288–293, May 2001.
- [23] F. Muniz, E. Zalama, P. Gaudio, and J. Lopez-Coronado. Neural controller for a mobile robot in a nonstationary environment. In *Proceedings of 2nd IFAC Conference on Intelligent Autonomous Vehicles*, pages 279–284, 1995.
- [24] S. Nof, editor. *Handbook of Industrial Robotics*. Wiley & Sons, second edition, 1999.
- [25] C. Ong and E. Gibert. Robot path planning with penetration growth distance. *Journal of Robotic Systems*, 15(2):57–74, 1998.
- [26] J. Orlin. A polynomial-time primal network simplex algorithm for minimum cost flows. In *Proc. 7th Annual ACM/SIAM Symposium on Discrete Algorithms*, pages 474–481, 1996.
- [27] A. Requicha. Nanorobots, NEMS, and nanoassembly. *Proceedings of the IEEE*, 91(11):1922–1933, 2003.
- [28] A. Requicha, C. Baur, A. Bugacov, B. Gazen, B. Koel, A. Madhukar, T. Ramachandran, R. Resch, and P. Will. Nanorobotic assembly of two-dimensional structures. In *Proc. 1998 IEEE International Conference on Robotics and Automation*, volume 4, pages 3368–3374, May 1998.
- [29] T. Sato. Micro/nano manipulation world. In *Proc. 1996 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 834–841, 1996.
- [30] W. Shen, C. Chuong, and P. Will. Simulating self-organization for multi-robot systems. In *Proc. IEEE International Conference on Intelligent and Robotic Systems*, pages 2776–2781, October 2002.
- [31] G. Skidmore, M. Ellis, A. Geisberger, K. Tsui, K. Tuck, R. Saini, T. Udeshi, M. Nolan, R. Stallcup, and J. Von Ehr. Assembly technology across multiple length scales from the micro-scale to the nano-scale. In *Proc. 17th IEEE International Conference on Micro Electro Mechanical Systems*, pages 588–592, 2004.
- [32] J. Wein and S. Zenios. Massively parallel auction algorithms for the assignment problem. In *Proc. 3rd Symposium on Frontiers of Massively Parallel Computation*, pages 90–99, Nov. 1990.
- [33] X. Yuan and S. Yang. Virtual assembly with biologically inspired intelligence. *IEEE Transactions on Systems, Man, and Cybernetics*, 33(2):159–167, 2003.

- [34] D. Zhu and J. Latombe. New heuristic for efficient hierarchical path planning for mobile robot. *IEEE Transaction on Robotics and Automation*, 7(1):9–20, 1991.