Virtual Engineering: Optimal Cell Layout Method for Improving Productivity for Industrial Robot

Dandan Zhang, Liwei Qi
ABB Corporate Research China
No. 31 Fu Te Dong San Rd., Waigaoqiao Free Trade Zone, 200131
Shanghai, P. R. China
emily-dandan.zhang@cn.abb.com, levy-liwei.qi@cn.abb.com

***Abstract*—This paper presents a method to optimize the positions of robot tasks in a robotic work cell in terms of minimum cycle time. In order to improve the efficiency of the overall algorithm, the method is proposed to be decomposed into 3 stages. Firstly, a good position for each task is derived by putting such robot task to a preferred region; secondly, based on the results from the first stage, a switched method is designed to seek the best space sorting of the tasks; with the benefit from these two stages which can reduce the search space greatly in optimization, in the third stage, the positions of all robot tasks are adjusted simultaneously by means of Simulated Annealing Method. Several test cases verified the effectiveness of the proposed method.**

I. INTRODUCTION

In automated manufacturing processes, such as welding, painting, cutting and material handling, a robot always performs a repetitive operation sequence along a predefined path. The productivity of a robot can be improved considerably by reducing the cycle time for completing an operation sequence. The cycle time of a given robot depends on many factors, in which the positions of the tasks that the robot will perform are very important. The cycle time can be reduced significantly provided with a good work cell layout.

In recent years, several research work has been done to address the cell layout problem. Basically, these methods can be categorized into the following groups. 1) Adjusting the position of the robot [1][3][4]. 2) Optimizing the position of one task relative to the robot [6]. 3) Optimizing the positions of several tasks relative to the robot [2][5]. 4) Designing robot tool. 5) Placing external axes relative to the robot or to each other [6]. For the first method group, In [1], a system for optimizing the assembly work cell layout in the context of industrial robotic CAD software products was given. The idea is to find the free acceptable domain according to the obstacle positions, and then the position of robot is optimized using Simulated Annealing (SA) method. In [3], a simple method for deciding the robot layout was presented, but only workable layout was given, and no layout optimization was involved. Another work from [4] aimed to find an optimal robot placement relative to a task in terms of minimal cycle time, robot performance from kinematics and kinetics point of view. The method need firstly use several provisional robot positions, from which the index for robot performance can be defined, and hence to conduct optimization. This method considered both cycle time and robot performance, however, it can only handle one single task, not several sub-tasks together, and hence can only search optimal solution based on an existing task, so it is not adaptive in the cell layout optimization involving several subtasks. For the second method group, in[6], an approach was given trying to optimize the cycle time of a robot when executing one task based on a platform simulating the real robot. The idea was to optimize the position of the task using experiment-based method. The polynomial fitting method was employed to find the optimal task position. Although the time cost is relatively low, it is not easy to use a low degree polynomial function to fit the function between cycle time and task position with limited sampling points. When two or more tasks are involved, the polynomial fitting method can't be expected to find even better layout. For the third method group, in [2], another system was presented which optimized the positions of the machines relative to the robot with SA approach. This system can work with a real-time objective function on CAD software platform, but the machines were put into consideration one by one, not handled equally together. In fact, a good position for one machine can not be guaranteed still good if another machine is put into consideration. While in F. Masmoudi's work [5], the station sorting and robot travel sequencing problems were addressed, but the robot performance factor was not considered, which is important to reduce the cycle time.

Based on above analysis, to find optimal cell placement involving multiple sub tasks for a robot remains a valuable topic regarding shorten cycle time and improve robot performance. In the presented method, the positions of multiple tasks are optimized in terms of minimal robot cycle time. To be time efficient, the method is decomposed into 3 main stages. Firstly, each robot task is put into a defined preferred region to derive good initial positions; then the best space sorting of the tasks is obtained through a switched method; finally, the positions of all the tasks are adjusted simultaneously in a greatly reduced search space by means of SA. The given method was compared to other existed approach through several test cases and the results demonstrated the effectiveness of the method.

The rest of the paper is organized as follows. Section II presents the detailed cell layout method. In Section III, several cases are given to test the proposed method. Section IV concludes the paper.

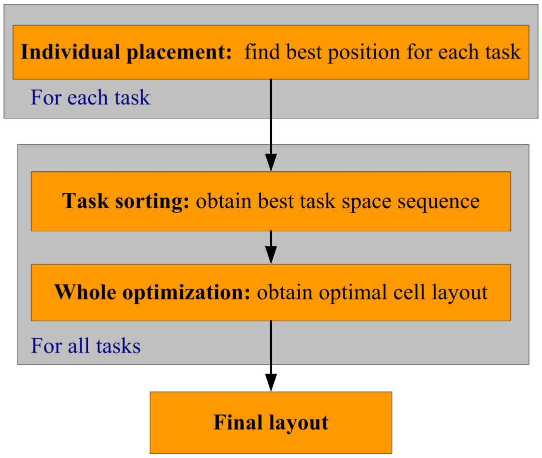


Fig. 1. Solution map of optimal cell layout method.

II. OPTIMAL CELL LAYOUT METHOD

A. Solution map

Considering the time efficiency of the whole algorithm, the cell layout method is decomposed into three stages: 1) Individual task placement, 2) Task sorting and 3) Whole optimization. Through the first two stages, a near-optimal layout can be obtained in short time, which is very beneficial in reducing search space for the whole optimization involving all tasks in the final stage.

B. Individual task placement

In Individual task placement, each task is placed into a *Preferred region* and then rotated along its own frame to get the best position. Firstly the definition of *Preferred region* and also *robot target* are given.

Preferred region: Based on statistics analysis, most of the robot tasks are placed in a concentrated region within the robot reachable range. This region can guarantee good robot performance in terms of robot kinematics and kinetics. The region is called as *Preferred region*, shown in Fig. 2. If a robot task is put into *Preferred region*, good robot performance, and hence reduced cycle time can be expected.

Robot target: Robot tasks are defined using targets. A target can be viewed as an infinitely small object with three coordinates defining its position and three angles defining its orientation in a three dimensional Cartesian space. Changing the position of a task is the same thing as changing all targets used to define the task[6].

With above definitions, Individual task placement can be divided into two steps.

Step 1: Putting the task into *Preferred region*. When putting a task into *Preferred region*, the following criterion are used:

$$Pos_A = \arg(\min(\sum_{i=1}^{N_t} dis(T_i, P))), \qquad (1)$$

in which Pos_A is the derived position of the task, N_t denotes the target number involved in the task, T_i denotes target i, P is the center of *Preferred region*, and $dis(T_i, P)$ is the Euclidean distance of T_i to P.

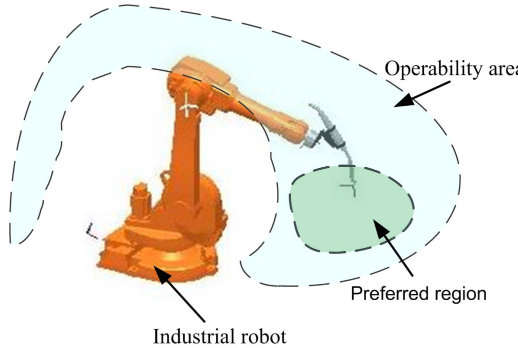


Fig. 2. Preferred region of the robot.

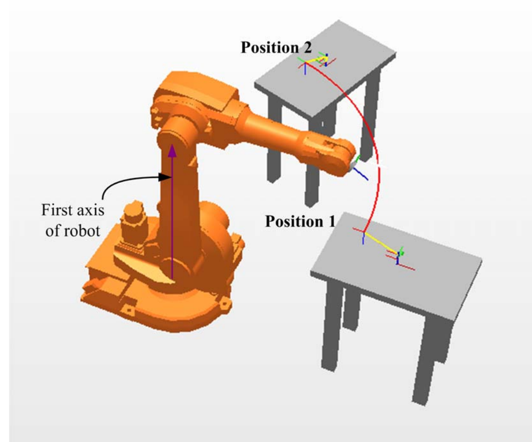


Fig. 3. Equivalent positions of a task relative to the robot in terms of cycle time.

Step 2: Rotating the task along its own frame to find best orientation of the task. Based on **Step 1**, the task orientation is adjusted according to the following criterion:

$$Pos_B = \arg(\min(CT(\alpha))), \qquad (2)$$

in which Pos_B is the derived position; α is the orientation of the task, $\alpha \in [-\pi, \pi]$ (the adjustment range of α can be changed according to the actual work cell restrictions); $CT(\alpha)$ is the obtained cycle time as the function of orientation α. It is to mention that even when lacking a mathematical model of the robot, the best orientation of the task can be obtained in short time through performing a small group of experiments based on the sampling values of α via running robot controller to execute the task.

According to the above two steps, the best position for each individual task can be obtained.

C. Task sorting

When the best position of each individual task is obtained, it comes to a question that how to put all tasks into consideration for a best overall work cell layout. It is assumed and verified

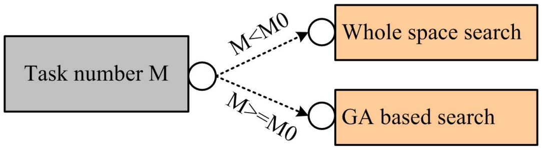


Fig. 4. Switched method for task sorting.

through experiments that the best position of a task will remain best when only the first axis of the robot rotates for tending the task. As shown in Fig. 3, when the task position is changed from Position 1 to Position 2, the task is still in the best position for the robot to tend. Task sorting is to find the best space sequence of the tasks provided with a logical visiting sequence. To provide a fast sorting, a switched method is designed here, shown in Fig. 4. Suppose there are M tasks to be sorted. Then there are totally $M!$ possible solutions for task sorting. When M is small, searching in the space with dimension of $M!$ is possible. However, if M is large, searching the whole solution space is quite time consuming. In such a case, the sorting method switches to a simplified Genetic algorithm (GA) method to seek the best sequence of tasks, since GA has been found to be an effective optimization method when solving combinational optimization problems with large dimension.

1) Evaluation function: As stated above, if a task rotates only around the first axis of the robot, it will not influence the cycle time when robot executing the task. In such a case, the solution for task sorting can be concluded: all tasks are put into the work cell together. Each task keeps its position and orientation obtained in stage 1 but may rotate with different angles around the first axis of the robot. The space sequence of all tasks is adjusted for minimizing the travelling distance when robot executes all tasks according to the predefined logical visiting sequence. The measurement for the travelling distance is expressed as:

$$Fn(s) = \sum_{i,j} dis(Task_i, Task_j), \tag{3}$$

where $Fn(s)$ is the evaluation of sequence s; i is the ID number of task i; j is the ID number of the task next to task i that the robot will visit; $dis(stn_i, stn_j)$ is the robot travel distance from task i to task j. Equation (3) is a fast calculation considering the present computer capacity. This brings great advantage in optimization method selection which may involve relatively large search space.

2) GA principle: As is known, GA is an efficient stochastic and non-gradient search method [7][8][9]. It manipulates iteratively a population of chromosomes, which encode a set of possible solutions, to obtain solutions with better performance. The variables of a problem are represented by genes in the chromosome, and chromosomes are evaluated by the fitness function. Recombination typically involves two genetic operators: 1) crossover and 2) mutation. The offspring are created through altering the compositions of the genes by genetic operators. After several generations, GA can converge to the

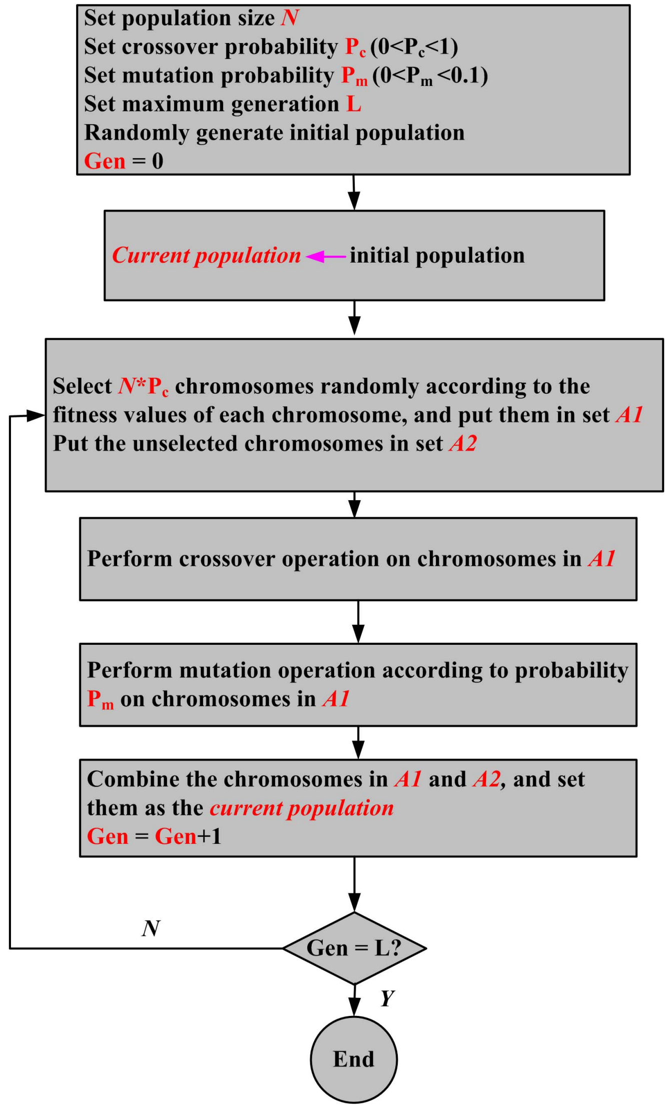


Fig. 5. Flowchart of GA based search in task sorting.

optimal solution[10].

3) Solution description: Based on above analysis, the problem of task sorting is proposed to be solved using a switched method, as shown in Fig. 4. When total task number $M < M0$ ($M0$ is a threshold), the best sequence will be searched in the whole search space with dimension of $M!$; while when $M >= M0$, the search method switches to GA based method to reduce the time cost.

1) Whole space search: this search method is straightforward.

2) GA based search: the flowchart of GA based search is shown in Fig. 5. Each chromosome is a sequence of the task IDs (each task ID is a gene). The initial population is set as N, the crossover probability is set as Pc, and the mutation probability is set as Pm.

Crossover operation: traditional crossover operation is performed in a pair of chromosomes, for example, chromosome 1 and chromosome 2. But this may bring frequent trouble in which the constraint that a task ID can only appear and must appear once in one space sequence is not satisfied. Thus here the Enhanced Edge Recombination operation[11] is adopted.

Suppose the fitness of chromosome s in generation i is $Fn(s,i)$ (can be calculated via equation (3)), the crossover probability of chromosome s is calculated as:

$$pr_c(s,i) = Fn(s,i) / \sum_{j=1}^{N} Fn(j,i), \tag{4}$$

where N is the size of the population. Whether chromosome s is selected to perform crossover operation or not is determined by Roulette Wheel Selection according to Pc and $pr_c(s,i)$.

Mutation operation: in the selected chromosomes, select one gene to perform mutation operation according to probability Pm. The selected gene mutates in $\{1, 2, \ldots, M\}$, M is the total task number. Suppose the value of the selected gene is u, and it mutates to v, thus the gene whose value is v must change to u to satisfy the constraint that a task ID can only appear and must appear once in one space sequence.

End criterion: the maximum generation L is used as the end criterion of the search method. L is set to be sufficient large to guarantee the performance of the final solution. The best solution in generation L is the optimal sorting solution.

D. Whole optimization

In the stage of Whole optimization, SA approach is adopted to find the optimal cell layout involving all tasks. After the first two stages, a near-optimal cell layout has been obtained, which reduces the search space in this stage. SA is selected here because it can provide better and better solutions which provides flexibility to a user. That is, a user can stop the optimization procedure if he or she is satisfied with the found solution up to now. Firstly, the principle of SA is given.

1) Principle of Simulated Annealing Algorithm: The essence of SA is based on the manner that metals recrystalize in the process of annealing. In real annealing process, a metal, initially at high temperature and disordered is slowly cooled so that the system is approximately in thermodynamic equilibrium at any time. As cooling proceeds, the system becomes more ordered and approaches a "frozen" ground state. By analogy, SA to combinatorial optimization problems is straight forward and has been used in various combinatorial optimization problems. The current state of the thermodynamic system is analogous to the current solution of the combinatorial problem, the energy of the thermodynamic system is analogous to the objective function, and frozen state is analogous to the global minimum[12][13][14].

2) Solution Description: The initial solution *S0* of SA is the layout generated through stage 1 and stage 2. Each solution is composed of the positions of each task. Fig. 6 shows how SA works. SA is briefly described as follows:

1) Set initial temperature *T0*, decent parameter h, $0 < h < 1$, control parameter k, neighbor size L. T is a global time-varying parameter to control the search steps. At the beginning, it is set high, meaning that the new solution may be

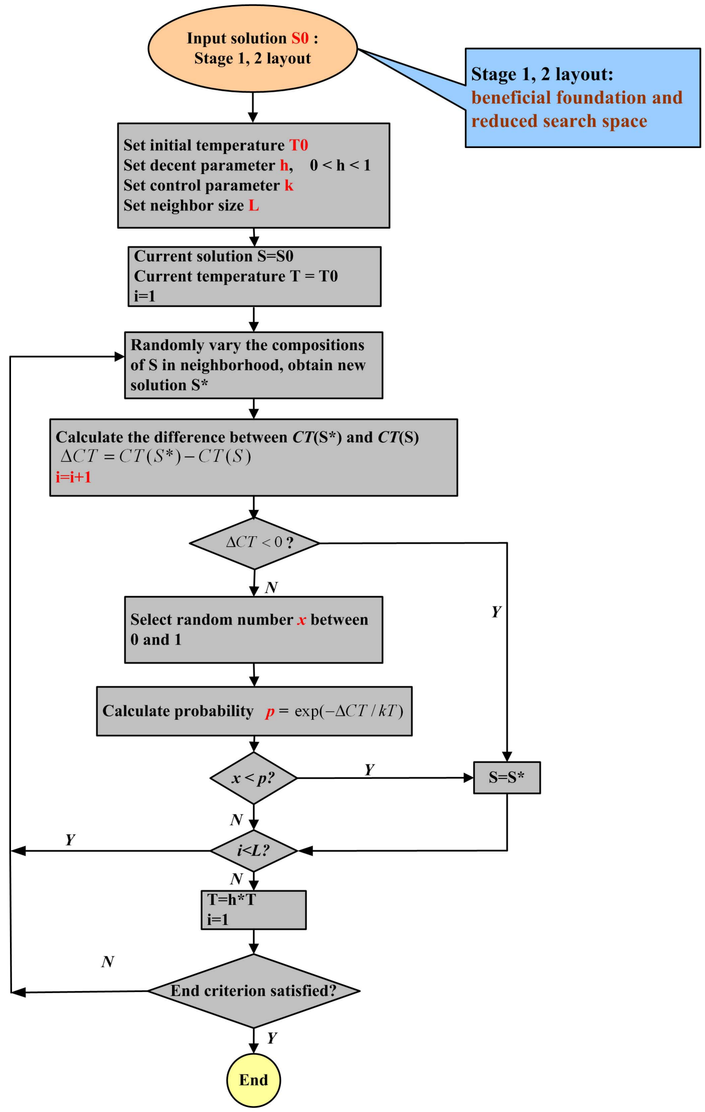


Fig. 6. Flowchart of Simulated Annealing based search in Whole optimization.

accepted even when it is worse than the current one according to the computed probability p. This feature prevents the method from getting stuck into a local minimum. L means to search L neighbors of the current solution. L can be determined according to the actual situation. k is a parameter to control the relative weight of ΔCT and T.

2) If L neighbors have been searched for current solution S, go to 7); else randomly vary the compositions of S in neighborhood, obtain new solution S^*.

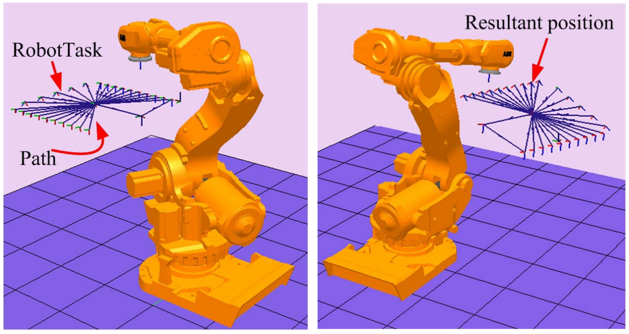
3) Calculate the difference between $CT(S^*)$ and $CT(S)$. $CT(S)$ is the cycle time of solution S.

4) If $\Delta CT < 0$, accept the current solution S^*, go to 2); else go to 5).

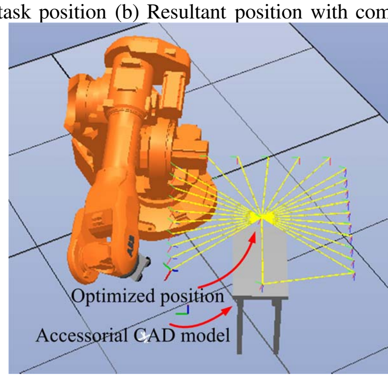
5) Select random number x between 0 and 1.

6) Calculate probability $p = \exp(-\Delta CT/kT)$. If $x < p$, accept the current solution S^*, go to 2); else go to 7).

7) Set $T = h * T$. If the end criterion is satisfied, end the



(a) Original task position (b) Resultant position with compared method



(c) Optimized position with given method

Fig. 7. Spot welding test case 1.

algorithm; else go to 2).

It is to point out that SA can start with any initial layout. This provides flexibility to a user: an initial solution can be specified by the user as the starting point to trigger the Whole optimization procedure; also the optimization process can be stopped during algorithm execution with an intermediate result, and resumed with the intermediate result as the starting point.

III. Experimental Results and Analysis

To evaluate the effectiveness of the proposed method, several application cases are given, including two spot welding application cases, one polishing case and one machine tending case. All the experiments are performed on ABB RobotStudio simulation platform, which can simulate different types of ABB industrial robots.

In spot welding cases, the robot is required to visit each spot point to perform welding operation, shown in Fig. 7. In spot welding case 1, the spot points are in alignment; while in spot welding case 2, the spot points are in specifically required positions, shown in Fig. 8. The testing results are shown in TABLE 1. Also the testing results are compared with the results from [6], in which the polynomial fitting method was employed to find the optimal task position. From TABLE 1 we can see, for either spot welding case, the given method shows significant improvement on the compared method. Reorientation of the tasks plays main role on cycle time reducing.

In polishing case, the robot task is quite simple with a good

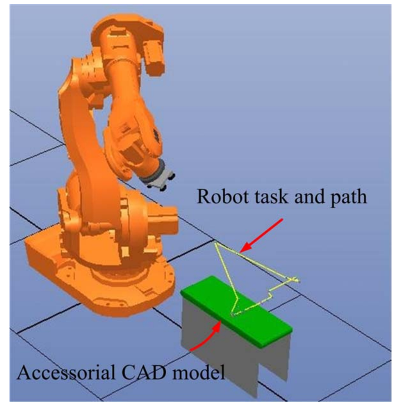


Fig. 8. Spot welding test case 1.

TABLE I
Test Cases and Results

Test cases	Compared solution	Given method	Improvement
Spot welding 1	34.86 s	28.9 s	17.1%
Spot welding 2	5.56 s	4.44 s	20.1%
Polishing	30.7 s	30.4 s	1.0%
Machine tending	21.9 s	19.1 s	12.8%

position already. Thus there is not too much space to improve the cycle time.

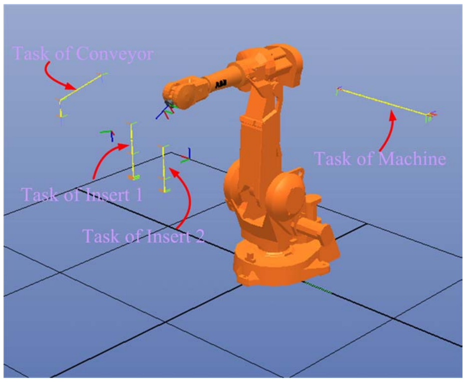
In machine tending test case extracted from real production, a machine for workpiece processing, two insert stations and one conveyor are included. The cycle is composed as: $Insert1 \rightarrow Machine \rightarrow Conveyor \rightarrow Insert2 \rightarrow Machine \rightarrow Conveyor$. Fig. 9 (a) shows the real cell layout before optimized, and Fig. 9 (b) shows the cell layout after optimization via the given method. The result shows a big cycle time improvement. The reason comes from the new space sorting of the tasks (stations), and also the relative positions adjustment of all tasks in optimization procedure.

As found in the experiments, for tasks with complex paths that usually include more targets with different orientations, the given method can exhibit significant benefits, but not for tasks with simple paths involving small amount of targets with similar orientations. The reason is that for simple paths, the robot performance doesn't vary much when the task is put at different positions. Also found in experiments, for the individual task, reorientation will always put more influence on the cycle time than translation. When multiple tasks are put into consideration, task sorting and then adjusting simultaneously are also effective ways for reducing cycle time.

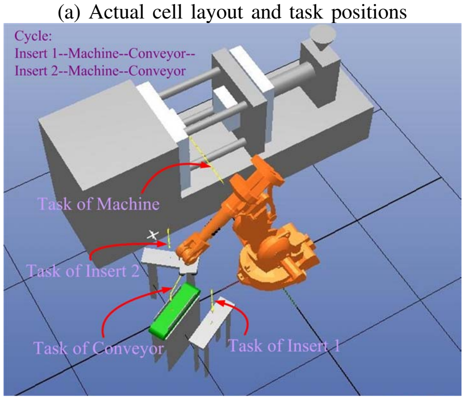
IV. Conclusion and Future Work

A. Conclusion

A method for optimizing the work cell layout which may involve one or more tasks has been given. The method can realize optimal work cell layout in terms of minimum cycle time with three major steps: firstly, the best position for each individual task is determined with use of preferred region;



(a) Actual cell layout and task positions



(b) Optimized cell layout and task positions

Fig. 9. Machine tending test case.

secondly, based on the results from the first stage, the optimal space sequence of the tasks is determined by means of a fast Genetic Algorithm; finally, the positions of the tasks are adjusted simultaneously in a greatly reduced search space using Simulated Annealing Method. The method was verified through several test cases and can be used as a reference for cell layout and improving robot performance for a new robotic cell setup or existing cell adjustment.

B. Future work

Although a method for optimizing the work cell layout has been implemented, there is still a gap when put into real work cell layout application. The gap mainly comes from the following aspects:

1) The real work cell is more complex. For example, a real work cell may involve obstacles with all kinds of types, the safety requirements, and other specific requirements. All these will definitely restrict the adjustment space and freedom when adjusting the task positions to get an optimal layout. To extract the information from real work cell, recover them and then combine them with the current approach is trivial work. Thus an effective tool to capture the real work cell scenario will be very helpful when the provided approach is put into real applications. The tool can be realized by existed sensing techniques, i. e., camera to SEE the real scenario, laser to TOUGH the real scenario, etc.

2) The proposed method mainly concerns with the robot task positioning, that is, the relative positions of the targets belongs to the same task are not adjusted. In fact, these positions, task position, and also tool data will put a comprehensive effect on cycle time. Of course, changing one aspect of these three aspects will influence the other aspects. Thus an integrated method taking into account these aspects together will further improve the work cell layout for actual production applications.

ACKNOWLEDGMENT

The authors would like to thank Dr. Steve Murphy for his constructive suggestions, and Mr. Haipeng Wang, Mr. Xingguo Yin and Mr. Li Tao for their technical support and daily help.

REFERENCES

[1] DASSAULT SYS OF AMERICA, "OPTIMIZATION TOOL FOR ROBOT PLACEMENT", Patent Number: US6526373, 2003.
[2] DASSAULT SYS OF AMERICA, "AN OPTIMIZATION TOOL FOR ASSEMBLY WORKCELL LAYOUT", Patent Number: EP1107082, 2001.
[3] FANUC LTD, "METHOD OF DECIDING ROBOT LAYOUT", Patent Number: US4979128, 1990.
[4] FANUC LTD, "ROBOT OFF-LINE SIMULATION APPARATUS", Patent Number: US2005004709, 2005.
[5] F. Masmoudi et. al, "OPTIMIZATION OF PRODUCT TRANSFER WITH CONSTRAINT IN ROBOTIC CELL USING SIMULATION", *Int. j. simul. Model.*, 2006, page 89-100.
[6] ABB RESEARCH LTD, "METHOD FOR OPTIMISING THE PERFORMANCE OF A ROBOT", Patent Number: US 2007106421, 2007.
[7] Lee, Z., Su, S., & Lee, C., "Efficiently Solving General Weapon-Target Assignment Problem by Genetic Algorithm with Greedy Eugenics", *IEEE Transactions on Systems, Man and Cybernetics*, Part B, 2003, 33, 113-121.
[8] Shima, T., Rasmussen, S. J., & Sparks, A. G., "UAV Cooperative Multiple Task Assignments Using Genetic Algorithms", in *American Control Conference*, 2005, Portland, USA, page 2989-2994.
[9] Juell, P., Perera, A. S., & Nygard, K. E., "Genetic Algorithm to Improve a Solution for the General Assignment Problem", in *16th International Conference on Computer Applications in Industry and Engineering*, Las Vegas, Nevada USA, 2003.
[10] Zhang, D., Wang, L., & Yu. J., "Geometric topology based cooperation for multiple robots in adversarial environments", *Journal of Control Engineering Practice*, 2008, (in publication)
[11] Starkweather, T. et al., "A comparison of genetic sequencing operations", in *Proceedings of the Third International Conference on Genetic Algorithms*, 1991, page 69-76.
[12] "Simulated Annealing", *available at: http://www.cs.sandia.gov/opt/survey/sa.html*.
[13] Cerny, V., "Thermodynamical Approach to the Traveling Salesman Problem: An Efficient Simulation Algorithm", *J. Opt. Theory Appl.*, 45, 1985, page 41-51.
[14] Kirkpatrick, S. et. al, "Optimization by Simulated Annealing", *Science*, 220, 1983, page 671-680.