Computers & Industrial Engineering 64 (2013) 537–544

Contents lists available at SciVerse ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie

Multiobjective layout optimization of robotic cellular manufacturing systems

Kazuhiro Izui [a,*], Yutaka Murakumo [a], Issei Suemitsu [a], Shinji Nishiwaki [a], Akio Noda [b], Tatsuya Nagatani [b]

[a] Department of Mechanical Engineering and Science, Kyoto University, Yoshida-honmachi, Sakyo-ku, Kyoto 606-8501, Japan
[b] Mitsubishi Electric Corporation, Tsukaguchi-honmachi 8-1-1, Amagasaki, Hyogo, Japan

ARTICLE INFO

Article history:
Received 9 June 2012
Received in revised form 4 December 2012
Accepted 5 December 2012
Available online 14 December 2012

Keywords:
Robot cell
Assembly layout
Design optimization
Genetic algorithm
Sequence-pair representation

ABSTRACT

This paper proposes a multiobjective layout optimization method for the conceptual design of robot cellular manufacturing systems. Robot cellular manufacturing systems utilize one or more flexible robots which can carry out a large number of operations, and can conduct flexible assemble processes. The layout design stage of such manufacturing systems is especially important since fundamental performances of the manufacturing system under consideration are determined at this stage. In this paper, the design criteria for robot cellular manufacturing system layout designs are clarified, and objective functions are formulated. Next, layout design candidates are represented using a sequence-pair scheme to avoid interference between assembly system components, and the use of dummy components is proposed to represent layout areas where components are sparse. A multiobjective genetic algorithm is then used to obtain Pareto optimal solutions for the layout optimization problems. Finally, several numerical examples are provided to illustrate the effectiveness and usefulness of the proposed method.

1. Introduction

Robotic cellular manufacturing systems (RCMSs) are a new type of manufacturing system in which one or more flexible robots carry out a large number of assembly operations that would be performed by human workers in conventional cellular manufacturing systems. When compared with conventional human cellular manufacturing systems, RCMS are seen to offer similar advantages, such as reduction of material flow distances and local inventory. However, although reduced operation costs due to automation of the manufacturing systems can be achieved by introducing RCMS, the design of assembly operations and robot teaching can become quite awkward and time-consuming when launching new manufacturing systems.

As with all manufacturing systems, the layout design stage, where fundamental performances of the manufacturing system under consideration are determined, is one of the most important stages when building efficient RCMS. The layout design stage is an upstream stage of the manufacturing system design process, and decisions made then exert considerable influence on the detailed design of robot motion planning and the teaching process. Therefore, skillful decision-making during the layout design stage is essential for minimizing design changes during the detailed design stage and teaching process, and to enhance the efficiency and reliability of the manufacturing systems when deployed.

With this background in mind, optimization techniques have been applied to assist the layout design of RCMS (Tay & Ngoi, 1996; Barral, Perrin, Dombre, & Liegeois, 2001). Unfortunately, these methods require explicit constraint handling regarding component overlapping, since component coordinates are handled as design variables, and this implementation obstructs global searching of the solution space.

Layout problems have raised important issues in many research fields, such as printed circuit board problems (Shahookar & Mazumder, 1991) and facility layout problems (Drira, Pierreval, & Gabouj, 2007; Gen, Lin, & Zhang, 2009), and effective optimization methods which can avoid the handling of overlapping constraints have been reported. For example, Kleinhans, Sigl, Johannes, and Antreich (1991), Quinn and Breuer (1979), and Dunlop and Kernighan (1985) proposed multi-step optimization techniques. In these methods, several component overlaps are allowed in the first optimization step, and such overlaps are subsequently resolved. However, the overlap resolution step may degrade solutions and optimization computational requirements tend to be significant. Tree-structure representation techniques (Dai & Kuh, 1987; Chen & Chang, 2006) are very effective for avoiding overlaps, but these methods can be applied only to partitioning problems. Birgin and Lobato (2010) proposed a rectangular packing technique which avoids overlapping, however this technique assumes that all rectangles have an identical shape.

On the other hand, sequence-pair representation can avoid overlaps among components and allow various sizes of rectangles (Murata, Fujiyoshi, Nakatake, & Kajitani, 1996). Several papers

* Corresponding author. Tel.: +81 75 753 5198, fax: +81 75 753 5857.
E-mail address: izui@prec.kyoto-u.ac.jp (K. Izui).

0360-8352/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.cie.2012.12.003

report that this representation enables very effective layout optimization in packing-type problems (Drakidis, Mack, & Massara, 2006) and facility layout problems (Meller, Chen, & Sherali, 2007; Liu & Meller, 2007). Therefore, this paper proposes a new layout optimization method for RCMS that uses sequence-pair representation. The major difference between RCMS layout problems and conventional packing-type or general facility layout problems is that the minimization of packing area is the most important criterion in the former, but optimal spacing among distributed components is paramount in the latter. This paper proposes a dummy component approach to provide such spacing between a single robot and other components. In the following section, the design requirements for the RCMS are clarified first and then quantitative design criteria are formulated. Next, a sequence-pair representation scheme for layout optimization is introduced and an optimization procedure is proposed. Finally, the proposed optimization method is applied to numerical examples to demonstrate the effectiveness of the proposed method.

2. Criteria for layout design

2.1. Layout design problem

The most important consideration when developing new manufacturing systems is operation time, and Tay and Ngoi (1996) and Barral et al. (2001) proposed single-objective layout optimization algorithms for minimizing RCMS operation time. However, during the layout design stage, multiple design requirements must be considered, so single-objective optimization approaches are insufficient in many RCMS design cases. Feasibility verification of assembly tasks is an essential process in the RCMS layout design stage. Another highly important consideration is the minimization of layout area, to enhance the efficiency of factory-level layouts. The quantitative criteria for these requirements are discussed in the following subsections.

2.2. Operation time

Consider an RCMS composed of part feeders, an assembly table and an assembly robot. The total operation time of the RCMS can be classified into assembly time and robot motion time. Robot motion time is time spent moving the robot end-effector to retrieve assemblies or assembly parts and move them to another location, and the manufacturing system layout greatly influences these durations, which must be minimized.

However, precise evaluation of robot motion time is almost impossible in the layout design stage where no detailed motion trajectory information is given. Therefore, we evaluate robot motion time based on the rotational angle of robot arm joints between starting and terminal points for each robot motion. Since the maximum angular velocity value of each assembly robot joint is usually given at the layout design stage, we approximately calculate the robot motion time by assuming that the joint motion has a uniform velocity based on its maximum angular velocity. Hence, the robot motion time T_k^i of the kth joint performing the ith motion is given in the following equation:

$$T_k^i = \frac{|\bar{q}_k^i|}{\omega_k} \tag{1}$$

where ω_k indicates the maximum angular velocity of the kth joint and $\bar{q}_k^i$ is the angle through which the kth joint moves during the ith motion.

We assume that all joints can simultaneously have their angle changed, and the ith motion time is evaluated using the maximum joint motion time as follows:

$$T_i = \max_k \left[\frac{|\bar{q}_k^i|}{\omega_k} \right] \tag{2}$$

Therefore, the total robot motion time for one assembly cycle T is evaluated as follows:

$$T = \sum_{i=1}^{h} T_i \tag{3}$$

where h is the number of robot motions in one assembly cycle.

2.3. Feasibility of assembly tasks

In RCMS, robots must conduct skilled assembly tasks. This means that having a robot arm reach appropriate task positions is only one of the necessary conditions required to conduct an assembly task, but not a sufficient condition alone. If the robot posture at a task position is close to a singular point, the assembly task may not be successfully completed. Therefore, consideration of how to avoid such singular points at the layout design stage is essential for obtaining a feasible assembly process.

In order to avoid singular points, evaluation of kinematic manipulability (Yoshikawa, 1985) is beneficial. Manipulability is a quantitative measure of the manipulating ability of robot arms that can be calculated based on the relationship among the position of the end-effector and joint vectors.

Consider a manufacturing system using an n degree-of-freedom assembly robot which conducts tasks at h positions, with its kth joint variable at the jth task position denoted as q_k^j, with joint vector $\mathbf{q}^j$. The robot posture vector $\mathbf{r}^j$ at the jth position ($j = 1, 2, \ldots, h$)is defined as follows:

$$\mathbf{q}^j = [q_1^j, q_2^j, \ldots, q_n^j]^T \quad (j = 1, 2, \ldots, h) \tag{4}$$

$$\mathbf{r}^j = [r_1^j, r_2^j, \ldots, r_m^j]^T \tag{5}$$

where m is the number of dimensions needed to represent the position and angle of the end-effector. Then, the relationship between $\mathbf{r}^j$ and $\mathbf{q}^j$ is represented as follows:

$$\mathbf{r}^j = f(\mathbf{q}^j) \tag{6}$$

In the following discussion, the notation of task position j is ignored for simplicity. By the chain-rule, velocity vector $\mathbf{v}(=\dot{\mathbf{r}})$ is obtained as,

$$\mathbf{v} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}} \tag{7}$$

where $\dot{\mathbf{q}}$ indicates the joint velocity vector and $\mathbf{J}(\mathbf{q})$ is the Jacobian.

Consider subset S_v of the realizable velocity $\mathbf{v}$ such that the corresponding joint velocity $\dot{\mathbf{q}}$ satisfies

$$\dot{\mathbf{q}} = \left(\dot{q}_1^2 + \dot{q}_2^2 + \cdots + \dot{q}_n^2\right)^{1/2} \leqslant 1 \tag{8}$$

This subset is an m dimensional Euclidean ellipse and its volume V is

$$V = c_m w \tag{9}$$

where the coefficient c_m is

$$c_m = \begin{cases} \frac{(2\pi)^{\frac{m}{2}}}{2\cdot 4\cdot 6\cdots(m-2)\cdot m} & \text{if } m \text{ is an even} \\ \frac{2(2\pi)^{\frac{m-1}{2}}}{1\cdot 3\cdot 5\cdots(m-2)\cdot m} & \text{if } m \text{ is an odd} \end{cases} \tag{10}$$

Since c_m is a constant value, the volume of the ellipse can be evaluated by w which is referred to as a manipulability measure (Yoshikawa, 1985). The larger the manipulability w, the faster the end-effector can move in any direction, implying that larger manipulability values of w indicate that the robot can conduct more complicated tasks. Therefore, we calculate the value of w here to

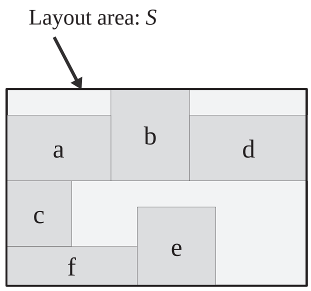


Fig. 1. Layout area.

evaluate the feasibility of the assembly tasks. In the case of non-redundant robots, w is calculated as follows:

$$w = |\det \mathbf{J}(\mathbf{q})| \tag{11}$$

Therefore, using the superscript of task position j, the overall feasibility of assembly task W in a robotic cellular manufacturing system is evaluated using the weighted summation of manipulability values at h task positions, as follows:

$$W = \sum_{j=1}^{h} a^j w^j \tag{12}$$

where w^j is the manipulability at the jth task position and a^j denotes a weighting coefficient.

2.4. Layout area

Layout area is an important design criterion when designing new assembly lines (Sherali, Fraticelli, & Meller, 2003). In this paper, every manufacturing system component is assumed to be rectangular, and layout area S is evaluated using the area of the smallest rectangle which can contain all allocated components, as shown in Fig. 1.

2.5. Optimization problem

Using the above three design criteria for the RCMS layout, a multiobjective optimization problem for its design can be formulated as follows:

$$\begin{cases} \text{minimize} & T,\ S \\ \text{maximize} & W \end{cases} \tag{13}$$

If one or more task positions like beyond the working envelope of a robot, a very large penalty value is assigned to each objective function.

Another constraint condition for this optimization problem is to avoid overlapping among components and to handle this constraint, and a sequence-pair representation scheme is introduced in the following section.

3. Layout representation scheme

3.1. Sequence-pair

RCMS layout problems can be regarded as allocation optimization problems for manufacturing system components of various sizes. One intuitive approach for solving such optimization problems is to handle the coordinates of each component as design variables and use local optimization techniques, however, such approaches involve complicated handling of constraints regarding overlapping among the components. The sequence-pair representation approach can avoid a constraint resolution process for overlapping constraints, since the native representation inherently prohibits overlaps. This paper employs this scheme to represent the layout of RSMS components.

The sequence-pair approach employs a pair of sequences, Γ_+ and Γ_-, where each sequence indicates the relative positions of manufacturing system components. Furthermore, in this paper, an additional variable, θ_l ($l = 1, 2, \ldots, L$), is introduced to represent the orientation of each allocated rectangular component, where l denotes the component number and L is the total number of components. Since we assume that all components are rectangular, their orientation, *i.e.*, long side vertical or long side horizontal, is represented by $\theta_l \in \{0, 1\}$.

3.2. Decoding procedure

Layouts represented by a sequence-pair can be decoded using an oblique grid. For example, in the case of $\Gamma_+ = \{cdebfa\}$, $\Gamma_- = \{dfecab\}$, these two permutations, Γ_+ and Γ_-, represent relative positions of components on the oblique grid as shown in Fig. 2, where the sequences of the two grid line labels are the same as the permutations, Γ_+ and Γ_-. That is, the label sequence from upper-left to lower-right is the same as Γ_+ and the label sequence from lower-left to upper-right is the same as Γ_-. Furthermore, components are located at the intersection of the two corresponding component grid lines.

Next, two directed and vertex-weighted graphs, horizontal and vertical constraint graphs, are constructed as shown in Fig. 3. In the horizontal constraint graphs, two components are connected when one component is to the left of another component in the both sequences. Components in the vertical constraint are connected when a component letter label is to the left of the other component letter label in Γ_+ and, correspondingly, to the right in Γ_-. The position of each component is then calculated by solving the longest path problem, considering the orientation of the components indicated by θ_l. Consequently, a layout is obtained as shown in Fig. 4. See the reference (Murata et al., 1996) for details of the procedure.

3.3. Representation of distance between components

Since the sequence of letter labels in the sequence-pair approach represents the relative position of components, the space between components is not considered. Thus, sequence-pair representation has been applied to packing problems such as VLSI design, where the packing area of components is to be minimized (Drakidis et al., 2006). However, RCMS layout design problems must consider the inclusion of adequate space between components, since robot manipulation requires certain minimal distance from task points to allow complex assembly tasks to be conducted at these points.

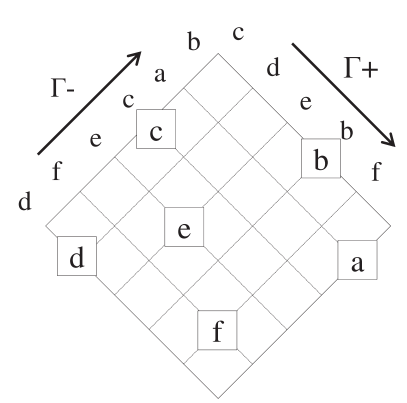
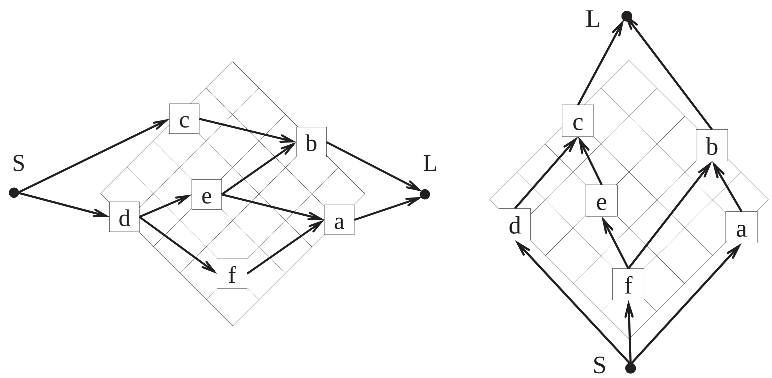


Fig. 2. Relative arrangement.



(a) Horizontal constraint graph (b) Vertical constraint graph

Fig. 3. Constraint graph.

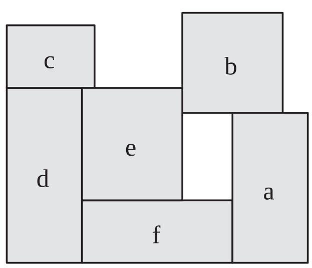


Fig. 4. Component layout.

This paper proposes the use of dummy components to represent such spaces in RCMS layout problems. Several rectangular dummy components are introduced in an optimal layout design problem, and layout of both ordinary and dummy components is simultaneously optimized. Consequently, dummy components are allocated at places where space is required, allowing layout optimization considering the spacing requirements of employed robots.

4. Optimization method

4.1. Genetic algorithm

The proposed layout optimization problem has permutation variables which represent the relative position of components, and binary variables which represent the orientation of rectangular components. Branch-and-bound algorithms (Solimanpur & Jafari, 2008; Xie & Sahinidis, 2008) may provide a global optimum solution, but the problem size must be small due to combinatorial explosion. The use of meta-heuristic approaches such as genetic algorithms (GAs) (Liu & Meller, 2007; Solimanpur & Kamran, 2010), particle swarm optimization (Samarghandi, Taabayan, & Jahantigh, 2010), simulated annealing (Sahin, Ertogral, & Turkbey, 2010) and an ant system (Komarudin & Wong, 2010) are effective for solving large-scale combinatorial optimization problems, and a further advantage is that GAs can provide non-dominated solutions for multiobjective optimization problems. Therefore, this paper uses a GA to solve the RCMS layout problems and the following sections describe its crossover, mutation and selection operators.

4.2. Crossover

This paper utilizes Placement-based Partially Exchanging Crossover (PPEX) (Nakaya, Wakabahashi, & Koide, 2000). In PPEX, a pair of individuals is first chosen, one component in one individual of the pair is then selected at random, and a window domain is created around this component. The sequence and orientation of components inside this window is exchanged with those in the other individual.

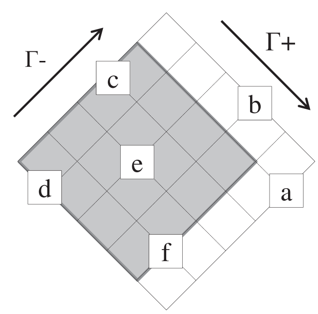


Fig. 5. PPEX window.

An example of the PPEX procedure is described in Figs. 5 and 6. Fig. 5 shows the window domain around component "e", indicated by the gray area. The sequence and orientation of components "c", "d", "e" and "f" are inherited from Parent 2 and the sequence and orientation of components "a" and "b" are inherited from Parent 1.

4.3. Mutation

A mutation operator is applied to a component at a preassigned mutation probability. When the mutation operator acts on a component, another component in the same individual is chosen at random. The sequence of these two components in permutations Γ_+ and Γ_- are then exchanged and the orientation of θ_l of these two components is changed.

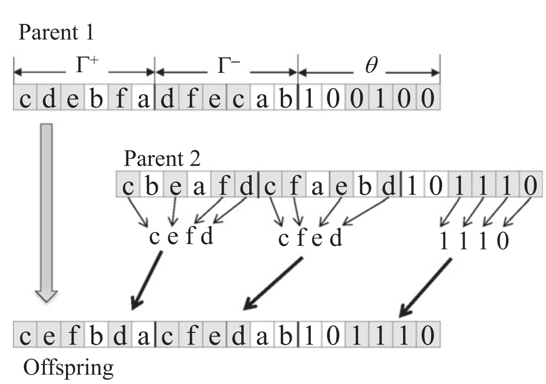


Fig. 6. PPEX manipulation.

4.4. Selection

The layout design optimization problem proposed in this paper is a multiobjective optimization problem regarding robot motion time, manipulability and layout area. Typical ways to solve multiobjective optimization problems include the use of the weighted sum method or ϵ-constraint method, which converts a multiobjective optimization problem to a single objective optimization problem. However, adjustment of weighting coefficients for the weighted sum method, or determination of the constraint threshold for the ϵ-constraint method requires an awkward trial-and-error process. Therefore, in this paper, Non-dominated Sorting Genetic Algorithm (NSGA)-II (Deb, Pratap, Agarwal, & Meyarivan, 2002), a multiobjective optimization technique based on GA, is used. NSGA-II can provide a non-dominated solution set for a multiobjective optimization problem in a single computation. The obtained non-dominated solution set provides trade-off information among the objective functions.

In the NSGA-II selection scheme, non-dominated solutions are preserved and carried over into the next generation, and a crowdedness criterion is used to select non-dominated solutions existing in sparsely distributed areas, to obtain widely distributed non-dominated solutions.

5. Numerical examples

5.1. Problem settings

The proposed RCMS layout design method is applied to example problems using several parameter settings. The assembly robot used in the following problems is shown in Fig. 7 and Table 1 shows the sizes and number of operations required of the layout components. The number of operations means the number of times the robot goes to the corresponding part box and then returns to the assembly table. This means that the robot effector retrieves a part from a particular part box location and returns to assembly table, and repeats this operation for the listed number of operations during the assembly task.

The number of individuals, the terminal generation number, cross-over rate and mutation rate are set to 80, 10,000, 0.8 and $1/2L$, respectively, where L is the number of components including dummy components. The optimization calculation was conducted five times for each problem setting, and only the non-dominated solutions of the five optimization calculations are shown in the figures.

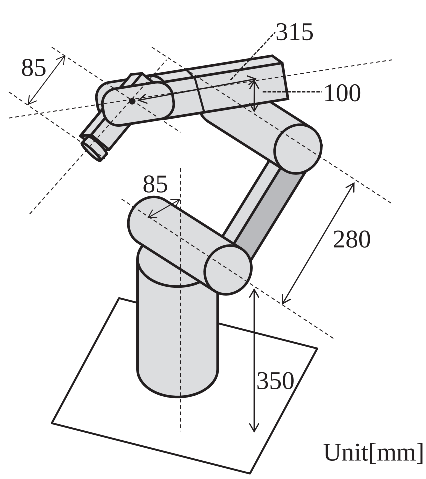


Fig. 7. Assembly robot configurations.

Table 1
Cellular manufacturing system components.

Machine ID (Machine name)	Size (height, width) (mm)	Number of operations
0 (Robot)	(255, 255)	–
1 (Assembly Table)	(270, 188)	10
2 (Part box 1)	(150, 100)	2
3 (Part box 2)	(150, 100)	2
4 (Part box 3)	(150, 100)	2
5 (Part box 4)	(150, 100)	1
6 (Part box 5)	(150, 100)	1
7 (Part box 6)	(150, 100)	1
8 (Part box 7)	(150, 100)	1

5.2. Effectiveness of dummy components

First, optimization problems considering only two objective functions are solved in order to compare the optimization performance for different problem settings, since a planer distribution of non-dominated solutions is easy for performance evaluation. The example problems in this and the following sections consider robot motion time T and manipulability W, while layout area S is ignored.

Fig. 8 shows the non-dominated solutions obtained using the proposed optimization method in two cases, one where no dummy components were used and the other where 13 dummy components were used. In this problem, all dummy components are square and 90 mm on a side. Twelve solutions were obtained in the case when dummy components were not used, while six solutions were obtained when dummy components were used.

The layout of solutions A–C in Fig. 8 are shown in Figs. 9–11. The numbers inside the rectangles in the figures indicate the component number shown in Table 1, and gray squares in Fig. 11 indicate the dummy components.

Fig. 8 clearly shows the trade-off relationship between robot motion time and manipulability, namely, larger manipulability requires longer robot motion time and shorter robot motion time can be achieved at the cost of less manipulability. Fig. 8 also shows that superior solutions regarding manipulability are obtained in the case when dummy components are used. As a result, we observe that superior optimization performance is possible when dummy components are included.

Solution A obtained in the no dummy component case is superior concerning robot motion time, but inferior in manipulability. This feature of solution A can be understood from Fig. 9, which shows the component layout. Part boxes, components 2–10, are located around the assembly table, component 1, so robot motion time between the components and the assembly table can be shortened. However, since components 4, 6 and 8 are too close to the robot, component 0, manipulability for these components is quite low.

Solution B is the best solution regarding manipulability in the no dummy component case, but worse in terms of robot motion time. The component layout for solution B is shown in Fig. 10. Several components are located far from the robot, component 0, which enhances manipulability at these locations. These figures show that component overlapping does not occur, and appropriate and understandable solutions are obtained using the proposed optimization method. However, the positions of components 7 and 8 in solution B are too close to the robot and ruin the manipulability for this solution. Thus, the use of dummy components is a practical necessity.

Fig. 11, the layout of solution C obtained when using dummy components, shows that the assembly table and part boxes are located at appropriate distances from the robot, due to the intercession of dummy components, so that manipulability is preserved for

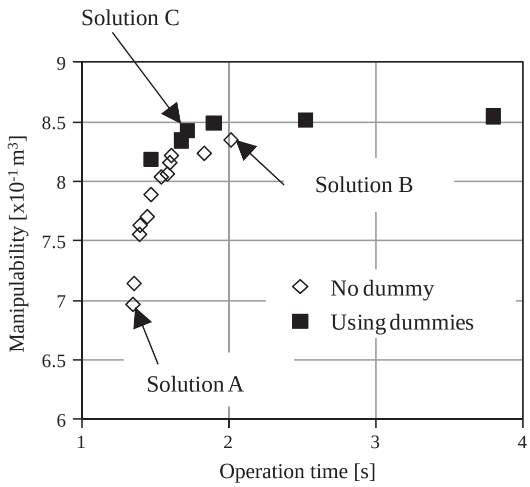


Fig. 8. Comparison of obtained non-dominated solutions.

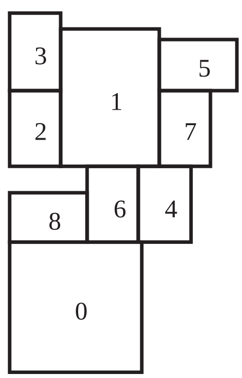


Fig. 9. Layout of solution A.

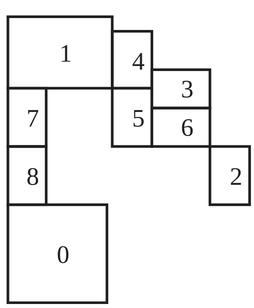


Fig. 10. Layout of solution B.

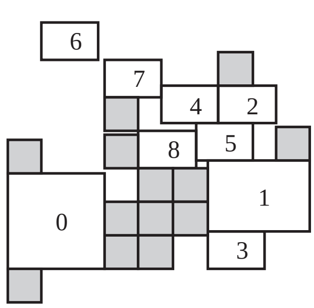


Fig. 11. Layout of solution C.

all parts boxes. This result further illustrates the necessity of using dummy components for sequence-pair based RCMS layout optimization.

5.3. Size of dummy components

Next, the effect of the size of dummy components on optimization results is discussed. Fig. 12 shows non-dominated solutions in three cases where the edge length a of 13 square dummy components is set to 30 mm, 90 mm and 150 mm, respectively.

From this figure, we observe that the 30 mm square dummy components fail to provide high manipulability compared with the 90 mm case, since 30 mm is too small. In the 150 mm square dummy component case, compromised solutions having shorter robot motion time and larger manipulability are inferior to the 90 mm case. These comparative results imply that an adequate size of dummy components is required to obtain better solutions, however, prediction of the most appropriate size prior to optimization calculation is a challenging issue.

Next, 13 rectangular dummy components having various edge lengths as shown in Table 2 are used. Non-dominated solutions for these cases are shown in Fig. 13 and compared with the 90 mm square dummy component case.

Fig. 13 shows that the use of rectangular dummy components can provide better non-dominated solutions than square dummy components. Fig. 14 shows the component layout of solution D in Fig. 13, where dummy components having appropriate edge lengths were selected. This result shows that the use of rectangular dummy components having various edge lengths can be effective when the most appropriate size is unknown.

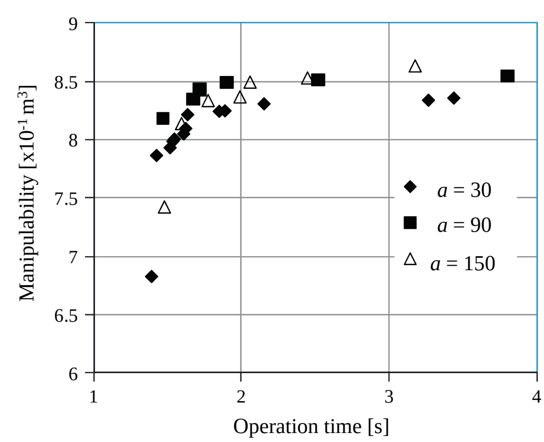


Fig. 12. Comparison of solutions for different dummy component sizes.

Table 2
Dummy component sizes.

Dummy ID	Height (mm)	Width (mm)
9	60	70
10	70	80
11	80	90
12	100	110
13	120	130
14	60	70
15	70	80
16	80	90
17	100	110
18	60	70
19	70	80
20	80	90
21	100	110

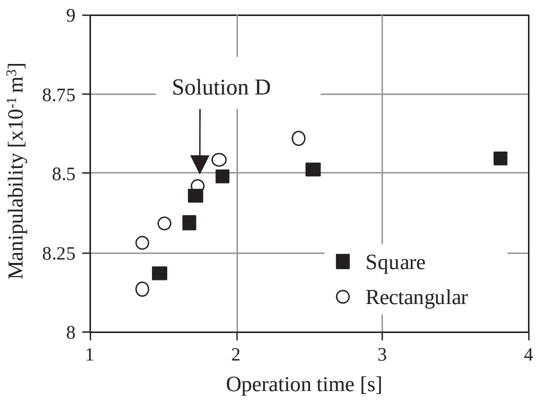


Fig. 13. Non-dominated solutions when using square and rectangular dummy components.

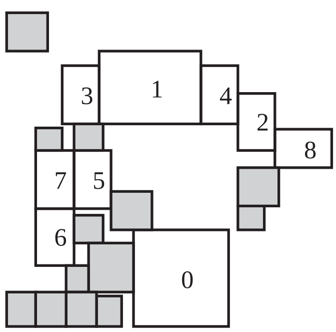


Fig. 14. Layout of solution D.

5.4. The number of dummy components

In this section, we discuss the effect of the number of dummy components. The optimization result shown in Section 5.3, where 13 rectangular dummy components N_d are included, illustrates that some of the 13 dummy components in this layout are unnecessary. That is, the same layout can be represented using fewer dummy components, and the inclusion of several irrelevant dummy components does not negatively affect the obtained optimal layout. Therefore, this layout representation scheme has a certain level of robustness against the parameter setting for the number of dummy components. We now compare this result with the results where N_d is set to 5 or 25.

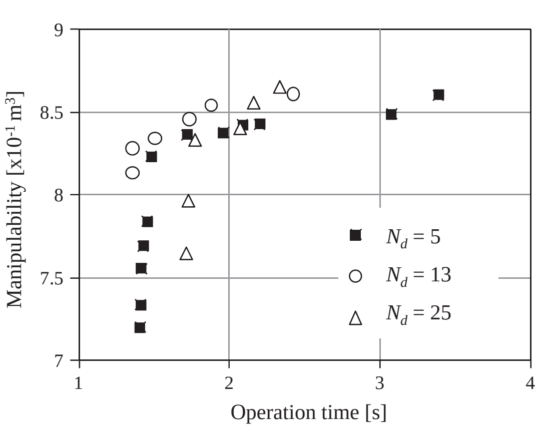


Fig. 15. Non-dominated solutions when different numbers of dummy components are used.

The rectangular dummy components, from No. 9 to No. 13, are used in the $N_d = 5$ case. In the $N_d = 25$ case, two dummy components are used for each component from No. 9 to No. 20, and a single additional No1. 21 dummy component is used. The obtained non-dominated solutions are shown in Fig. 15. This figure illustrates that solutions having higher manipulability but inferior operation times are obtained when N_d is increased. These comparative results clarify that the optimization problem become very complex when N_d has a large value, since the number of design variables increases. Conversely, when N_d is a small number, superior solutions regarding operation time are obtained, but they are inferior in terms of manipulability. These results indicate that the number of components, N_d, should be carefully determined, since this parameter affects the outcome of the optimization results. Therefore, how to determine an appropriate parameter for the number of dummy components prior to optimization is one of the key research issues we hope to address in future work.

5.5. Three-objective optimization problem

Finally, the proposed layout optimization method is applied to three objective optimization problems considering robot motion time, manipulability and layout area. This example uses the same settings as the two-objective optimization case discussed above

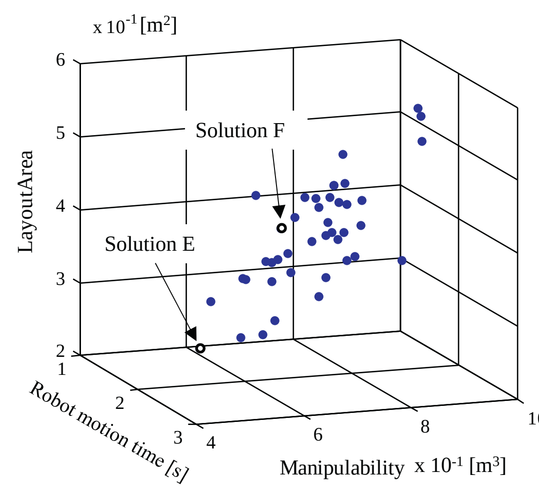


Fig. 16. Non-dominated solutions for three-objective problem.

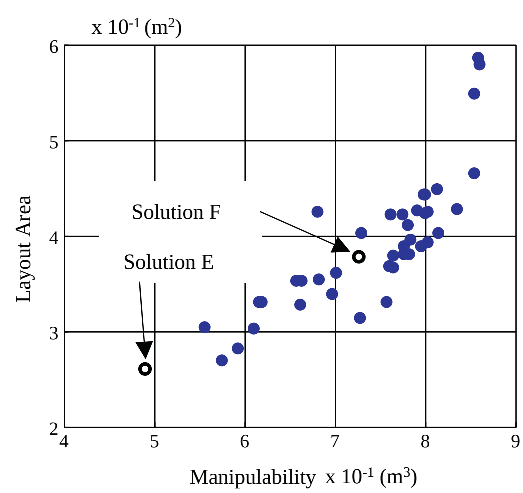


Fig. 17. Non-dominated solutions mapped to the layout area-manipulability plane.

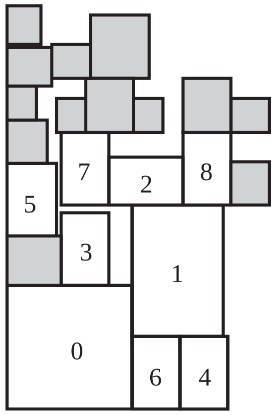


Fig. 18. Layout of solution E.

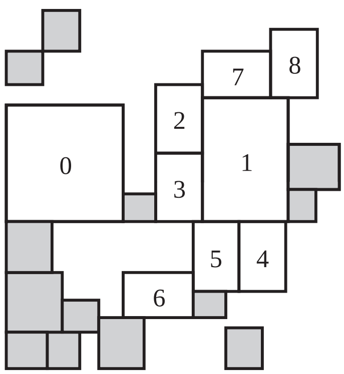


Fig. 19. Layout of solution F.

regarding the robot configuration, the number and size of rectangular dummy components and genetic algorithm parameters.

The distribution of the obtained non-dominated solutions is shown in Fig. 16 where trade-off relationships among the three objective functions can be observed. The trade-off relationship between layout area and manipulability is especially significant, as shown in Fig. 17.

The layouts of two solutions indicated by black circles are illustrated in Figs. 18 and 19. Fig. 18 shows the layout of solution E, which has the smallest layout area among the non-dominated solutions. Fig. 19 shows the layout of compromise solution F which has reasonable performance values for every criterion. These figures also imply the existence of a trade-off relationship between layout area and manipulability. That is to say, a smaller layout area reduces manipulability since the robot arm must undergo more radical movements during assembly operations. Thus, an adequate layout area is necessary to obtain sufficient manipulability.

6. Conclusions

This paper proposed a new layout optimization method for RCMS. First, the design requirements for RCMS layouts were clarified. Robot motion time, manipulability and layout area, were proposed as design criteria and an optimization problem was formulated. A sequence-pair scheme was introduced for the RCMS layout representation to avoid overlapping among assembly system components during the optimization procedure. Furthermore, the use of dummy components was proposed, to enhance obtaining solutions that have optimal spacing between components. A genetic algorithm was used to solve the multiobjective combinatorial optimization problems. A PPEX-type operator was used for crossover of sequence-pair chromosomes and NSGA-II was used to obtain non-dominated solutions. The proposed method was applied to numerical examples, and its effectiveness was demonstrated. Moreover, a simple guide for determining appropriate dummy component sizes was proposed, based on the results of several parameter studies.

References

Barral, D., Perrin, J. P., Dombre, E., & Liegeois, A. (2001). Simulated annealing combined with a constructive algorithm for optimising assembly workcell layout. *The International Journal of Advanced Manufacturing Technology, 17*(1), 593–602.

Birgin, E. G., & Lobato, R. D. (2010). Orthogonal packing of identical rectangles within isotropic convex regions. *Computers & Industrial Engineering, 59*(4), 595–602.

Chen, T.-C., & Chang, Y.-W. (2006). Modern fioorplanning based on B*-tree fast simulated annealing. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 25*(4), 637–650.

Dai, W. N., & Kuh, E. S. (1987). Simultaneous floor planning and global routing for hierarchical building-block layout. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 6*(5), 828–837.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transaction Evolutionary Computation, 6*(2), 182–197.

Drakidis, A., Mack, R. J., & Massara, R. E. (2006). Packing-based VLSI module placement using genetic algorithm with sequence-pair representation. *IEE Proceedings – Circuits, Devices & Systems, 156*(6), 545–551.

Drira, A., Pierreval, H., & Gabouj, S. H. (2007). Facility layout problems: A survey. *Annual Reviews in Control, 31*(2), 255–267.

Dunlop, A. E., & Kernighan, B. W. (1985). A procedure for placement of standard-cell VLSI circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 4*(1), 92–98.

Gen, M., Lin, L., & Zhang, H. (2009). Evolutionary techniques for optimization problems in integrated manufacturing system: State-of-the-art-survey. *Computers & Industrial Engineering, 56*(3), 779–808.

Kleinhans, J. M., Sigl, G., Johannes, F. M., & Antreich, K. J. (1991). GOR- DIAN: VLSI placement by quadratic programming and slicing optimization. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 10*(3), 356–365.

Komarudin & Wong, K. Y. (2010). Applying ant system for solving unequal area facility layout problems. *European Journal of Operational Research, 202*(3), 730–746.

Liu, Q., & Meller, R. D. (2007). A sequence-pair representation and MIP-model-based heuristic for the facility layout problem with rectangular departments. *IIE Transactions, 39*(4), 377–394.

Meller, R. D., Chen, W. P., & Sherali, H. D. (2007). Applying the sequence-pair representation optimal facility layout designs. *Operations Research Letters, 35*(5), 651–659.

Murata, H., Fujiyoshi, K., Nakatake, S., & Kajitani, Y. (1996). VLSI module placement based on rectangle-packing by the sequence-pair. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 15*(12), 1518–1524.

Nakaya, S., Wakabahashi, S., & Koide, T. (2000). An adaptive genetic algorithm with sequence-pair representation of solutions for VLSI floorplanning. In *The 2000 IEEE international symposium on circuits and systems* (Vol. 3). pp. 65–68.

Quinn, N. R., & Breuer, M. A. (1979). A forced directed component placement procedure for printed circuit boards. *IEEE Transactions on Circuits and Systems, CAS-26*(6), 377–388.

Sahin, R., Ertogral, K., & Turkbey, O. (2010). A simulated annealing heuristic for the dynamic layout problem with budget constraint. *Computers & Industrial Engineering, 59*(2), 308–313.

Samarghandi, H., Taabayan, P., & Jahantigh, F. F. (2010). A particle swarm optimization for the single row facility layout problem. *Computers & Industrial Engineering, 58*(4), 529–534.

Shahookar, K., & Mazumder, P. (1991). VLSI cell placement techniques. *ACM Computing Surveys, 23*(2), 143–220.

Sherali, H. D., Fraticelli, B. M. P., & Meller, R. D. (2003). Enhanced model formulations for optimal facility layout. *Operations Research, 51*(4), 629–644.

Solimanpur, M., & Jafari, A. (2008). Optimal solution for the two-dimensional facility layout problem using a branch-and-bound algorithm. *Computers & Industrial Engineering, 55*(3), 606–619.

Solimanpur, M., & Kamran, M. A. (2010). Solving facilities location problem in the presence of alternative processing routes using a genetic algorithm. *Computers & Industrial Engineering, 59*(4), 830–839.

Tay, M. L., & Ngoi, B. K. A. (1996). Optimising robot workcell layout. *The International Journal of Advanced Manufacturing Technology, 12*(5), 377–385.

Xie, W., & Sahinidis, V. (2008). A branch-and-bound algorithm for the continuous facility layout problem. *Computers & Industrial Engineering, 32*(4–5), 1016–1028.

Yoshikawa, T. (1985). Manipulability of robotic mechanisms. *International Journal of Robotics Research, 4*(2), 3–9.