A new Algorithm for Pick-and-Place Operations

V. Aminzadeh, H.A. Wurdemann, J.S. Dai, J. Reed, G. Purnell

Industrial Robot: An International Journal, 2010

http://dx.doi.org/10.1108/01439911011081678

1. Introduction

In the food industry robotics is used regularly in pick and placing operation with both serial and parallel robot especially for handling disarrayed product into structured state. The current algorithms used to pick-and-place from disarray to structure are usually variations of greedy algorithm (Kleinberg and Tardos, 2006), which tends to pick the product ahead of the others on the conveyor belt or chooses the one which is closest to the desired position and although there have been scattered effort for special handling cases, no systematic effort has been done to develop algorithm for handling from disarray to structure.

This paper approaches this problem and derives a solution based on assignment problem. Based on this, the paper for the first time proposes an optimal algorithm to be used for the pick-and-place operation from disarray to structure and simulates the results by comparing the algorithm with existing pick-and-place algorithms.

Although formulation and optimization of pick-and-place is not a new subject to the best of our knowledge little or none has been done in disarray to structure and majority of the research done has been in structure to structure pick-and-place where apart from the final positions, the initial location of the products are known too. Also in most of the cases the products are different and for each product there is a pre-assigned final position (common case in all the assembly processes and particularly investigated for printed circuit board (PCB) manufacturing).

In structure to structure modelling the pick-and-place optimization is being performed by optimizing two subproblems; feeder assignment and placement sequence. Feeder assignment problem can be modelled as a quadratic assignment problem and the placement sequence is solved as a Travelling salesman problem. Various algorithms have been developed to solve the problem with above decomposition (Kumar and Li, 1988; Ahamdi et al, 1998; Mainmon and Shtub, 1991; Grotzinger, 1992; Leipala and Nevalainen, 1989; Foulds and Hamacher 1993; Sohn and Park, 1996, Sadiq et al, 1993; Moyer and Gupta, 1996).

This is the basis of first efforts in mathematical modelling of structure to structure handling by Ahmadi et al, (Ahmadi et al, 1998) which have proposed a solution using mixed integer linear programming. A more advanced model with a dual head placement has been utilized in (Mainmon and Shtub, 1991; Grotzinger, 1992).

In contrast with above structure to structure modelling and formulation in food manufacturing pick-and-place there are no feeder assignment and the only problem considered is the position assignment. The algorithm developed in this paper minimizes the distance travelled by the robot by choosing the right assignment between the disarrayed products and the structured positions.

2. Assignment problem and relevant algorithms

Assume there are *n* products on the conveyor belt which are supposed to be placed at *n* positions.

The disarray and structure can be modelled as two sets; disarray set (DS) and structure set (SS). $DS = \{\boldsymbol{x} | \boldsymbol{x} \text{ is coordinate of a dissarayed product}\},\ |DS| = n$. Structure set is defined as $SS = \{\boldsymbol{x} | \boldsymbol{x} \text{ is coordinate of a Structured positions}\}, |SS| = n$.

In a 2D case disarray and structure sets are members of R^2 if orientation is not considered and $\mathrm{R}^2 \times [0,2\pi)$ otherwise. In such a case the handling process is a bijective mapping between the two sets. The total number of possible mappings (assuming the order does not matter) is *n!*.

The pick-and-place operation can be considered as an assignment problem in the mathematical domain. The formulation of the problem here has been done based on a 2D pick-and-place on a conveyor belt.

The optimization problem here is to find the assignment which is the most beneficial. In order to do so it is required to define an appropriate measure to judge between the assignments. The two possible parameters are handling total time and total distance. It is obvious that the two stated parameters are not independent of each other and a distance minimizing problem can be converted into a time minimizing problem if the handling system parameters are known.

In structure to structure formulation (Kumar and Li, 1988; Ahamdi et al, 1998; Mainmon and Shtub, 1991; Grotzinger, 1992; Leipala and Nevalainen, 1989; Foulds and Hamacher 1993; Sohn and Park, 1996, Sadiq et al, 1993; Moyer and Gupta, 1996; Carmon et al, 1989; Shih et al, 1996; Yeo et al, 1996; Su and Srihari 1996) usually the time is the parameter utilized to judge between the possible assignments. This is mainly because the structure to structure has been divided into two parts which are optimized independently and the parameter common between them is time. Also it is possible to associate all the costs of the handling (mainly translation, rotation and inspection) with a total time.

On the other hand in disarray to structure utilizing distance is more beneficial; it is independent of the robotic system, it is possible to optimize the structure parameters based on disarray parameters and it does not need the real-time conversion of distances to time for each assignment.

Utilizing the total distance travelled by the robot as the optimization parameter the problem can be formulating as follows

$$c = \sum_{j=1}^{n} \sum_{i=1}^{n} q_{i,j} . d_{i,j} \tag{1}$$

$$q_{i,j} = \begin{cases} 1 & \text{if } i\text{th product is placed in } j\text{th position} \\ 0 & \text{otherwise} \end{cases}$$

$$\sum_{j=1}^{n} q_{i,j} = 1, \sum_{i=1}^{n} q_{i,j} = 1 \tag{2}$$

$$i = 0,1,\ldots,n \quad , \quad j = 0,1,\ldots,n$$

$$d_{i,j} = \sqrt{(x_i - x'_j)^2 + (y_i - y'_j)^2} \qquad (3)$$

$$d_{i,j} \geq 0$$

$$\mathbf{C} = \left[d_{i,j} \right]_{n \times n} \qquad (4)$$

Where $d_{i,j}$ is the distance between the *i*th product and the *j*th position, $\left(x_i, y_i\right)$ is the coordinate of the *i*th product, $\left(x'_j, y'_j\right)$ is the coordinate of the *j*th position, $q_{i,j}$ is the binary parameter which is 1 if the *i*th product has been placed in the *j*th position and zero otherwise, *c* is the total distance travelled by the robot and **C** is the cost function whose *i*th row and *j*th column element is $d_{i,j}$.

It is obvious that each product can be assigned to one position and similarly each position can be filled with one product. Eq. (2) represents these conditions.

It should be noted that the actual distance travelled by the robot is slightly different from the Eq. (1). In reality the robot picks a product, moves to the position and then returns to the next product position, however if the products are relatively close to each other (compare to the distance travelled by the robot for placing the product) it can be assumed that the returning distance is almost equal with the first distance and the total cost becomes twice as the Eq. (1), as a result optimizing Eq. (1) gives the optimum solution for that case as well. The inaccuracy in the model has been considered in the simulations.

The assignment problem is a mixed integer linear programming (MILP) problem and there are a number of possible solutions to the assignment problem including

Cutting plane

Branch and bound

Branch and cut

Hungarian algorithm

The first three algorithms are general MILP algorithms and although they derive a solution but due to the special structure of the assignment problem they take large number of iterations to converge.

A better solution for the assignment problem has been published by Harold Kuhn (Kuhn, 1955; Kuhn, 1956). This algorithm is also known as Hungarian algorithm. It was first investigated by G. Monge (Monge, 1781), and further developed by D. Konig (Kőnig, 1915), and J. Egervary (Egerváry, 1955). Later J. Munkres (Munkres, 1957) reviewed the algorithm; this time with existence of higher computation power it was possible to investigate the method for more complicated situations and demonstrate its potential. As a result the algorithm is sometimes called Munkres algorithm.

3. Cost Function Based Elementary Operation for Optimal Assignment

Consider the cost function of Eq. (1). The optimization problem is to find the set of $q_{i,j}$ s that minimize the cost subject to the constraints of Eqs. (2) and(3). In order to find the optimum assignment utilizing Hungarian algorithm each assignment should be represented with a matrix of zeros and ones. This matrix is called assignment matrix Eq. (7).

$$\mathbf{M}=[q_{i,j}]_{n\times n} \tag{5}$$

The rows of the assignment matrix represent the positions and the columns represent products. If the product P_i has been assigned to the position J_j then $q_{i,j}$ is *1* and otherwise *0*.

In equivalence of Eq. (2) each row and column of the assignment matrix has only one *1* and the rest of the elements are *0*.

The Hungarian algorithm starts with the cost matrix of Eq. (4) and derives the optimal assignment based on two fundamental theorems:

> Elementary matrix operations on cost function do not affect the optimum assignment
>
> If the cost matrix of an assignment problem has non-negative entries and at least *n* zeros, then the optimal solution to the problem exists if *n* zeros lie in the position of the ones of the $n \times n$ assignment matrix.

It should be noted that a cost function with *n* zeros does not necessarily generate an assignment matrix and the generated assignment matrix should satisfy the conditions of Eq. (2). Therefore the Hungarian algorithm divides the zeros into two groups of assigned and not assigned zeros. In each row and column there can only be one assigned zero. At the next step the algorithm manipulates the cost matrix to obtain assigned zeros that satisfy the required conditions. The steps of the algorithm are as follows.

(i) Subtract the smallest entity in each row from that row. Then subtract the smallest entity in each column from that column.

(ii) For each row assign the first zero which is not in any previously assigned column. If n zero has been found then stop the algorithm. Otherwise go to step 3.

(iii) Assuming that i_0 is the index of one of the rows for which in step two a zero has not been assigned. It is obvious that at least one zero is in the row. If the zero is in column j_0 there should be an assigned zero in the column j_0 otherwise the zero at (i_0, j_0) could be assigned. The index of this cell is (i_1, j_0). Starting at element (i_0, j_0) we construct a path consisting of alternating vertical and horizontal segments joining cells that are alternately assigned and not assigned. Continuing this process two possible outcomes are:

(a) If at a not assigned zero in element (i_k, j_k) an assigned zero can be found in the column j_k, it will be added to the sequence. Otherwise (no assigned zeros in the column) each assigned zero is changed to a not assigned zero and vice versa.

(b) If at an assigned zero in cell (i_{k+1}, j_k) a not assigned zero is found it will be added to the sequence. Otherwise label the column j_k as a necessary column and delete cells (i_{k+1}, j_k) and (i_k, j_k) from the sequence. Continue with the sequence. If there are no more cells in the sequence go to step 4.

(iv) Find k lines ($k<n$) that cover all the zeros of the cost matrix. Subtract the smallest uncovered element from the rest of the uncovered elements and add it to the elements covered twice. Go to step 2. It should be noted that Hungarian algorithm is strongly polynomial (Munkres, 1957) and the time complexity is

$$(11n^3 + 12n^2 + 31n)/6 \tag{6}$$

4. Comparison to the Current Pick-and-place Algorithms

In order to compare the optimum assignment with the current algorithms a set of simulations has been developed. The simulations generate a random placement of product and perform the pick and placing based on greedy algorithm and Hungarian algorithm. Fig. 2 shows the simulation setup and the decisions made for the handling.

In each of the cases after derivation of the assignment, the actual distance travelled by the robot to perform the handling is being calculated. The ratio between the actual distances travelled by the robot has been chosen as a measure of improvement of the developed algorithm.

Fig. 3 shows the improvement and the average improvement for a set of 1000 simulations. As it can be seen the average improvement obtained by employing the developed algorithm is more than 30%. It should be noted that there are various parameters that affect the improvement ratio e.g. initial position of products, statistical distribution and parameters of the products placement and the distances between the positions. As a result depending on the chosen parameters the improvement ratio can increase or decrease; however in all the cases it is beneficial.

Also one of the first questions asked is the effect of the number of products on the improvement ratio. In order to investigate it a similar set of simulations has been performed but in each iteration the number of products has been increased and the average improvement of a 1000 handling has been calculated. Fig. 4 shows the improvement versus the number of products. As it can be seen the average improvement has dropped significantly as the number of products gets higher than 100 however the fall in the improvement is because of the simulation setup which has a limited location for the input products and by increasing the number of product this space becomes filled which is the reason the algorithm does not improve as much as the cases with smaller number of products.

5. Conclusion

This paper has been proposed a new way to mathematically model and formulate the pick-and-place operation. The Hungarian algorithm has been employed to optimize the total distance travelled by the robot to perform the operation. The algorithm capability for increasing the utilization of the process has been demonstrated with the aid of the simulation software developed.

6. Acknowledgement

The authors gratefully wish to acknowledge the support of EPSRC under grant No. EP/E031394/1.

7. References

Ahamdi, J., Grotzinger, S. and Johnson, D. (1998), Component allocation and partitioning for a dual delivery placement machine, *Oper. Res.*, vol. 36, No. 2, pp. 176–191.

Carmon, T. F., Maimon, O. Z. and Dar-el, E. M. (1989), Group setup for printed circuit board assembly, *Int. J. Prod. Res.*, vol. 77, No. 10, pp. 1795–1810.

Egerváry, J. (1955), Matrixok kombinatorius tulajdonságairól (English translation by H.W. Kuhn): On combinatorial properties of matrices, *Logistics Papers*, George Washington University, issue 11, paper 4, pp. 1–11.

Foulds, L. R. and Hamacher, H. W. (1993), Optimal bin location and sequencing in printed circuit board assembly, *Eur. J. Oper. Res.*, vol. 66, pp. 279–290.

Grotzinger, S. (1992), Feeder assignment models for concurrent placement machines, *IIE Trans.*, vol. 24, No. 4, pp. 31–44.

Kleinberg, J. and Tardos, E. (2006), *Algorithm design*, Pearson/Addison-Wesley, Boston.

Kőnig, D. (1915), Vonalrendszerek és determinánsok (Hungarian; Line systems and determinants), *Mathematikai és Természettudományi Értesitő*, Vol. 33, pp. 221–229.

Kuhn, H. W. (1955), The Hungarian Method for the assignment problem, *Naval Research Logistics Quarterly*, Vol. 2, pp. 83–97.

Kuhn, H. W. (1956), Variants of the Hungarian method for assignment problems, *Naval Research Logistics Quarterly*, Vol. 3, pp. 253–258.

Kumar, R. and Li, H. (1988), Integer programming approach to printed circuit board assembly time optimization, *Oper. Res.*, Vol. 36, No. 2, pp. 192-201.

Leipala, T., and Nevalainen, O. (1989), Optimization of the movements of a component placement machine, *Eur. J. Oper. Res.*, vol. 38, pp. 167–177.

Mainmon, O. and Shtub, A. (1991), Grouping method for printed circuit board assembly, *Int. J. Prod. Res.*, vol. 29, No. 7, pp. 1379–1390.

Monge, G. (1781), Mémoire sur la théorie des déblais et des remblais, *Histoire de l'Académie Royale des Sciences*, pp. 666–704.

Moyer, L. K. and Gupta, S. M. (1996), SMT feeder slot assignment for predetermined component placement paths, *J. Electron. Manufact.*, vol. 6, No. 3, pp. 173–192.

Munkres, J. (1957), Algorithms for the Assignment and Transportation Problems, *Journal of the Society for Industrial and Applied Mathematics*, Vol. 5, No. 1, pp. 32–38.

Sadiq, M., Landers, T. L. and Taylor, G. D. (1993), A heuristic algorithm for minimizing total production time for a sequence of jobs on a surface mount placement machine, *Int. J. Prod. Res.*, vol. 31, No. 6, pp. 1327–1341.

Shih, W., Srihari, K., and Adriance, J. (1996), Expert system based placement sequence identification for surface mount PCB assembly, *Int. J. Adv. Manufact. Technol.*, vol. 11, pp. 413–424.

Sohn, J. and Park, S. (1996), Efficient operation of a surface mounting machine with a multihead turret, *Int. J. Prod. Res.*, vol. 34, No. 4, pp. 1131–1143.

Su,Y.-Y. and Srihari, K. (1996), Placement sequence identification using artificial neural networks in surface mount PCB assembly, *Int. J. Adv. Manufact. Technol.*, vol. 11, pp. 285–299.

Yeo, S. H., Low, C. W. and Yong, K. H. (1996), A rule-based frame system for concurrent assembly machines, *Int. J. Adv. Manufact. Technol.*, vol. 12, pp. 370–376.

Figure captions

Figure 1: Scattered products on conveyor belt (green circles) and final positions (red circles)

Figure 2: Pick and placing with (left) and without (right) utilization of the Hungarian algorithm, blue asterisks are the products and the red asterisks are the positions

Figure 3: Improvement of Hungarian algorithm in 1000 tests for a 25 product and a 5 by 5 position with a rectangular arrangement (same as Fig. 1) where *a*=*5*, *b*=*5*, x_0=20 and y_0=5.

The coordinates of the products are from a uniform distribution where x and y coordinates have uniform distributions between 0 and 15 and 0 and 30 respectively.

Figure 4: Improvement of the algorithm versus the number of products where a=5, b=5, x_0=20, y_0=5