Journal of the Operations Research Society of Japan
Vol. 55, No. 3, September 2012, pp. 181–191

CONSTANT FACTOR APPROXIMATION ALGORITHMS FOR REPETITIVE ROUTING PROBLEMS OF GRASP-AND-DELIVERY ROBOTS IN PRODUCTION OF PRINTED CIRCUIT BOARDS

Yoshiyuki Karuno
Kyoto Institute of Technology
Hiroshi Nagamochi   Aleksandar Shurbevski
Kyoto University

(Received April 15, 2011; Revised April 2, 2012)

Abstract In this paper, we consider a repetitive routing problem which we find on a printed circuit board assembly line. There are m different printed circuit boards to be processed. As an automated manipulator embeds electronic parts in a printed circuit board from above, n identical pins from underneath protect it against overbending. A dedicated pin configuration is designed for each printed circuit board so that pins do not obstruct its own circuit. A single grasp-and-delivery robot transfers the pins one by one to arrange them from a configuration to another one. Given an initial configuration of pins and a permutable set of m required configurations, our repetitive routing problem asks to find a configuration sequence, i.e., a processing order of m printed circuit boards, and a transfer route of the grasp-and-delivery robot so that the route length over all m transitions is minimized. We first design a polynomial time approximation algorithm with factor four to a restricted version of the repetitive routing problem with a non-permutable set of configurations, i.e., with a fixed processing order of m printed circuit boards. Applying the procedure, we then show that the repetitive routing problem with a permutable set of configurations admits a polynomial time approximation algorithm with factor six.

Keywords: Combinatorial optimization, repetitive routing problems, approximation algorithms, performance guarantees, constant factors

1. Introduction

In this paper, we consider a repetitive routing problem of a single grasp-and-delivery robot, which is used on a printed circuit board assembly line. The assembly line can process at most one printed circuit board at a time. A new unprocessed printed circuit board is brought into the assembly line along an automatic guided railway, and a finished printed circuit board is released from there by the automatic guided railway.

There are m different printed circuit boards to be processed. As a printed circuit board is processed, which means in practical situation that an automated manipulator embeds electronic parts in the printed circuit board from above, n identical pins support the printed circuit board from underneath to prevent it from overbending. (In this paper, we do not treat any automated manipulator routing issue, e.g., see Srivastav, Schroeter and Michel [13] for such an issue.) The board size is common to all m printed circuit boards, and pins are scattered in a given bounded two-dimensional area where distances between possible locations of pins are Euclidean.

Let $\mathcal{J} = \{J_1, J_2, \ldots, J_m\}$ denote a set of m printed circuit boards. A printed circuit board $J_i \in \mathcal{J}$ has its own circuit pattern, and a dedicated pin configuration C_i is designed for each J_i so that pins do not obstruct its circuit. This implies that if printed circuit board J_i is followed by another J_j in a processing order of m printed circuit boards, pin configuration C_i must be changed to C_j during the time interval between releasing J_i and sending for J_j. A single grasp-and-delivery robot works for changing pin configurations.

The grasp-and-delivery robot can grasp at most one pin at a time by using its fingers, i.e., it is a material handling device like a human hand with unit capacity. After grasping a pin, it can deliver the pin from the current location to a different location in the given bounded area. In this paper, we assume that the travel lengths of the grasp-and-delivery robot between possible locations of pins are symmetric and satisfy triangle inequality (since the distances between possible locations of pins in the given bounded area are Euclidean).

Let $\pi = (s_1, s_2, \ldots, s_m)$ denote a *configuration sequence*, which is a permutation on the index set $I = \{1, 2, \ldots, m\}$ of m printed circuit boards. It can also represent a processing order of m printed circuit boards. For $i = 1, 2, \ldots, m$, the grasp-and-delivery robot arranges n pins from pin configuration $C_{s_{i-1}}$ to next C_{s_i} by transferring the pins one by one, where we define by C_0 a given initial configuration and $s_0 = 0$ for notational convenience. We refer to such an arrangement from a pin configuration to another one as a *transition*. The repetitive routing problem asks to find a configuration sequence and a transfer route of the grasp-and-delivery robot so that the route length over all m transitions is minimized.

Note that for $i = 1, 2, \ldots, m$, current configuration $C_{s_{i-1}}$ in the i-th transition has been the next required one in the preceding $(i-1)$-th transition. In every transition except for the first one, the grasp-and-delivery robot starts by grasping the pin transferred last in the preceding transition, since it is not advantageous for the grasp-and-delivery robot to start by grasping any other pin. We regard this as the *constraint of consecutiveness*. (In the next section, we shall provide an example to show the constraint of consecutiveness, i.e., Figure 1.)

A restricted version of the repetitive routing problem with a non-permutable set of configurations, i.e., with a fixed processing order of printed circuit boards, admits a polynomial time approximation algorithm with factor two (see Karuno, Nagamochi and Shurbevski [9]), which is based on a weighted matroid intersection algorithm (see Frank [5], and see also a textbook by Korte and Vygen [11]). However, it was left to investigate whether the repetitive routing problem with a permutable set of configurations admits a constant factor approximation algorithm. In this paper, we first design a new polynomial time approximation algorithm with factor four to the repetitive routing problem with a non-permutable set of configurations. Then, applying the procedure, we propose a polynomial time approximation algorithm with factor six to the repetitive routing problem with a permutable set of configurations.

2. Problem Description

We are given n identical pins ($n \geq 2$), a set $\mathcal{J} = \{J_1, J_2, \ldots, J_m\}$ of m printed circuit boards to be processed ($m \geq 1$), and a set $\mathcal{C} = \{C_0, C_1, \ldots, C_m\}$ of $m+1$ configurations of pins, where C_0 is the initial configuration and C_i is the configuration required for printed circuit board J_i ($i = 1, 2, \ldots, m$). We also define by C_i the set of n points corresponding to the pin locations required for printed circuit board J_i. We assume that for each $i = 0, 1, \ldots, m$, n pin locations in configuration C_i are different from each another, and that configuration C_i is different from any other configuration C_j ($i \neq j$).

For each pair of distinct configurations C_i and C_j, we first define a weighted complete graph $K_{i,j} = (V_{i,j} = C_i \cup C_j, E_{i,j} = V_{i,j} \times V_{i,j})$, where $V_{i,j}$ is a set of $2n$ points (which may be *vertices* in this context) and $E_{i,j}$ is a set of edges. Let $u \in V_{i,j}$ and v $(\neq u) \in V_{i,j}$ be two end points of an edge $e \in E_{i,j}$, i.e., $e = (u, v)$. Then, a non-negative weight $w(u, v)$ is associated with edge $e = (u, v)$ which represents the travel length of the grasp-and-delivery robot from point u to point v. The edge weights are symmetric, i.e., $w(u, v) = w(v, u)$, and satisfy triangle inequality, i.e., edge weight function w is a metric. Also, the sum of weights $w(e)$ of edges e belonging to a graph H is denoted by $w(H)$.

In this paper, the initial position of the grasp-and-delivery robot, i.e., the pin to be transferred first in the first transition is prescribed, whose point we define by $u_0 \in C_0$. When the grasp-and-delivery robot transfers a pin from a location to another one, we call the performance a *transfer move*. On the other hand, when it goes without taking a pin from a location to another one, we call the performance an *empty move*. Note that the travel length of the grasp-and-delivery robot from point u to point v has been defined to be $w(u, v)$ for either type of move.

We also define two kinds of subgraphs $G_i = (C_i, C_i \times C_i)$ and $B_{i,j} = (C_i \cup C_j, C_i \times C_j)$ of $K_{i,j}$, where G_i is a complete graph with edge weight w and $B_{i,j}$ is a complete bipartite graph with edge weight w. Note that graph $K_{i,j}$ is the edge-disjoint union of subgraphs G_i, G_j and $B_{i,j}$.

A configuration sequence $\pi = (s_1, s_2, \ldots, s_m)$ is a permutation on the index set $I = \{1, 2, \ldots, m\}$ of printed circuit boards to be processed. It can also represent a processing order of m printed circuit boards. For $i = 1, 2, \ldots, m$, in the i-th transition the grasp-and-delivery robot arranges n pins from current configuration $C_{s_{i-1}}$ to next C_{s_i} by transferring the pins one by one, where $s_0 = 0$ for notational convenience.

Let $\langle C_i \mapsto C_j \rangle$ denote a transition of pin configuration from C_i to C_j. For a transition $\langle C_i \mapsto C_j \rangle$, let

$$\sigma_i = (\sigma_i(1), \sigma_i(2), \ldots, \sigma_i(n))$$

denote a permutation on n points of C_i, and let

$$\tau_j = (\tau_j(1), \tau_j(2), \ldots, \tau_j(n))$$

denote a permutation on n points of C_j. Then, a trajectory of the grasp-and-delivery robot in transition $\langle C_i \mapsto C_j \rangle$ can be represented as a Hamiltonian path $P_{i,j}$ in weighted complete bipartite graph $B_{i,j}$ such that

$$\sigma_i(1) \longrightarrow \tau_j(1) \underbrace{\longrightarrow}_{\emptyset} \sigma_i(2) \longrightarrow \tau_j(2) \underbrace{\longrightarrow}_{\emptyset} \cdots \underbrace{\longrightarrow}_{\emptyset} \sigma_i(n) \longrightarrow \tau_j(n), \tag{2.1}$$

where points in C_i and C_j appear alternatively (since the grasp-and-delivery robot has the unit capacity). In Equation (2.1), we express by $\emptyset$ an empty move of the grasp-and-delivery robot. We refer to such a path $P_{i,j}$ in $B_{i,j}$ as an *alternating Hamiltonian path*.

For a configuration sequence $\pi = (s_1, s_2, \ldots, s_m)$ and for $i = 2, 3, \ldots, m$, an alternating Hamiltonian path $P_{s_{i-2}, s_{i-1}}$ adopted in the $(i-1)$-th transition determines which pin is to be transferred first in the i-th transition. Then, we represent the constraint of consecutiveness as

$$\sigma_{s_{i-1}}(1) = \tau_{s_{i-1}}(n) \quad \text{for} \quad i = 1, 2, \ldots, m, \tag{2.2}$$

where we define $\tau_0(n) = u_0$ (and $s_0 = 0$ for notational convenience) since the initial position of the grasp-and-delivery robot is prescribed by $u_0 \in C_0$. We remind the readers that two permutations σ_i and τ_i are defined on common set C_i of n points.

For an alternating Hamiltonian path $P_{i,j}$ in a transition $\langle C_i \mapsto C_j \rangle$, the path length is denoted by $w(P_{i,j})$, which is the sum of weights of edges belonging to the path, i.e.,

$$w(P_{i,j}) = w(\sigma_i(1), \tau_j(1)) + \sum_{k=2}^{n} \Big\{ w(\tau_j(k-1), \sigma_i(k)) + w(\sigma_i(k), \tau_j(k)) \Big\}. \tag{2.3}$$

Let $\mathcal{P}_\pi = (P_{s_0,s_1}, P_{s_1,s_2}, \ldots, P_{s_{m-1},s_m})$ denote a sequence of m alternating Hamiltonian paths, which we call a *transfer route* (associated with a configuration sequence π). We refer to the sum of lengths of the m paths, defined by

$$L(\mathcal{P}_\pi) = \sum_{i=1}^{m} w(P_{s_{i-1},s_i}), \tag{2.4}$$

as the *route length*. A transfer route $\mathcal{P}_\pi$ is *feasible* if all the m alternating Hamiltonian paths satisfy the constraint of consecutiveness in Equation (2.2).

The repetitive routing problem asks to find a configuration sequence and a feasible transfer route of the grasp-and-delivery robot so that the route length is minimized. Let π^* and $\mathcal{P}^*_{\pi^*}$ denote the *optimal configuration sequence* and *optimal transfer route*, respectively. We refer to $L^* = L(\mathcal{P}^*_{\pi^*})$ as the *optimal route length*, or simply the *optimal value*. We call the repetitive routing problem RRP (Repetitive Routing with a Permutable set of configurations) for short. If the repetitive routing problem asks to find a feasible transfer route $\mathcal{P}^*_\pi$ that minimizes the route length associated with a fixed configuration sequence π, we call the problem RRN (Repetitive Routing with a Non-permutable set of configurations). We may call the optimal transfer route $\mathcal{P}^*_\pi$ for an instance of problem RRN (associated with a fixed configuration sequence π) a *minimal transfer route* for the corresponding instance of problem RRP. For problem RRN, we may omit notation π from a transfer route, i.e., we may express by $\mathcal{P} = (P_{0,1}, P_{1,2}, \ldots, P_{m-1,m})$ a transfer route.

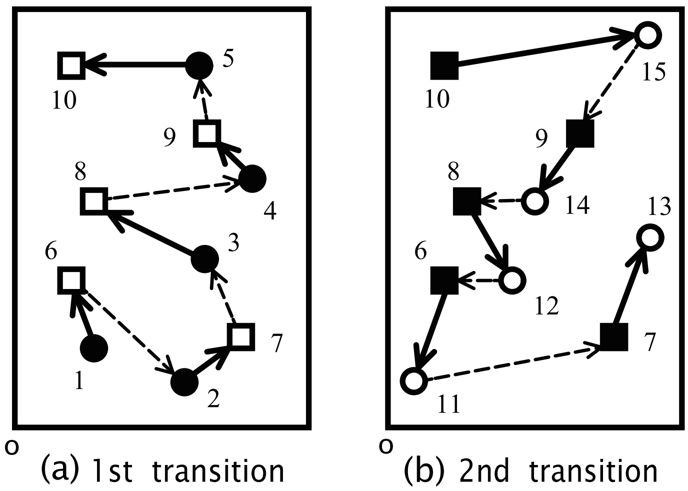

Figure 1: An example of transfer route with five pins

We provide an example with two printed circuit boards, i.e., $m = 2$, and five pins, i.e., $n = 5$, in Figure 1. Suppose that π $(= (s_1, s_2)) = (1, 2)$ is the configuration sequence of two printed circuit boards. We show in (a) of this figure the first transition, and in (b) the second transition. In (a), the five black circles are points corresponding to the initial configuration $C_0 = \{1, 2, 3, 4, 5\}$ of five pins, and the five white squares are points corresponding to the next required configuration $C_1 = \{6, 7, 8, 9, 10\}$. These five white squares become points corresponding to the current configuration in the second transition. The points are illustrated again as the five black squares in (b) of the figure. The new five white circles in (b) are points corresponding to the next required configuration $C_2 = \{11, 12, 13, 14, 15\}$ in the second transition.

In Figure 1, thick arrows indicate transfer moves of the grasp-and-delivery robot, while dotted thin arrows indicate its empty moves. The initial position of the grasp-and-delivery robot is given by $u_0 = 1 \in C_0$. The transfer route $\mathcal{P}_\pi = (P_{0,1}, P_{1,2})$ illustrated in this figure consists of the following two alternating Hamiltonian paths:

$$\begin{aligned} P_{0,1} : 1 &\to 6 \to 2 \to 7 \to \\ &3 \to 8 \to 4 \to 9 \to 5 \to 10, \quad \text{and} \\ P_{1,2} : 10 &\to 15 \to 9 \to 14 \to \\ &8 \to 12 \to 6 \to 11 \to 7 \to 13, \end{aligned}$$

where $\sigma_0(1) = 1$ and $\sigma_1(1) = \tau_1(5) = 10$, i.e., the transfer route meets the constraint of consecutiveness in Equation (2.2).

3. Non-permutable Configurations

In this section, we design an approximation algorithm with factor four to problem RRN, i.e., a restricted version of problem RRP. The proposed approximation algorithm adopts an individual-and-concatenate approach. That is, we compute an (approximate) alternating Hamiltonian path $P'_{i,j}$ in each $\langle C_i \mapsto C_j \rangle$ of m transitions in a step-by-step manner, concatenating the alternating Hamiltonian paths so that we obtain an approximate transfer route $\mathcal{P}' = (P'_{0,1}, P'_{1,2}, \ldots, P'_{m-1,m})$.

In a transition $\langle C_i \mapsto C_j \rangle$ of pin configuration, let T^*_i (resp., T^*_j) denote a minimum cost spanning tree in graph G_i (resp., G_j) with edge weight w, and let $M^*_{i,j}$ denote a minimum cost matching in complete bipartite graph $B_{i,j}$ with edge weight w. Recall that weighted complete graph $K_{i,j}$ is the edge-disjoint union of G_i, G_j and $B_{i,j}$.

Lemma 3.1. *Let $P_{i,j}$ be an alternating Hamiltonian path in weighted complete bipartite graph $B_{i,j}$ (and hence in weighted complete graph $K_{i,j}$). Then it holds*

$$w(P_{i,j}) \geq \max\{w(T^*_i), w(T^*_j), w(M^*_{i,j})\}. \tag{3.1}$$

Proof. The minimum cost spanning tree T^*_i (resp., T^*_j) is a subgraph of $K_{i,j}$ that connects all points in C_i (resp., C_j) with a minimum cost. On the other hand, by traversing alternating Hamiltonian path $P_{i,j}$ with shortcuts which skip over the points in C_j (resp., C_i), we obtain a path P_i (resp., P_j) which connects all points in C_i (resp., C_j). By noting that edge weight function w is a metric and that the resulting path P_i (resp., P_j) is a spanning tree in graph G_i (resp., G_j), we have $w(P_{i,j}) \geq w(T^*_i)$ (resp., $w(P_{i,j}) \geq w(T^*_j)$). Also, since $P_{i,j}$ contains a perfect matching between C_i and C_j, it must hold $w(P_{i,j}) \geq w(M^*_{i,j})$. □

Let $H_{i,j}$ denote an *alternating Hamiltonian cycle* in complete bipartite graph $B_{i,j}$ (and hence in complete graph $K_{i,j}$), where points in C_i and C_j appear alternatively. As for an alternating Hamiltonian path $P_{i,j}$ (see Equation (2.1)), it can be represented by

$$\sigma_i(1) \longrightarrow \tau_j(1) \underbrace{\longrightarrow}_{\emptyset} \sigma_i(2) \longrightarrow \tau_j(2) \underbrace{\longrightarrow}_{\emptyset} \cdots \underbrace{\longrightarrow}_{\emptyset} \sigma_i(n) \longrightarrow \tau_j(n) \underbrace{\longrightarrow}_{\emptyset} \sigma_i(1), \tag{3.2}$$

using two permutations $\sigma_i = (\sigma_i(1), \sigma_i(2), \ldots, \sigma_i(n))$ and $\tau_j = (\tau_j(1), \tau_j(2), \ldots, \tau_j(n))$ on points in C_i and C_j, respectively.

Lemma 3.2. *In weighted complete bipartite graph $B_{i,j}$ (and hence in weighted complete graph $K_{i,j}$), there exists an alternating Hamiltonian cycle $H'_{i,j}$ such that*

$$w(H'_{i,j}) \leq w(T^*_i) + w(T^*_j) + 2w(M^*_{i,j}). \tag{3.3}$$

Proof. We first convert the minimum cost spanning tree T_i^* into a Hamiltonian cycle H_i on C_i with $w(H_i) \leq 2w(T_i^*)$ by traversing T_i^* in a depth-first manner with shortcuts (e.g., see Johnson and Papadimitriou [8]). During the process of visiting each point $u \in C_i$, we also visit $v \in C_j$ with $(u, v) \in M_{i,j}^*$ by making a trip to v via edge (u, v) before moving to the next point in C_i. By exchanging the roles between i and j, we can make the similar argument, i.e., we have $w(H_j) \leq 2w(T_j^*)$. Therefore, we obtain a desired cycle $H'_{i,j}$ in weighted complete bipartite graph $B_{i,j}$ such that it satisfies $w(H'_{i,j}) \leq \min\{2w(T_i^*), 2w(T_j^*)\} + 2w(M_{i,j}^*)$. □

The minimum cost spanning trees T_i^* and T_j^* can be computed in polynomial time, e.g., in $O(n^2)$ time by Prim's algorithm with Fibonacci heaps (see Prim [12], and see also Fredman and Tarjan [6]). Also, the minimum cost matching $M_{i,j}^*$ can be computed in polynomial time, e.g., in $O(n^3)$ time by Edmonds' algorithm [4] (see also Korte and Vygen [11]).

The proposed approximation algorithm is described as follows. For a given instance of problem RRN associated with a fixed configuration sequence $\pi = (1, 2, \ldots, m)$, in each transition $\langle C_{i-1} \mapsto C_i \rangle$ of $i = 1, 2, \ldots, m$, we first obtain an alternating Hamiltonian cycle $H'_{i-1,i}$ that satisfies Equation (3.3) in polynomial time. Next, we extract an alternating Hamiltonian path $P'_{i-1,i}$ from $H'_{i-1,i}$. To meet the constraint of consecutiveness in Equation (2.2), we delete one arc from $H'_{i-1,i}$ such that it is the incoming arc to point $\tau_{i-1}(n)$. It is obvious that the resulting path is an alternating Hamiltonian path starting from point $\sigma_{i-1}(1) := \tau_{i-1}(n)$ in $B_{i-1,i}$. Finally, we obtain a feasible transfer route $\mathcal{P}' = (P'_{0,1}, P'_{1,2}, \ldots, P'_{m-1,m})$ by concatenating the m alternating Hamiltonian paths. We call the approximation algorithm AAN (Approximation Algorithm for the problem with a Non-permutable set of configurations) for short. A time complexity of algorithm AAN is $O(mn^3)$. By Lemma 3.2, we have

$$w(P'_{i-1,i}) \leq w(H'_{i-1,i}) \leq w(T_{i-1}^*) + w(T_i^*) + 2w(M_{i-1,i}^*). \tag{3.4}$$

Let $\mathcal{P}^* = (P_{0,1}^*, P_{1,2}^*, \ldots, P_{m-1,m}^*)$ be an optimal transfer route for a given instance of problem RRN, where $P_{i-1,i}^*$ is the alternating Hamiltonian path in the i-th transition for $i = 1, 2, \ldots, m$. By Lemma 3.1, we have $w(P_{i-1,i}^*) \geq \max\{w(T_{i-1}^*), w(T_i^*), w(M_{i-1,i}^*)\}$ for $i = 1, 2, \ldots, m$. Hence, together with Equation (3.4), we also have $w(P'_{i-1,i}) \leq 4 \times w(P_{i-1,i}^*)$ for $i = 1, 2, \ldots, m$, which implies

$$L(\mathcal{P}') = \sum_{i=1}^{m} w(P'_{i-1,i}) \leq 4 \times \sum_{i=1}^{m} w(P_{i-1,i}^*) = 4L(\mathcal{P}^*). \tag{3.5}$$

Therefore, we obtain the following theorem.

Theorem 3.1. *For an instance of problem* RRN, *algorithm* AAN *delivers a 4-approximation transfer route in polynomial time.*

As mentioned before, it has already been known that problem RRN is 2-approximable in polynomial time. The known time complexity is $O(mn^7)$ (see Karuno, Nagamochi and Shurbevski [9], and see also Baltz and Srivastav [1], Chalasani and Motwani [2] for the bipartite traveling salesperson problem). However, to derive a constant factor of approximation for problem RRP (which is the main target of this paper) using constant factor approximation techniques for the traveling salesperson problem (see Christofides [3] and Hoogeveen [7]), we shall apply the procedure of algorithm AAN with factor four in the next section. The procedure applied in the 2-approximation algorithm for problem RRN presented by Karuno, Nagamochi and Shurbevski [9] delivers different matroid intersections between transition $\langle C_i \mapsto C_j\rangle$ and the inverse transition $\langle C_j \mapsto C_i\rangle$. On the other hand, $H'_{i,j}$ obtained for transition $\langle C_i \mapsto C_j\rangle$ in Lemma 3.2 can also be regarded as an alternating Hamiltonian cycle $H'_{j,i}$ in weighted complete bipartite graph $B_{j,i}$ $(= B_{i,j})$ for the inverse one $\langle C_j \mapsto C_i\rangle$ such that $w(H'_{j,i}) \le w(T^*_i) + w(T^*_j) + 2w(M^*_{i,j})$. This makes it easier to define a symmetric weight as a certain *cost* incurred by every pair of configurations. (In the next section, we shall define a symmetric weight function ω between $m+1$ configurations.)

4. Permutable Configurations

In this section, we return to problem RRP. To determine an approximate configuration sequence of m printed circuit boards, we construct a *universal graph* $\mathcal{G} = (\mathcal{C}, \mathcal{E} = \mathcal{C} \times \mathcal{C})$ by preparing a node for each configuration and an arc between every two nodes in $\mathcal{C}$. Regarding Equations (3.1) and (3.3), a weight $\omega(a)$ of an arc $a = (C_i, C_j)$ is defined to be

$$\omega(C_i, C_j) = \frac{1}{4}\left[w(T^*_i) + w(T^*_j) + 2w(M^*_{i,j})\right], \quad i, j = 0, 1, \ldots, m,\ i \ne j. \tag{4.1}$$

It is obvious that $\omega(C_i, C_j) = \omega(C_j, C_i)$ for any two distinct indices i and j.

Lemma 4.1. *Weight function ω is a metric in $\mathcal{G}$.*

Proof. It suffices to show that $\omega(C_i, C_j) + \omega(C_j, C_k) \ge \omega(C_i, C_k)$ for any three indices i, j and k. Note that given two minimum cost matchings $M^*_{i,j}$ and $M^*_{j,k}$, matching $M'_{i,k} = \{\ (u, u'') \mid (u, u') \in M^*_{i,j},\ (u', u'') \in M^*_{j,k},\ u \in C_i,\ u' \in C_j,\ u'' \in C_k\}$ satisfies $w(M'_{i,k}) \le w(M^*_{i,j}) + w(M^*_{j,k})$ due to triangle inequality of edge weight w (see Figure 2). Hence, it holds

$$w(M^*_{i,j}) + w(M^*_{j,k}) \ge w(M'_{i,k}) \ge w(M^*_{i,k}).$$

Therefore, we have

$$\begin{aligned} 4\omega(C_i, C_j) + 4\omega(C_j, C_k) &= w(T^*_i) + w(T^*_j) + 2w(M^*_{i,j}) + w(T^*_j) + w(T^*_k) + 2w(M^*_{j,k}) \\ &\ge w(T^*_i) + w(T^*_k) + 2w(M^*_{i,k}) \\ &= 4\omega(C_i, C_k), \end{aligned}$$

which completes the proof. □

Let $\mathcal{T}^*$ denote a minimum cost spanning tree of $\mathcal{G}$ with weight ω, which can be computed in $O(m^2)$ time by Prim's algorithm with Fibonacci heaps [6, 12], and let $\omega(\mathcal{T}^*)$ denote the sum of weights of arcs belonging to $\mathcal{T}^*$. Let $\mathcal{F}$ denote a Hamiltonian path starting from the node for the initial configuration C_0 in $\mathcal{G}$, and let $\omega(\mathcal{F})$ denote the sum of weights of arcs belonging to $\mathcal{F}$. By definition, $\mathcal{F}$ is also a spanning tree of $\mathcal{G}$. Notice that an optimal configuration sequence π^* specifies a certain Hamiltonian path $\mathcal{F}^{\rm O}$ starting from the node for the initial configuration C_0 in $\mathcal{G}$, and by Lemma 3.1 and Equation (4.1), $\omega(\mathcal{F}^{\rm O})$ is a lower bound on the route length. Further, let $\mathcal{F}^*$ denote a shortest Hamiltonian path starting from the node for the initial configuration C_0 in $\mathcal{G}$ with weight ω. Then, we have the following lemma.

Lemma 4.2. *For an instance of problem* RRP*, let L^* be the optimal route length. Then, it satisfies*

$$L^* \ge \omega(\mathcal{F}^*) \ge \omega(\mathcal{T}^*). \tag{4.2}$$

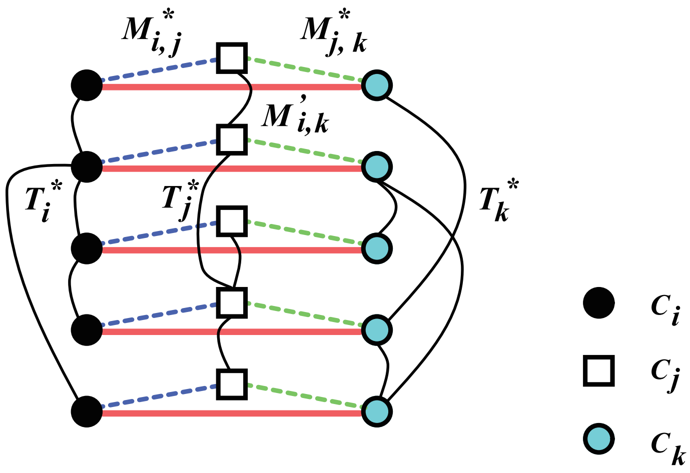

Figure 2: An illustration for minimum cost spanning trees T_i^*, T_j^* and T_k^* on C_i, C_j and C_k, respectively, and minimum cost matchings $M_{i,j}^* \subseteq C_i \times C_j$ and $M_{j,k}^* \subseteq C_j \times C_k$, and matching $M'_{i,k} \subseteq C_i \times C_k$ with $w(M'_{i,k}) \leq w(M_{i,j}^*) + w(M_{j,k}^*)$

For the traveling salesperson problem with metric distances, Christofides' algorithm delivers a tour whose length is less than $\frac{3}{2}$ times the optimum tour length (see Christofides [3]). For the problem of finding a shortest Hamiltonian path starting from a prespecified starting node, a modification of Christofides' algorithm has been presented by Hoogeveen [7], which delivers a Hamiltonian path whose length is no more than $\frac{3}{2}$ times the length of an optimal Hamiltonian path (see also Johnson and Papadimitriou [8]). The algorithm is sketched as follows:

(1) Find a minimum cost spanning tree $\mathcal{T}^*$ of the universal graph $\mathcal{G}$;
(2) Construct the set $S_{\rm odd}$ of all odd-degree nodes in $\mathcal{T}^*$;
(3) If the node for the initial configuration C_0 is an odd-degree node in $\mathcal{T}^*$, then remove it from the set $S_{\rm odd}$. Otherwise, add it to the set $S_{\rm odd}$. We define by S the resulting set of nodes.
(4) After augmenting a dummy node in an obvious fashion, compute a minimum cost perfect matching $\mathcal{M}^*$ on the set S (the matching $\mathcal{M}^*$ excludes the arc which is incident with the dummy node);
(5) For the subgraph of $\mathcal{G}$ with arcs of $\mathcal{T}^*$ and $\mathcal{M}^*$, if the node for the initial configuration C_0 belongs to S and it is not incident with any arc of $\mathcal{M}^*$, then delete an arbitrary arc which is incident with the node;
(6) For the resulting graph (notice that the graph has two odd-degree nodes, and one of them is the node for the initial configuration C_0), find an Eulerian path starting from the node for the initial configuration C_0;
(7) Convert the Eulerian path into a Hamiltonian path $\mathcal{F}'$ by shortcuts.

Since the resulting graph in (6) has $O(m)$ nodes and $O(m)$ arcs, an Eulerian path from the node for the initial configuration C_0 can be found in $O(m)$ time (e.g., see Korte and Vygen [11]). In Lemma 4.2, we have seen that $w(\mathcal{T}^*) \leq L^*$. The proof of Hoogeveen [7] can state the following lemma.

Lemma 4.3. [7] *For an instance of problem* RRP*, let L^* be the optimal route length. Then, it satisfies*

$$L^* \geq 2\omega(\mathcal{M}^*). \tag{4.3}$$

Since weight function ω is a metric in the universal graph $\mathcal{G}$ (see Lemma 4.1), by applying the algorithm of Hoogeveen's modification, we can find a Hamiltonian path $\mathcal{F}'$ in $\mathcal{G}$ starting from the the node for the initial configuration C_0 in polynomial time such that its length satisfies

$$\frac{\omega(\mathcal{F}')}{\omega(\mathcal{F}^*)} \leq \frac{3}{2}. \tag{4.4}$$

Without loss of generality, such a configuration sequence that corresponds to $\mathcal{F}'$ is denoted by $\pi = (1, 2, \ldots, m)$. Instead of finding a minimal transfer route (associated with π), we construct an approximate transfer route associated with π by calling algorithm AAN proposed in the previous section.

Let $\mathcal{P}' = (P'_{0,1}, P'_{1,2}, \ldots, P'_{m-1,m})$ be a transfer route where each $P'_{i-1,i}$ of the m alternating Hamiltonian paths is obtained by algorithm AAN. We refer to the proposed approximation algorithm for obtaining $\mathcal{P}'$ as AAP (Approximation Algorithm for the problem with a Permutable set of configurations). For $i = 1, 2, \ldots, m$, we see that $w(P'_{i-1,i}) \leq 4\omega(C_{i-1}, C_i)$ by Equations (3.4) and (4.1). Hence, together with Lemma 4.2 and Equation (4.4), we have

$$L(\mathcal{P}') = \sum_{i=1}^{m} w(P'_{i-1,i}) \leq 4 \times \left(\sum_{i=1}^{m} \omega(C_{i-1}, C_i) \right) = 4\omega(\mathcal{F}') \leq 6\omega(\mathcal{F}^*) \leq 6L^*. \tag{4.5}$$

Therefore, we obtain the following theorem.

Theorem 4.1. *Problem* RRP *is 6-approximable in polynomial time.*

Constructing $\mathcal{G}$ with weight ω requires $O(mn^2 + m^2n^3) = O(m^2n^3)$ time, since we compute the minimum cost spanning trees T_i^* for $i = 0, 1, \ldots, m$ and minimum cost matchings $M_{i,j}^*$ for $i, j = 0, 1, \ldots, m,\ i \neq j$. It takes $O(m^2)$ time to find a minimum cost spanning tree $\mathcal{T}^*$ of $\mathcal{G}$, and $O(m^3)$ time to find a minimum cost perfect matching $\mathcal{M}^*$ on the set S (which is defined in the algorithm of Hoogeveen's modification). After finding a Hamiltonian path $\mathcal{F}'$ in $O(m)$ time, algorithm AAP calls algorithm AAN, which requires $O(mn^3)$ time. Hence, a time complexity of algorithm AAP is evaluated as $O(m^3 + m^2n^3)$.

5. Concluding Remarks

In this paper, we considered a repetitive routing problem of a single grasp-and-delivery robot with unit capacity, which is used on a printed circuit board assembly line. Let m be the number of printed circuit boards to be processed. A printed circuit board is supported by n identical pins from underneath while it is processed. The grasp-and-delivery robot arranges n identical pins from their current configuration to the next required configuration by transferring the pins one by one. Given an initial configuration of n pins and a permutable set of m configurations, the repetitive routing problem asks to find a configuration sequence, i.e., a processing order of m printed circuit boards, and a feasible transfer route of the grasp-and-delivery robot so that the route length over all m transitions is minimized. In this paper, we first designed an $O(mn^3)$ time approximation algorithm with factor four to the repetitive routing problem with a non-permutable set of configurations, i.e., with a fixed processing order of printed circuit boards. Then, applying the procedure, we proposed an $O(m^3 + m^2n^3)$ time approximation algorithm with factor six to the repetitive routing problem with a permutable set of configurations.

However, the factor six seems to be still large. For future research, it would be interesting to investigate whether the problem admits a polynomial time approximation algorithm with a better constant factor.

Acknowledgment

This research has partially been supported by a Grant-in-Aid for Scientific Research (C) of the Japan Society for the Promotion of Science (JSPS). A part of this research has been allowed to be presented at International Symposium on Scheduling 2011 (ISS 2011), JSME/SSJ, July 2011, Osaka, Japan, and this paper is the refined version of [10], in which an approximation algorithm with factor eight is proposed to the repetitive routing problem with a permutable set of configurations.

References

[1] A. Baltz and A. Srivastav: Approximation algorithms for the Euclidean bipartite TSP. *Operations Research Letters*, **33** (2005), 403–410.

[2] P. Chalasani and R. Motwani: Approximating capacitated routing and delivery problems. *SIAM Journal on Computing*, **28** (1999), 2133–2149.

[3] N. Christofides: Worst-case analysis of a new heuristic for the travelling salesman problem. Technical Report 388, Graduate School of Industrial Administration, Carnegie-Mellon University (1976).

[4] J. Edmonds: Paths, trees, and flowers. *Canadian Journal of Mathematics*, **17** (1965), 449–467.

[5] A. Frank: A weighted matroid intersection algorithm. *Journal of Algorithms*, **2** (1981), 328–336.

[6] M.L. Fredman and R.E. Tarjan: Fibonacci heaps and their uses in improved network optimization problems. *Journal of the ACM*, **34** (1987), 596–615.

[7] J.A. Hoogeveen: Analysis of Christofides' heuristic: Some paths are more difficult than cycles. *Operations Research Letters*, **10** (1991), 291–295.

[8] D.S. Johnson and C.H. Papadimitriou: Performance guarantees for heuristics. In E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, and D.B. Shmoys (eds.): *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization* (Wiley, Chichester, 1985), 145–180.

[9] Y. Karuno, H. Nagamochi, and A. Shurbevski: An approximation algorithms with factor two for a repetitive routing problem of grasp-and-delivery robots. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, **15** (2011), 1103–1108.

[10] Y. Karuno, H. Nagamochi, and A. Shurbevski: Approximating cyclic routing problems of grasp-and-delivery robots in production of printed circuit boards. *Proceedings of International Symposium on Scheduling 2011 (ISS 2011)*, (2011), 247–252.

[11] B. Korte and J. Vygen: *Combinatorial Optimization: Theory and Algorithms, 3rd Edition* (Springer, Berlin, 2005, the Japanese language translation by T. Asano, T. Hirata, T. Ono, and Y. Asano, Springer Tokyo, Tokyo, 2005).

[12] R.C. Prim: Shortest connection networks and some generalization. *Bell System Technical Journal*, **36** (1957), 1389–1401.

[13] A. Srivastav, H. Schroeter, and C. Michel: Approximation algorithms for pick-and-place robots. *Annals of Operations Research*, **107** (2001), 321–338.

Yoshiyuki Karuno
Department of Mechanical and System Engineering
Kyoto Institute of Technology
Matsugasaki, Sakyo, Kyoto 606-8585, Japan
E-mail: karuno@kit.ac.jp