Feasibility of Motion Planning on Directed Graphs

Zhilin Wu[1] and Stéphane Grumbach[2]

[1] CASIA-LIAMA, zlwu@liama.ia.ac.cn
[2] INRIA-LIAMA, stephane.grumbach@inria.fr

Abstract. Because of irreversibility of movements, motion planning on directed graphs is much more intricate than that on graphs. Recently we showed that the feasibility of motion planning on acyclic and strongly connected directed graphs can be decided in time $O(nm)$ (n, m are respectively the number of vertices and arcs of the directed graph), but left the feasibility of motion planning on (general) directed graphs open. In this paper, we complete the solution by showing that the feasibility of motion planning on directed graphs can be decided in time $O(n^2m)$.

1 Introduction

Motion planning is a fundamental problem of robotics. It has been extensively studied [LaV06], and has numerous practical applications beyond robotics, such as in manufacturing, animation, games [MPG] as well as in computational biology [SA01,FK99]. The study of motion planning on graphs was proposed by Papadimitriou et al. [PRST94] to strip away the geometric considerations of the general motion planning problem and concentrate on the combinatorial aspects.

Motion planning on graphs is defined as follows. Suppose a graph $G = (V, E)$ is given. There is one robot at a source vertex s and some of the other vertices contain a movable obstacle. The objective is to move the robot from s to a destination vertex t with the smallest number of moves, where a move consists in moving an object (robot or obstacle) from one vertex to an adjacent vertex that contains a hole (if a vertex does not contain an object, then it is said to contain a hole; if an object is moved from v to w, we can also say that a hole is moved from w to v).

If there are too many obstacles, it may be impossible to move the robot from the source to the destination. So before considering the optimization problem, the decision problem whether a given instance of the problem of motion planning on graphs is feasible or not, should be considered first.

Motion planning on graphs is an abstraction of the practical problems, such as track transportation systems [Per88] and packet transfer in communication buffers of networks.

In practice, for the above two examples, tracks or links might be asymmetric. For instance, there may be unidirectional links in networks, especially in wireless networks, due to the heterogeneity of receiver and transmitter hardware

[MD02,JJ06]. This motivates the study of motion planning on directed versus undirected graphs.

Directed graphs (abbreviated as digraphs from now on) differ from undirected graphs mainly in that movements in digraphs are irreversible. In [WG08], we proposed two algorithms to decide the feasibility of motion planning on acyclic and strongly-connected digraphs in time $O(nm)$ (n, m are respectively the number of vertices and arcs).

For digraphs which are neither acyclic nor strongly connected, the motion planning problem may become much more tricky. For instance, the motion planning problem given in Fig. 1.(a) is feasible. Let C_s denote the strongly connected component of s, then if initially the hole in v_7 is moved into C_s through v_2 (see Fig. 1.(b)), the problem will become infeasible, which the reader can easily verify.

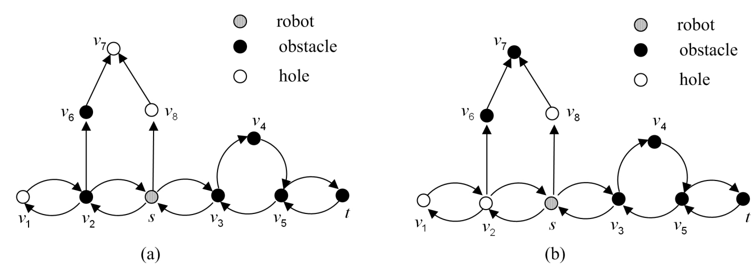

Fig. 1. Motion planning on digraphs

In this paper, we give a complete solution to the feasibility of motion planning on digraphs and show that it can be decided in time $O(n^2m)$. We distinguish between the cases whether C_s, the strongly connected component containing s, is trivial or not, and whether s and t belong to the same strongly connected component or not. If C_s is trivial, then the feasibility can be solved by combining the two algorithms for feasibility of motion planning on acyclic and strongly connected digraphs in [WG08]. Otherwise, a greedy strategy to move the outside holes into C_s can be designed to solve the feasibility problem.

Without loss of generality, we assume in this paper that in the motion planning problem, the source vertex s and the target vertex t of the robot are different, and there is at least one path from s to t.

The paper is organized as follows: In Section 2, some preliminaries are given. The structure of digraphs is discussed in Section 3. Then in Section 4, feasibility of motion planing on digraphs is solved case by case.

2 Preliminaries

The notations of this paper follow those in [Wes00,BJG00].

Let $D = (V, E)$ (resp. $G = (V, E)$) be a digraph (resp. graph), $X \subseteq V$. The sub-digraph (resp. subgraph) *induced by* X, denoted $D[X]$ (resp. $G[X]$), is the sub-digraph (resp. subgraph) containing all the vertices in X and all the arcs (resp. edges) between the vertices in X. And the sub-digraph (resp. subgraph) obtained from D (resp. G) by deleting all the vertices in X and all the arcs (resp. edges) with at least one of its endpoints in X, is denoted as $D - X$ (resp. $G - X$).

The *underlying graph* of a digraph $D = (V, E)$, denoted $\mathcal{G}(D)$, is the graph obtained from D by ignoring the directions of the arcs.

Let $G = (V, E)$ be a graph. The *biconnected-component graph* of G, denoted by $\mathcal{G}_{bc}(G)$, is a bipartite graph (V_{bc}, W_{bc}, E_{bc}) defined by

- V_{bc} contains the biconnected components of G;
- W_{bc} contains all $v \in V$ such that v is shared by at least two distinct biconnected components of G;
- E_{bc} is defined as follows: let $B \in V_{bc}$ and $w \in W_{bc}$, then $\{B, w\} \in E_{bc}$ iff $w \in V(B)$.

In [WG08], strongly biconnected digraphs were introduced to decide the feasibility of motion planning on strongly connected digraphs.

Definition 1. *Let D be a digraph. D is said to be strongly biconnected if D is strongly connected and $\mathcal{G}(D)$ is biconnected. The strongly biconnected components of D are the maximal strongly biconnected sub-digraphs of D.*

With regard to the feasibility of the motion planning problem, strongly biconnected digraphs have the following nice property.

Theorem 1 ([WG08]). *The motion planning problem on a strongly biconnected digraph D is feasible iff there is at least one hole in D.*

Strongly connected digraphs admit the following decomposition.

Theorem 2 ([WG08]). *Let $D = (V, E)$ be a nontrivial strongly connected digraph. Then the strongly biconnected components of D are those $D[V(B)]$, the sub-digraph of D induced by $V(B)$, where B is a biconnected component of $\mathcal{G}(D)$.*

Let $D = (V, E)$ be a strongly connected digraph, the *strongly-biconnected-component graph* of D, denoted $\mathcal{G}_{sbc}(D) = (V_{sbc}, W_{sbc}, E_{sbc})$, is the biconnected-component graph of $\mathcal{G}(D)$. Let $v \in V$, v is called a *branching vertex* if $v \in W_{sbc}$ and the degree of v is greater than 2 in $\mathcal{G}_{sbc}(D)$. A *chain* of $\mathcal{G}_{sbc}(D)$ is a path $B_0 v_1 B_1 \cdots B_{k-1} v_k B_k$ $(k \geq 1)$ in $\mathcal{G}_{sbc}(D)$ such that $|V(B_i)| = 2$ for all $1 \leq i \leq k-1$, and v_i is not a branching vertex for all $1 < i < k$.

The *length* of a chain is the number of vertices in W_{sbc} on the chain.

Since the biconnected-component graph of a graph is a tree [Wes00], it follows that $\mathcal{G}_{sbc}(D)$ is a tree as well.

Theorem 3 ([WG08]). *Feasibility of motion planning on acyclic and strongly connected digraphs can be decided in time $O(nm)$, where n, m are respectively the number of vertices and arcs of the digraph.*

3 Structure of digraphs

Let $D = (V, E)$ be a digraph. Then the vertex sets of strongly connected components of D form a partition of V.

Definition 2. *Let $D = (V, E)$ be a digraph. The strongly-connected-component digraph of D, $\mathcal{D}_{scc}(D) = (V_{scc}, E_{scc})$, is defined as follows:*

- *$V_{scc} = V_{tr} \cup V_{nt} \cup V_{pt}$, where*
  - *V_{tr} contains all $v \in V$ such that v is a trivial strongly connected component of D;*
  - *V_{nt} contains all nontrivial strongly connected components of D;*
  - *V_{pt} contains all $v \in V$ such that v belongs to some nontrivial strongly connected component of D (say C) and there is some $w \notin V(C)$ such that $(v, w) \in E$ or $(w, v) \in E$. Those v's are called the* ports *of C.*
- *E_{scc} is defined by the following two rules:*
  - *If $C \in V_{nt}$, $v \in V_{pt}$, and $v \in V(C)$, then $(C, v) \in E_{scc}$ and $(v, C) \in E_{scc}$;*
  - *If $v, w \in V_{tr} \cup V_{pt}$, $(v, w) \in E$, and v, w do not belong to the same strongly connected component, then $(v, w) \in E_{scc}$.*

Remark 1. If for each nontrivial strongly connected component C, we contract the set of vertices of $\mathcal{D}_{scc}(D)$ related to C, including $C \in V_{nt}$ and all $v \in V_{pt} \cap V(C)$, into one single vertex, and delete the resulted self-loops, then $\mathcal{D}_{scc}(D)$ will become the same as the classical definition of strongly-connected-component digraph in [BJG00]. □

4 Motion planning on digraphs

Throughout this section, let $D = (V, E)$ be a digraph, and $\mathcal{D}_{scc}(D) = (V_{scc}, E_{scc})$ ($V_{scc} = V_{tr} \cup V_{nt} \cup V_{pt}$) be the strongly-connected-component digraph of D.

Theorem 4. *Feasibility of motion planning on D can be decided in time $O(n^2m)$ where n, m are resp. the number of vertices and arcs of D.*

In the following, we design an algorithm to prove the theorem. We illustrate the main idea of the algorithm, but leave the correctness proof and the detailed complexity analysis to the full paper. We first introduce some notations.

Let C_s and C_t be the strongly connected components which s and t belong to respectively.

For each $v \in V$, let $h(v)$ denote the number of holes that are reachable from v, namely, to which there is a path from v in D.

Let V_{cr} denote the set of vertices $v \in V_{tr} \cup V_{pt}$ such that t is reachable from v, and v is reachable from s in D. The vertices in V_{cr} are called the *critical* vertices of the motion planning problem on D.

We consider motion planning on digraphs case by case:

Case I: C_s is trivial;
Case II: C_s is nontrivial and $C_s = C_t$;
Case III: C_s is nontrivial and $C_s \neq C_t$.

Note that since we assume that $s \neq t$, if C_s is trivial, then $C_s \neq C_t$.

4.1 Case I: C_s is trivial

We introduce some additional notations.

Let C be a nontrivial strongly connected component of D, $In(C)$ (resp. $Out(C)$) are used to denote the set of ports of C, i.e. vertices $v \in V_{pt} \cap V(C)$, such that there is some $w \in (V_{tr} \cup V_{pt}) \backslash V(C)$ satisfying that $(w, v) \in E$ (resp. $(v, w) \in E$). Vertices in $In(C)$ (resp. $Out(C)$) are called *inward* ports (resp. *outward* ports) of C. Note that $In(C) \cap Out(C)$ may be nonempty.

Let V_{cr}^{in} denote the set of vertices $v \in V_{cr}$ such that either $v \in V_{tr}$ and $v \neq s$, or $v \in In(C)$ for some nontrivial strongly connected component C. And let V_{cr}^{out} denote the set of vertices $v \in V_{cr}$ such that either $v \in V_{tr}$ and $v \neq t$, or $v \in Out(C)$ for some nontrivial strongly connected component C such that $C \neq C_t$.

For $v \in V_{cr}^{in}$, define $h_{in}(v)$ as follows:

> Let $w \in V_{cr}$ such that $(w, v) \in E_{scc}$, imagine that the robot is in w. If the robot can be moved from w to t under the restriction that the first move of the robot is from w to v, then $h_{in}(v)$ is the minimal number of (distinct) holes used during the movement of the robot from w to t; otherwise, $h_{in}(v) := \infty$.

For $v \in V_{cr}^{out}$, define $h_{out}(v)$ as follows:

> Imagine that the robot is in v. If the robot can be moved from v to t under the restriction that the first move of the robot is from v to some $w \in V_{cr}$ such that $(v, w) \in E_{scc}$, then $h_{out}(v)$ is the minimal number of holes used during the movement of the robot from v to t; otherwise, $h_{out}(v) := \infty$.

The algorithm for deciding the feasibility of motion planning on digraphs in **Case I** goes as follows: Starting from the vertices in $V_{cr} \cap V(C_t)$, compute $h_{in}(v)$ and $h_{out}(v)$ for all $v \in V_{cr}$ inductively in a backward way. When these computations are finished, the algorithm reports "yes" (the motion planning problem is feasible) iff $h_{out}(s) < \infty$.

Initial step: For vertices in $V_{cr} \cap V(C_t)$,

- If C_t is trivial, then $t \in V_{cr}^{in}$ and $t \notin V_{cr}^{out}$: if $h(t) \geq 1$, then $h_{in}(t) := 1$, otherwise $h_{in}(t) := \infty$;
- If C_t is nontrivial, then $In(C_t) \subseteq V_{cr}^{in}$ and $Out(C_t) \cap V_{cr}^{out} = \emptyset$: for $v \in In(C_t)$, if $h(v) \geq MinNum(C_t, v, t) + 1$, then $h_{in}(v) := MinNum(C_t, v, t) + 1$, otherwise, $h_{in}(v) := \infty$.

Remark 2. $MinNum(D, v, w)$ is used in [WG08] to compute the minimal number of holes used to move the robot from v to w in a strongly connected digraph D over all instances of the motion planning on D such that v, w are respectively the source and the destination. $MinNum(D, v, w)$ works as follows:

- If $v = w$, then return 0;

– Otherwise if v, w belong to the same strongly biconnected component of D, then return 1 (according to Theorem 1);
– Otherwise, let $P = B_0 v_1 B_1 ... B_{r-1} v_r B_r$ $(r \geq 1)$ be the path in $\mathcal{G}_{sbc}(D)$ such that $v \in B_0$, $w \in B_r$, $v \neq v_1$ and $w \neq v_r$, and l be the maximal length of the chains of $\mathcal{G}_{sbc}(D)$ such that they are contained in P. Return $l+1$. □

Induction step: For $v \in V_{cr} \cap V_{tr}$, if for each $w \in V_{cr}$ such that $(v, w) \in E_{scc}$, the computation of $h_{in}(w)$ has been finished, then

– $h_{out}(v) := \min\{h_{in}(w) | w \in V_{cr}, (v, w) \in E_{scc}\}$;
– if $v \neq s$: if $h(v) \geq h_{out}(v)+1$, then $h_{in}(v) := h_{out}(v)+1$, otherwise $h_{in}(v) := \infty$.

For each nontrivial strongly connected component C such that $C \neq C_t$ and $In(C) \cup Out(C) \subseteq V_{cr}$, if for each $v \in V_{cr} \cap Out(C)$ and each $w \in V_{cr}$ such that $(v, w) \in E_{scc}$, the computation of $h_{in}(w)$ has been finished, then

– for each $v \in V_{cr} \cap Out(C)$, $h_{out}(v) := \min\{h_{in}(w) | w \in V_{cr}, (v, w) \in E_{scc}\}$;
– for each $v \in V_{cr} \cap In(C)$, if $h(v) \geq \min\{MinNum(C, v, v')+h_{out}(v')+1 | v' \in Out(C)\}$, then $h_{in}(v) := \min\{MinNum(C, v, v')+h_{out}(v')+1 | v' \in Out(C)\}$, otherwise $h_{in}(v) := \infty$.

*Example 1 (***Case I***: C_s is trivial).* The digraph D is given in Fig.2.(a), C_s is trivial, and $\mathcal{D}_{scc}(D)$, the strongly-connected-component digraph of D, is given in Fig.2.(b). The critical vertices, V_{cr}, are those within the dashed cycle in Fig.2.(b), $V_{cr}^{in} = \{v_1, v_4, t, v_8, v_9\}$ and $V_{cr}^{out} = \{s, v_2, v_3, v_4, v_7, v_{11}\}$. The $h(v)$'s are given in Fig.2.(a) and pairs $(h_{in}(v), h_{out}(v))$ for $v \in V_{cr}$ are given in Fig.2.(b). Because $h_{out}(s) = 4$, the motion planning problem is feasible. Four holes can be moved to v_8, v_9, v_{11} and t before moving the robot, then the robot can be moved from s to v_8, and moved to v_9 by rotating around the cycle $v_8 v_9 v_{10}$, then to v_{11}, and finally to t. □

4.2 Case II: C_s is nontrivial and $C_s = C_t$

Let inside (outside) holes denote the holes in some $v \in V(C_s)$ $(v \notin V(C_s))$.

We first use the algorithm for feasibility of motion planning on strongly connected digraphs in [WG08] to decide whether the inside holes are sufficient to move the robot from s to t. If it is, then report "yes"; otherwise, the motion planning problem may still be feasible since the outside holes can be moved into C_s and used to move the robot from s to t.

For each outside hole, there may be multiple ports of C_s through which the hole can be moved into C_s, we should choose carefully those ports, otherwise, the feasibility may be destroyed, which has been illustrated in Fig.1. We introduce a greedy strategy to move the outside holes into C_s to avoid this.

Before presenting the greedy strategy, we recall a definition about the relative positions of the vertices in [WG08].

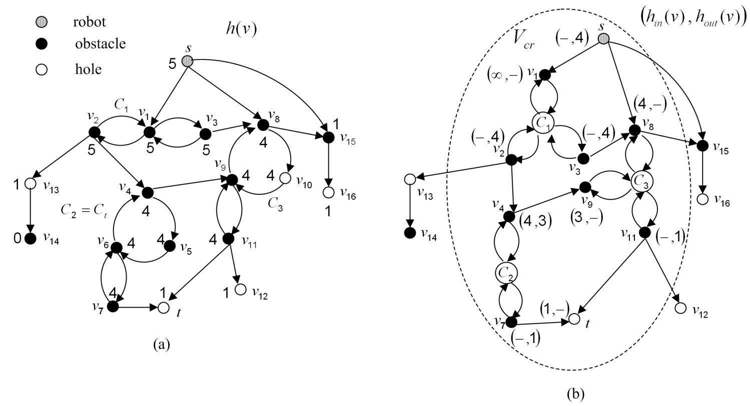

Fig. 2. Case I: C_s is trivial

Definition 3. *Let* $D = (V, E)$ *be a strongly connected digraph,* $\mathcal{G}_{sbc}(D) = (V_{sbc}, W_{sbc}, E_{sbc})$ *be the strongly-biconnected-component graph of* D*,* $u, v, w \in V$ *and* $v \neq w$*. Then* u *is said to be on the* w*-side of* v*, if* $u \neq v$ *and one of the following two conditions holds:*

1. $v \in W_{sbc}$ *(*v *is shared by at least two strongly biconnected components of* D*), and* u, w *are in the same connected component of* $\mathcal{G}(D - v)$*.*
2. $v \notin W_{sbc}$*, and either* u, w *are in the same connected component of* $\mathcal{G}(D - V(B))$*, or* $u \in V(B)$*, where* B *is the unique strongly biconnected component of* D *to which* v *belongs.*

Otherwise, u *is said to be not on the* w*-side of* v*. And* u *is said to be on the non-*w*-side of* v *if* u *is not on the* w*-side of* v *and* $u \neq v$*.*

In C_s, when we say that an inside hole is on the w-side of the robot, or on the non-w-side of the robot, and so on, we are talking about the positions of the inside hole and the robot.

An outside hole of C_s is said to be on the w-side (resp. non-w-side) of v if it can be moved into C_s through some port of C_s which is on the w-side (resp. non-w-side) of v. Note that an outside hole can be both on the w-side of v and on the non-w-side of v, since it can be moved into C_s both through some port on the w-side of v and through some other port on the non-w-side of v. An outside hole is said to be not on the w-side of v if it cannot be moved into C_s through some port on the w-side of v.

Let $h_{w-side}(v), h_{not-w-side}(v), h_{non-w-side}(v)$ denote respectively the number of (inside or outside) holes on the w-side of v, not on the w-side of v, and on the non-w-side of v.

Now we introduce the greedy strategy. The intuition of the strategy is that when it is necessary to move the robot away from t, first use the inside holes not on the t-side of the robot, then use the outside holes not on the t-side of the robot and farthest from t (the distance between an outside hole and t is the minimal distance between a port, through which the hole can be moved into C_s, and t).

If s, t do not belong to the same strongly biconnected component of C_s, then let $P := B_0 v_1 B_1 \cdots v_p B_p (p \geq 1)$ be a path in $\mathcal{G}_{sbc}(C_s) = (V_{sbc}, W_{sbc}, E_{sbc})$ such that $s \in V(B_0)$, $s \neq v_1$, $t \in V(B_p)$, $t \neq v_p$; otherwise, let $P := B_0$ and $p := 0$, where B_0 is the strongly biconnected component such that $s, t \in V(B_0)$.

We distinguish the following five cases,

1. $s \notin W_{sbc}$;
2. $s \in W_{sbc}$ and $h_{t-side}(s) \geq MinNum(C_s, s, t)$;
3. $s \in W_{sbc}$, $h_{t-side}(s) < MinNum(C_s, s, t)$ and $|V(B_0)| \geq 3$;
4. $s \in W_{sbc}$, $h_{t-side}(s) < MinNum(C_s, s, t)$, $|V(B_0)| = 2$ and s is a branching vertex;
5. $s \in W_{sbc}$, $h_{t-side}(s) < MinNum(C_s, s, t)$, $|V(B_0)| = 2$ and s is not a branching vertex.

In the following, we illustrate the greedy strategy by considering the **5th case**. The discussions of the other cases are similar and they are omitted due to space limitation.

Since $s \in W_{sbc}$ and s is not a branching vertex, there is a unique strongly biconnected component B such that $B \neq B_0$ and $s \in V(B)$.

If $t \in V(B_0)$, let $i_0 := 0$, otherwise, let
$i_0 := \min(\{p\} \cup \{i : |V(B_i)| \geq 3, \text{ or } v_i \text{ is a branching vertex}\})$.

We further distinguish the following four subcases,

Subcase 5.1. $h_{t-side}(s) \geq i_0 + 1$;
Subcase 5.2. $h_{t-side}(s) \leq i_0$ and $|V(B)| \geq 3$;
Subcase 5.3. $h_{t-side}(s) \leq i_0$, $|V(B)| = 2$ and B is a leaf of $\mathcal{G}_{sbc}(C_s)$;
Subcase 5.4. $h_{t-side}(s) \leq i_0$, $|V(B)| = 2$ and B is not a leaf of $\mathcal{G}_{sbc}(C_s)$.

Due to space limitation, we consider only Subcase 5.4. in the following.
Subcase 5.4. Because $h_{t-side}(s) \leq i_0$, it is necessary to move the robot away from t to move more holes to the t-side of the robot.

Let $v' \in V(B)$ such that $(s, v') \in E$, and $Q := B'_0 v'_1 B'_1 \ldots v'_q B'_q$ $(q \geq 1)$ be a path in $\mathcal{G}_{sbc}(C_s)$ such that

1. $B'_0 = B$;
2. either $|V(B'_q)| \geq 3$, or v'_q is a branching vertex, or B'_q is a leaf of $\mathcal{G}_{sbc}(C_s)$;
3. $\forall i : 1 \leq i < q, |V(B'_i)| = 2$, and v'_i is not a branching vertex.

Now we move the outside holes into C_s as follows:
Let $v'_0 = s$, then from $i = 1$ to $i = q$, do the following,

- If there are inside holes not on the t-side of v'_i, then move one such hole to v'_i, move the robot from v'_{i-1} to v'_i, and move the outside holes into C_s through v'_{i-1} as much as we can;

– If there are no inside holes not on the t-side of v'_i, but there are outside holes not on the t-side of v'_i, let k be the largest index such that there is at least one outside hole not on the t-side of v'_k, move one outside hole not on the t-side of v'_k into C_s, then to v'_i, move the robot from v'_{i-1} to v'_i, and move the outside holes into C_s through v'_{i-1} as much as we can.

Suppose the robot is in v'_r ($0 \leq r \leq q$) after the above loop.

If during the above loop, when the robot is in v'_i ($1 \leq i \leq r$), and the number of holes on the t-side of v'_i is $\geq i + i_0 + 1$, then: if $h(v'_i) \geq MinNum(C_s, v'_i, t)$, then report "yes", otherwise report "no".

Otherwise, there are two situations: $r < q$ or $r = q$.

In case of $r < q$: We must have $h_{not-t-side}(v'_{r+1}) = 0$. If $h_{t-side}(v'_r) > 0$ and $h_{not-t-side}(v'_r) > 0$, then move one hole on the t-side of v'_r to v'_{r-1}, move the robot from v'_r to v'_{r-1}, move one outside hole not on the t-side of v'_r into C_s through v'_r, then to v'_{r+1}, move the robot from v'_{r-1} to v'_{r+1}, now all the holes are on the t-side of v'_{r+1}, report "yes" iff $h(v'_{r+1}) \geq MinNum(C_s, v'_{r+1}, t)$. If $h_{t-side}(v'_r) = 0$, $h_{not-t-side}(v'_r) > 0$ and $h_{non-t-side}(v'_r) > 0$, then one outside hole not on the t-side of v'_r can be moved into C_s and to v'_{r+1} without moving the robot, move the robot to v'_{r+1}, now all the holes are on the t-side of v'_{r+1}, report "yes" iff $h(v'_{r+1}) \geq MinNum(C_s, v'_{r+1}, t)$.

In case of $r = q$: then there are the following three possibilities:

– $|V(B'_q)| \geq 3$;
– $|V(B'_q)| = 2$ and v'_q is a branching vertex;
– $|V(B'_q)| = 2$, v'_q is not a branching vertex, and B'_q is a leaf.

The first possibility above is reduced to Subcase 5.2., the second possibility above is reduced to Case 4., and the third possibility above is reduced to Subcase 5.3.

In all the other situations of Subcase 5.4., report "no".

Example 2. The motion planning problem is given in Fig.1.(a). To preserve the feasibility, we need first move the holes not on the t-side of v_2, i.e. the hole in v_1, to v_2, then move the robot to v_2 and move the outside holes into C_s through s as much as we can. So we move the two holes at v_7 and v_8 into C_s through s. Then all the holes are on the t-side of the robot, the problem is feasible iff the total number of holes is $\geq MinNum(C_s, v_2, t) = 3$. Thus the motion planning problem is feasible.

4.3 Case III: C_s is nontrivial and $C_s \neq C_t$

We can decide the feasibility in this case by combining the algorithms for Case I and Case II.

First, $h_{in}(v)$ and $h_{out}(v)$'s are computed just like in Case I. Then, holes outside C_s are moved into C_s by following the strategy similar to that in Case II, while preserving the enough holes outside C_s for moving the robot to t.

The details of the algorithm are omitted and they will appear in the full version of this paper.

5 Conclusions

In this paper, based on the work in [WG08] for motion planning on acyclic and strongly connected digraphs, we gave a complete solution to the feasibility of motion planning on digraphs. The most intricate part of this solution is to design a strategy to move the outside holes into C_s, the strongly connected component containing s, while not destroying the feasibility of the motion planning problem.

It would be interesting in the future to consider the optimization of motion planning on digraphs as well as the other variations of motion planning problem on digraphs, e.g. the reconfiguration problem that was considered on graphs in [KMS84].

References

[BJG00] J. Bang-Jensen and G. Gutin. *Digraphs: Theory, Algorithms and Applications, Springer Monographs in Mathematics*. Springer-Verlag, 2000.

[FK99] P. W. Finn and L. E. Kavmkit. Computational approaches to drug design. *Algorithmica*, 25:347–371, 1999.

[JJ06] Jorjeta G. Jetcheva and David B. Johnson. Routing characteristics of ad hoc networks with unidirectional links. *Ad Hoc Networks*, 4(3):303–325, 2006.

[KMS84] D. Kornhauser, G. Miller, and P. Spirakis. Coordinating pebble motion on graphs, the diameter of permutation groups, and applications. In *FOCS'84*, pages 241–250, 1984.

[LaV06] Steven M. LaValle. *Planning Algorithms*. Cambridge University Press, 2006.

[MD02] Mahesh K. Marina and Samir R. Das. Routing performance in the presence of unidirectional links in multihop wireless networks. In *in Proc. of ACM MobiHoc*, pages 12–23, 2002.

[MPG] Motion planning game. website. http://www.download-game.com/Motion_Planning_Game.htm.

[Per88] Yvonne Perrott. Track transportation systems. European patent, 1988. http://www.freepatentsonline.com/EP0284316.html.

[PRST94] C. H. Papadimitriou, P. Raghavan, M. Sudan, and H. Tamaki. Motion planning on a graph. In *FOCS'94*, pages 511–520, 1994.

[SA01] Guang Song and Nancy M. Amato. Using motion planning to study protein folding pathways. In *RECOMB '01: Proceedings of the fifth annual international conference on Computational biology*, pages 287–296, New York, NY, USA, 2001. ACM.

[Wes00] Douglas B. West. *Introduction to Graph Theory, second edition*. Prentice Hall, 2000.

[WG08] Zhilin Wu and Stéphane Grumbach. Feasibility of motion planning on acyclic and strongly connected directed graphs. *manuscript*, 2008.