# Motion Planning on a Graph

Christos H. Papadimitriou [*] Prabhakar Raghavan [†]
Madhu Sudan [†] Hisao Tamaki [‡]

August 8, 2010

## 1 Introduction

## 2 Preliminaries

In this paper, all graphs are simple, i.e., without self-loops or parallel edges, unless otherwise stated. A *weighted graph* $G$ is a triple $(V(G), E(G), \mathsf{cost}_G)$, where $V(G)$ is the vertex set of $G$, $E(G)$ is the edge set of $G$, and $\mathsf{cost}_G$ is a function from $E(G)$ to nonnegative reals, called the *cost function* of $G$. For each edge $e$ of $G$, we call $\mathsf{cost}_G(e)$ the *cost* of $e$. Let $G$ be a (weighted) graph. For $U \subseteq V(G)$, $G \setminus U$ will denote the subgraph of $G$ induced by $V(G) \setminus U$. When $G$ is weighted, we assume that the subgraph inherits the cost function of $G$ appropriately restricted. A *walk* in $G$ is a sequence of vertices of $G$ such that each pair of vertices consecutively appearing in the sequence are adjacent in $G$. If a walk starts from $u$ and ends with $v$, we call $u$ the *initial vertex* and $v$ the *final vertex* of the walk. We say that a walk visits a vertex $v$ if $v$ appears in the sequence. A walk is a *path* if no vertex is repeated in the sequence. Although a walk is primarily a sequence of vertices, we can also view it as a sequence of (oriented versions of) edges. In particular, the *length* of walk $W$, denoted by $|W|$, is defined to be the number of edges in the walk. For each walk $W$ of $G$, we extend the cost notation so that $\mathsf{cost}_G(W)$ denotes the *cost* of $W$, i.e., the sum of the costs of all the edges in $W$.

We start by formalizing the rules of our game. Let $G$ be a weighted graph. A *configuration* $\gamma$ on $G$ is a a pair $(O_\gamma, r_\gamma)$, where $O_\gamma$ is a subset of $V(G)$ and $r_\gamma$ is a vertex in $V(G) \setminus O_\gamma$. Informally, we interpret configuration $\gamma$ by regarding each vertex in $O_\gamma$ to have an *obstacle*, each vertex in $V(G) \setminus O_\gamma$ to have a *hole*, and vertex $r_\gamma$ to have the robot. Note that the definition of a configuration requires that the robot to coincide with a hole. We use the convention to denote $V(G) \setminus O_\gamma$, the set of vertices with holes, by $H_\gamma$.

[*]Department of Computer Science and Engineering, UCSD, La Jolla, CA 92093.
[†]IBM T.J. Watson Research Center, Yorktown Heights, NY 10598.
[‡]IBM Tokyo Research Laboratory, Kanagawa 242, Japan. Part of the work done at the IBM T.J. Watson Research Center.

A *robot move* on $G$ is an ordered pair of adjacent vertices $u$, $v$ of $G$, written as $u \Rightarrow v$. An *obstacle move* on $G$ is also an ordered pair of adjacent vertices, but written differently as $u \rightarrow v$. A robot move $u \Rightarrow v$ is *applicable* to configuration $\gamma$ if $r_\gamma = u$ and $v \in H_\gamma$. The *result of applying move* $u \Rightarrow v$ *to* $\gamma$, when applicable, is configuration $(O_\gamma, v)$. An obstacle move $u \rightarrow v$ is *applicable* to configuration $\gamma$ if $u \in O_\gamma$ and $v \in H_\gamma \setminus \{r_\gamma\}$. The *result of applying move* $u \rightarrow v$ *to* $\gamma$, when applicable, is configuration $((O_\gamma \setminus u) \cup \{v\}, r_\gamma)$. A *move* on $G$ is either a robot or an obstacle move on $G$. A move $u \rightarrow v$ or $u \Rightarrow v$ is said to be *from* $u$ *to* $v$. We say that a move *involves* vertex $v$ if the move is either to or from $v$. The *cost* of a move $m$ from $u$ to $v$, denoted by $\mathsf{cost}_G(m)$, is the cost $\mathsf{cost}_G(u, v)$ of the edge $(u, v)$.

A *plan* $\pi$ on $G$ is a possibly empty sequence of moves on $G$. We denote by $|\pi|$ the length of $\pi$, i.e., the number of moves of $\pi$, and by $\pi[i]$, $1 \leq i \leq |\pi|$, the $i$th move of $\pi$. We will use a similar notation for any sequence in general, without repeating the definition. Each $i$, $1 \leq i \leq |\pi|$, is called a *move index*, or simply an *index* of $\pi$; it is an *obstacle move index* if $\pi[i]$ is an obstacle move and a *robot move index* if $\pi[i]$ is a robot move. We denote by $I_\pi$ ($I_\pi^{ob}$, or $I_\pi^{rob}$, resp.) the set of move indices (obstacle move indices, or robot move indices, resp.) of $\pi$. The *cost* of a plan $\pi$ on $G$, denote by $\mathsf{cost}_G(\pi)$, is the sum of the costs of all the moves of $\pi$. When $G$ is clear from the context, we will often drop the subscript $G$ from notation $\mathsf{cost}_G$, whether it be the cost of an edge, path, move, plan, or the cost of other entities we are yet to define. We also denote by $\mathsf{ocost}_G(\pi)$ the sum of the costs of obstacle moves of $\pi$ and by $\mathsf{rcost}_G(\pi)$ the sum of the costs of robot moves of $\pi$. For two plans $\pi_1$ and $\pi_2$, we denote by $\pi_1 \parallel \pi_2$ the concatenation of $\pi_1$ with $\pi_2$, i.e., the sequence $\pi_1$ followed by $\pi_2$. We say that a plan $\pi$ is *applicable* to configuration $\gamma$ if the moves of $\pi$ are successively applicable to $\gamma$. More precisely, the definition is by induction as follows.

1. An empty plan is applicable to any configuration $\gamma$, and the result of the application is $\gamma$.
2. A plan $m \parallel \pi$, which consists of move $m$ followed by plan $\pi$, is applicable to $\gamma$ if $m$ is applicable to $\gamma$ and $\pi$ is applicable to $\gamma'$, where $\gamma'$ is the result of applying $\pi$ to $\gamma$. The result of applying $m \parallel \pi$ to $\gamma$ is defined to be the result of applying $\pi$ to $\gamma'$.

We say that two plans $\pi_1$ and $\pi_2$ are *equivalent on configuration* $\gamma$ if both of them are applicable to $\gamma$ and the results of the applications are identical.

An *instance* of problem GMP1R is a triple $(G, \gamma, t)$ where $G$ is a weighted graph, $\gamma$ is a configuration on $G$ and $t$ is a vertex of $G$. A plan $\pi$ is a *solution* to instance $(G, \gamma, t)$ if $\pi$ is applicable to $\gamma$ and $t = r_{\gamma'}$ where $\gamma'$ is the result of applying $\pi$ to $\gamma$, in other words, if plan $\pi$ applied to $\gamma$ brings the robot to $t$. An *optimal solution* is a solution with the minimum cost. Our goal is to find an optimal solution to the given instance.

It is often useful to view a group of obstacle moves without fixing the order among them. A *macro move* on $G$ is a multiset of obstacle moves on $G$. The *cost* of a macro move $M$ on $G$, $\mathsf{cost}_G(M)$, is the sum of the costs of moves in $M$. We denote by $|M|$ the *size* of $M$, i.e., the number of moves of $M$, counted according to multiplicity. One may think of a macro move on $G$ as a directed multigraph on $V(G)$: there is an edge from $u$ to $v$ for each move

$u \to v$ in the multiset. Let $M$ be a macro move on $G$. For each vertex $v$ of $G$, the *in-degree* of $v$ in $M$ is the number of moves in $M$ that are *to* $v$. The *out-degree* of $v$ in $M$ is the number of moves in $M$ that are *from* $v$. The *relative degree* of $v$ in $M$ is the out-degree of $v$ minus the in-degree of $v$. We call a macro move $M$ on $G$ is *well-formed* if the relative degree of each vertex of $G$ in $M$ is either 1, 0, or $-1$. A vertex of $G$ is a *source* of $M$ if its relative degree is 1; a *sink* if its relative degree is $-1$. We say a source-sink pair $(u, v)$ of $M$ is *connected* if there is a directed path from $u$ to $v$ when we view $M$ as a directed multigraph. We say macro moves $M_1$ and $M_2$ on $G$ are *equivalent* if the relative degree of each vertex of $G$ in $M_1$ is equal to its relative degree in $M_2$. In the directed multigraph view, $M_1$ is equivalent to $M_2$ if one can be obtained from the other by adding and/or deleting cycles. We say macro move $M_1$ is *reducible* to macro move $M_2$ if $M_2$ is a subset of $M_1$ (as a multiset) and is equivalent to $M_1$. The *reduced form* of macro move $M$ is the smallest subset of $M$ that is equivalent to $M$. We say a macro move $M$ is *irreducible* if its reduced form is $M$ itself.

For an arbitrary plan $\pi$ and each subset $I$ of $I^{ob}_{\pi}$,, we denote by $\mathcal{M}_{\pi} = \{\pi[i] \mid i \in I\}$ the macro move that is the multiset of all the obstacle moves of $\pi$ having their indices in $I$. In particular, we denote by $\mathcal{M}_{\pi} = \mathcal{M}_{\pi}[I^{ob}_{\pi}]$ the macro move consisting of all the obstacle moves of $\pi$. We call plan $\pi$ a *sequencing* of macro move $M$, if it consists only of obstacle moves and $M = \mathcal{M}_{\pi}$. We say that a macro move $M$ is applicable to configuration $\gamma$, if there is a sequencing of $M$ that is applicable to $\gamma$.

**Lemma 1** *Let $M$ be an irreducible macro move on $G$ and $\gamma$ be a configuration on $G$. Then, $M$ is applicable to $\gamma$ if and only if*
*(1) no move of $M$ involves $r_{\gamma}$,*
*(2) $M$ is well-formed, and*
*(3) each source of $M$ is in $O_{\gamma}$ and each sink of $M$ is in $H_{\gamma}$.*

**Proof:** That these conditions are necessary is obvious. The sufficiency of the conditions is proved by induction on the size of $M$. If $M$ is empty, it is clearly applicable to $\gamma$. Suppose $M$ is non-empty. Since $M$ is irreducible and well-formed, $M$ has at least one connected source-sink pair. By traversing from the source to the sink along the chain of moves of $M$, we can find a move $u \to v$ such that $u \in O_{\gamma}$ and $v \in H_{\gamma}$. Let $\gamma'$ be the result of applying $u \to v$ to $\gamma$ and let $M' = M \setminus u \to v$. Then, $M'$ and $\gamma'$ satisfies the above two conditions and therefore $M'$ is applicable to $\gamma'$ by the induction hypothesis. It follows that $M$ is applicable to $\gamma$, completing the induction. $\square$

As a special case, if a macro move $M$ is irreducible and has exactly one source-sink pair $(u, v)$, then the moves of $M$ constitute a path $P$ of $G$ from $u$ to $v$. In that case, we denote $M$ by $u \overset{P}{\to} v$. When the path $P$ is unique or clear from the context, we drop the it from the notation and write $u \to v$. In describing plans, we will allow inclusion of macro moves in the plan. This should be understood as a convention for the ease of description and a macro move is intended to denote a plan that is an appropriate sequencing of it.

Let $\gamma$ and $\delta$ be configurations on $G$ with $r_{\gamma} = r_{\delta}$ and $|O_{\gamma}| = |O_{\delta}|$. A *matching* $\mu$ from $\gamma$ to $\delta$ is a bijection from $O_{\gamma}$ to $O_{\delta}$. A matching $\mu$ from $\gamma$ to $\delta$ is *standard* if $\mu(v) = v$ for every $v \in O_\gamma \cap O_\delta$. The *cost* of matching $\mu$, $\mathsf{cost}(\mu)$, is the sum of $\mathsf{cost}(P_v)$ for $v \in O_\gamma$, where $P_v$ is a path from $v$ to $\mu$ in $G$ with the minimum cost.

**Lemma 2** *Let $\mu$ be a matching from configuration $\gamma$ to $\delta$ and suppose that there is a minimum cost path from each $v \in O_\gamma$ to $\mu(v)$ that does not contain $r_\gamma$. Then, there is a plan $\pi$ with $\mathsf{cost}(\pi) = \mathsf{cost}(\mu)$ such that the result of applying $\pi$ to $\gamma$ is $\delta$. Conversely, if there is a plan $\pi$ consisting of obstacle moves such that $\pi$ is applicable to $\gamma$ and results in $\delta$, then there is a matching $\mu$ from $\gamma$ to $\delta$ with $\mathsf{cost}(\mu) \leq \mathsf{cost}(\pi)$.*

**Proof:** For the first half, let $M$ be the union of macro moves $v \xrightarrow{P_v} \mu(v)$ for all $v \in O_\gamma$, where $P_v$ is the minimum cost path from $v$ to $\mu(v)$ which does not contain $r_\gamma$. Then, $M$ satisfies the conditions of Lemma 1 and therefore is applicable to $\gamma$. Take an appropriate sequencing of $M$ as $\pi$. For the second half, suppose $\pi$ is a plan applicable to $\gamma$ and consider the macro move $\mathcal{M}_\pi$ consisting of all the obstacle moves of $\pi$. The reduced form of $\mathcal{M}_\pi$ can be decomposed into paths, each connecting a source-sink pair of $\mathcal{M}_\pi$. Fix such a decomposition and define $\mu$ by putting $\mu(u) = v$ for each source-sink pair $(u, v)$ in the decomposition and $\mu(u) = u$ for each $u \in O_\gamma$ that is not a source of $\mathcal{M}_\pi$. □

Let $i \in I_\pi^{ob}$ be an obstacle move index of $\pi$. Being slightly sloppy, we say $i$ is *from $u$* (*to $v$*, resp.) if move $\pi[i]$ is from $u$ (to $v$, resp.). We also say that $i$ *involves* $v$ if $\pi[i]$ involves $v$. Let $i, j \in I_\pi^{ob}$ be two obstacle move indices of $\pi$ and $v$ a vertex of $G$. We say $i$ *connects to $j$ at $v$* in $\pi$, if $i$ is to $v$ and $j$ is from $v$. We say $i$ *connects to* $j$ in $\pi$, if $i$ connects to $j$ at some vertex $v$. We say $i$ *obstacle-wise connects to* $j$ in $\pi$, if $j$ is the smallest move index larger than $i$ such that $i$ connects to $j$. When a plan is played out on a board with pebbles representing obstacles, obstacle-wise connections trace the motion of individual pebbles. We call $i \in I_\pi^{ob}$ *starting* if there is no $j$ that obstacle-wise connects to $i$. We call $i \in I_\pi^{ob}$ *ending* if there is no $j$ to which $i$ obstacle-wise connects. A starting (ending, resp.) obstacle move index corresponds to the first (last, resp.) move applied to a pebble representing an obstacle. We say $i$ *hole-wise connects to* $j$ in $\pi$, if $j$ is the largest move index smaller than $i$ such that $i$ connects to $j$. To make an analogous interpretation, we need to represent the holes, rather than obstacles, by pebbles: hole-wise connections *back trace* the motion of individual pebbles. A *thread* $\theta$ in plan $\pi$ is a nonempty sequence of move indices of $\pi$ such that $\theta[k]$ connects to $\theta[k+1]$ for $1 \leq k < |\theta|$. Each thread $\theta$ in plan $\pi$ naturally corresponds to a walk on $G$, which we denote by $\mathsf{walk}(\theta)$: the $k$th edge of $\mathsf{walk}(\theta)$ is from $u$ to $v$ if $\pi[\theta[k]]$ is the move from $u$ to $v$. We denote by $\mathsf{init}(\theta)$ the initial vertex of $\mathsf{walk}(\theta)$ and by $\mathsf{fin}(\theta)$ the final vertex of $\mathsf{walk}(\theta)$. Thread $\theta$ is *obstacle-wise* (*hole-wise*, resp.) if $\theta[k]$ obstacle-wise (hole-wise, resp.) connects to $\theta[k+1]$ for $1 \leq k < |\theta|$.

We call an obstacle-wise thread in $\pi$ *maximal* if its first element is a starting index of $\pi$ and its last element is an ending index of $\pi$. We call a thread an *MOT-chain* if it is a concatenation of one or more maximal obstacle-wise threads. An MOT-chain $\theta$ is called an MOT-cycle if $\mathsf{init}(\theta) = \mathsf{fin}(\theta)$; we regard two MOT-cycles as identical if they have the same set of maximal obstacle-wise threads and only differ in the initial vertex to start the thread representation. An MOT-chain $\theta$ is *acyclic* if it is not an MOT-cycle. We call an

MOT-chain $\theta$ of $\pi$ *maximal* if either $\theta$ is a MOT-cycle or it is acyclic and there is no maximal obstacle thread $\theta'$ of $\pi$ such that $\mathsf{fin}(\theta') = \mathsf{init}(\theta)$ or $\mathsf{init}(\theta') = \mathsf{fin}(theta)$.

The following observations are obvious and will be used freely.

1. Any two distinct maximal obstacle-wise threads of $\pi$ are disjoint. Any two distinct maximal hole-wise threads of $\pi$ are disjoint. Any two distinct maximal MOT-chains are disjoint.

2. $I_\pi^{ob}$ is the union of the maximal obstacle-wise threads of $\pi$. It is the union of the maximal hole-wise threads of $\pi$. It is also the union of the maximal MOT-chains of $\pi$.

3. Let $\gamma$ be a configuration to which $\pi$ is applicable and let $\delta$ be the result of applying $\pi$ to $\gamma$. Then, for any maximal obstacle-wise thread $\theta$, $\mathsf{init}(\theta) \in O_\gamma$ and $\mathsf{fin}(\theta) \in O_\delta$. For any maximal hole-wise trace $\tau$, $\mathsf{init}(\tau) \in H_\delta$ and $\mathsf{fin}(\tau) \in H_\gamma$. Finally, for any acyclic maximal MOT-chain $\sigma$, $\mathsf{init}(\sigma) \in O_\gamma \setminus O_\delta$ and $\mathsf{fin}(\sigma) \in O_\delta \setminus O_\gamma$.

Finally, the *robot's walk* in plan $\pi$ on $G$ is the walk in $G$ defined by the sequence of robot moves in $\pi$ in an obvious manner.

## 3 Canonical plans for a tree

¿From now on, we assume that the graph $G$ in the problem instance is a tree. The goal of this section is to define a class of plans, which we call *canonical plans*, and show that an optimal solution to any problem instance may be found in this class of plans.

Let $u, v, w$ be vertices of $G$ with $u \neq v$ and $w \neq v$. We say that $w$ is in the *$u$-side of $v$* (in $G$) if $w$ is in the connected component of $G \setminus \{v\}$ that contains $u$. We call a plan *monotonic* if the robot's walk in the plan is a path.

**Lemma 3** *Let $\pi$ be a monotonic plan on tree $G$ applicable to configuration $\gamma$. Then there is a plan $\pi'$ equivalent to $\pi$ on $\gamma$ with $\mathsf{cost}(\pi') = \mathsf{cost}(\pi)$, in which all the robot moves appear consecutively.*

**Proof:** Let the robot's walk in $\pi$ be a path $P$ from $s$ to $t$. Let $r_i$ denote the location of the robot after the first $i$ moves of $\pi$ are applied to $\gamma$, $0 \leq i \leq |\pi|$. We say that an obstacle move index $i \in I_\pi^{ob}$ is *behind* (*ahead of*, resp.) the robot if the vertices involved in move $\pi[i]$ are both in the $u$-side ($v$-side, resp.) of $r_i$. Permute $\pi$ by first taking the obstacle moves with indices ahead of the robot, then the robot moves, and finally the obstacle moves with indices behind the robot, otherwise preserving the order in $\pi$; let the result be $\pi'$. Let $\pi(v)$ ($\pi'(v)$, resp.) denote the sequence of moves of $\pi$ ($\pi'$, resp.) that involves $v$, in the order of appearance in $\pi$ ($\pi'$, resp.). It is easily verified that $\pi(v) = \pi'(v)$ for every vertex $v$ of $G$. A straightforward induction based on this property shows that $\pi$ and $\pi'$ are equivalent on any configuration to which $\pi$ is applicable. $\square$

It would be nice if we could confine ourselves to monotonic plans in our search for an optimal solution for a given instance. This is unfortunately not the case; we need to define a few ways of deviating from a monotonic motion of the robot.

Let $P$ be a path of $G$ from $s$ to $t$. We call a vertex $u$ of $G$ a *sidestep vertex* of $P$ if $u \notin V(P)$ and $u$ is adjacent to an internal vertex of $P$, i.e., to a vertex in $V(P) \setminus \{s, t\}$. Suppose $P$ is of the form $s = v_0, v_1, \ldots, v_k = t$. A sequence of robot moves is
(1) an *advance* along $P$, if its a single move $v_{i-1} \Rightarrow v_i$ for some $i$, $1 \leq i \leq k$,
(2) a *sidestep at* $v_i$ along $P$, if it is a walk of the form $v_i \Rightarrow u \Rightarrow v_i$, where $1 \leq i \leq k-1$ and $u$ is a sidestep vertex of $P$ adjacent to $v_i$, or
(3) a *wiggle at* $v_i$ along $P$, if it is a walk of the form $v_i \Rightarrow v_{i+1} \Rightarrow v_i \Rightarrow v_{i-1} \Rightarrow v_i$, where $1 \leq i \leq k-1$.
We call a walk from $s$ to $t$ *quasi-monotonic* along $P$ if it is a concatenation of advances, sidesteps, and wiggles along $P$, such that at most one sidestep or wiggle occurs at each $v_i$, $1 \leq i \leq k-1$. We call a walk $W$ from $s$ to $t$ in $G$ *quasi-bitonic*, if there is some path $P$ of $G$ from $s'$ to $t$ such that $s$ is on $P$ (with $s'$ possibly being equal to $s$) and $W$ is a concatenation of the path from some $s$ to $s'$ and a quasi-monotonic walk from $s'$ to $t$. We call a plan *quai-monotonic* (*quasi-bitonic*, resp.) if the robot's walk in the plan is (quasi-monotonic, quasi-bitonic, resp.). Our first goal is to show that for any problem instance of GMP1R, where the graph is a tree, there is an optimal solution to the instance that is quasi-bitonic.

## 3.1 Path-plans

We start our analysis with a seemingly simple case, where the robot stays on a path. Let $s, t \in V(G)$ and $P$ a path from $s$ to $t$. Call a walk $W$ from $s$ to $t$ *path-shaped* on $P$ if (1) all the vertices visited by $W$ is on $P$ and (2) $W$ does not visit $t$ until the final step. Note that a path-shaped walk may contain any number of "turns" and thus may not be a path. We call a plan on $G$ a *path-plan* if the robot's walk in the plan is path-shaped. We want to show that any path-plan can be transformed into an equivalent quasi-monotonic plan without increasing the cost.

Let $\pi$ be a path-plan on tree $G$, in which the robot's walk is path-shaped on path $P$ from $s$ to $t$. Let $u$ be a sidestep vertex of $P$, $v$ the vertex of $P$ adjacent to $u$, $P_1$ the path from $s$ to $u$, and $P_2$ the path from $v$ to $t$. We say that $\pi$ is *decomposable at* $u$, if,
(1) $v$ is not adjacent to $t$, and
(2) for each configuration $\gamma$ to which $\pi$ is applicable, $\pi$ is equivalent to $\pi_1 \parallel u \Rightarrow v \parallel \pi_2$ on $\gamma$, where $\pi_1$ and $\pi_2$ are path-plans on $P_1$ and $P_2$ respectively, such that $\mathsf{cost}(\pi_1) + \mathsf{cost}(u \Rightarrow v) + \mathsf{cost}(p_2) \leq \mathsf{cost}(\pi)$.

**Lemma 4** *Let $\pi$ be a path-plan on tree $G$, in which the robot's walk is from $s$ to $t$. Let* $\mathrm{prev}(t)$ *denote the vertex adjacent to $t$ in the $s$-side of $t$. Then, for every configuration $\gamma$ to which $\pi$ is applicable, there is a quasi-monotonic plan $\rho$ equivalent to $\pi$ on $\gamma$, with* $\mathsf{cost}(\rho) \leq \mathsf{cost}(\pi)$, *in which no sidestep of wiggle occurs at* $\mathrm{prev}(t)$.

**Proof:** Let $P$ denote the path from $s$ to $t$. The proof is by induction on the length of $P$. If $|P| \leq 1$ then the result is obvious. Suppose $|P| = 2$ and $P = (s, v, t)$. The only way in which $\pi$ can avoid being monotonic is to "oscillate" the robot between $s$ and $v$, because a path-plan does not allow the robot to visit $t$ more than once. With a simple reordering of the obstacle moves of $\pi$, as in the proof of Lemma 3, we can remove that "oscillation" from the robot's walk. Now suppose $|P| \geq 3$. If $\pi$ is decomposable at some sidestep vertex of $P$, the result is easily obtained by applying the induction hypothesis to the two path-plans in the decomposition. In the following, we deal with the case where $|P| \geq 3$ and $\pi$ is not decomposable.

Throughout the proof, we fix the initial configuration $\gamma$ to which $\pi$ is applied. Let $\gamma_i$, $0 \leq i \leq |\pi|$, be the result of applying the first $i$ moves of $\pi$ to $\gamma$. Let $\delta = \gamma_{|\pi|}$ denote the result of applying $\pi$ to $\gamma$.

For each $v \in V(P) \setminus \{s\}$, let $\mathrm{next}(v)$ denote the vertex immediately following $v$ on $P$ and, for each $v \in V(P) \setminus \{t\}$, let $\mathrm{prev}(v)$ denote the vertex immediately preceding $v$ on $P$. For each vertex $u$ adjacent to some vertex of $P$, (1) let $b_u$ denote the vertex on $P$ adjacent to $u$, and (2) let $T_u$ denote the subtree of $G$ consisting of the vertices in the $u$-side of $b_u$. Let $U$ denote the set of sidestep vertices of $P$. Let $W$ denote the forest consisting of trees $T_u$ for all $u \in U$. Let $S$ denote the forest consisting of trees $T_u$ for all $u$ in the $s$-side of $\mathrm{next}(s)$; let $T$ denote the forest consisting of trees $T_u$ for all $u$ in the $t$-side of $\mathrm{prev}(t)$. Note that the vertex sets $V(P)$, $V(S)$, $V(T)$, and $V(W)$ partition $V(G)$.

Let $i \in I_\pi^{ob}$ be an obstacle move index of $\pi$. We say $i$ is *ahead of the robot* (*behind the robot*, resp.) if the vertices involved in move $\pi[i]$ are in the $t$-side ($s$-side, resp.) of $r_{\gamma_i}$, i.e., the robot's location at the $i$th step in the application of $\pi$. Let $u \in U$ be a sidestep vertex. An *obstacle-critical pair* of $\pi$ at $u$ is a pair $(i, j)$ of obstacle move indices of $\pi$ such that $i$ is from $b_u$ to $u$ and is behind the robot, $j$ is from $u$ to $b_u$ and is ahead of the robot, and there is an obstacle-wise thread starting with $i$ and ending with $j$. Similarly, a *hole-critical pair* of $\pi$ at $u$ is a pair $(i, j)$ of move indices of $\pi$ such that $i$ is from $u$ to $b_u$ and is behind the robot, $j$ is from $b_u$ to $u$ and is ahead of the robot, and there is an hole-wise thread starting with $j$ and ending with $i$. We call a pair *critical* if it is obstacle- or hole-critical. Note that it must be $i < j$ for any critical pair $(i, j)$.

Observe that, if $u$ is a sidestep vertex associated with a hole- or obstacle-critical pair, then each of the backward robot moves $\mathrm{next}(b_u) \Rightarrow b_u$ and $b_u \Rightarrow \mathrm{prev}(b_u)$ must appear in $\pi$ at least once. In particular, since the definition of a path-walk forbids the move $t \Rightarrow \mathrm{prev}(t)$, this implies that $u$ is such that $b_u \neq \mathrm{prev}(t)$. Define $U' \subseteq U$ by $U' = \{u \in U \mid b_u \neq \mathrm{prev}(t)\}$; it is the set of sidestep vertices that are possibly associated with critical pairs.

The proofs of the following three claims will be given later.

**Claim 1** *If there is an obstacle-critical pair of $\pi$ at some sidestep vertex $u$, then $\pi$ is decomposable at $u$.*

**Claim 2** *Suppose there are two distinct hole-critical pairs of $\pi$, one at $u$ and the other at $u'$ such that $b_u = b'_u$ (with $u$ and $u'$ possibly being identical). Then $\pi$ is decomposable at $u$ or $u'$.*

**Claim 3** *Suppose there are two distinct hole-critical pairs of $\pi$, at $u$ and at $u'$. If $b_{u'} = \mathrm{next}(b_u)$ and furthermore the backward robot move $b_{u'} \Rightarrow b_u$ appears at most once in $\pi$, then $\pi$ is decomposable at $u$ or $u'$.*

The following claim is used to simplify our problem.

**Claim 4** *The given plan $\pi$ is equivalent to $\pi_1 \parallel \pi' \parallel \pi_2$ for some $\pi_1$, $\pi_2$, and $\pi'$ such that $\pi_1$ and $\pi_2$ consist only of obstacle moves, the macro move $\mathcal{M}_{\pi'}$ (i.e., the multiset of all the obstacle moves of $\pi'$) does not have any sink in $T$, and $\mathsf{cost}(\pi_1) + \mathsf{cost}(\pi') + \mathsf{cost}(\pi_2) \leq \mathsf{cost}(\pi)$.*

**Proof:** Let $\Sigma$ denote the set of acyclic maximal MOT-chains of $\pi$ whose final vertices are in $T$. Let $\Sigma' \subseteq \Sigma$ be the set of acyclic maximal MOT-chains belonging to $\Sigma$ whose initial vertices are in $S$. Let $\sigma$ be a maximal MOT-chain in $\Sigma'$. Considering how the MOT-chain interacts with the robot's move, there must be some sidestep vertex $u \in U$ and a pair $(i, j)$ of move indices appearing in $\sigma$, $i$ appearing in $\sigma$ before $j$, such that $i$ is from $b_u$ to $u$ and behind the robot, $j$ is from $u$ to $b_u$ and ahead of the robot, and for any $i'$ appearing in $\sigma$ between $i$ and $j$, $\pi[i']$ does not involve $b_u$. If $i$ and $j$ belong to the same obstacle-wise thread in $\sigma$, then by Claim 1 $\pi$ is decomposable, contradicting our assumption. Therefore, $i$ and $j$ must belong to distinct obstacle-wise threads in $\sigma$. It follows that we can express $\sigma$ as a concatenation of two MOT-chains $\mathsf{left}(\sigma)$ and $\mathsf{right}(\sigma)$ such that $\mathsf{fin}(\mathsf{left}(\sigma)) = \mathsf{init}(\mathsf{right}(\sigma))$ is a vertex in $T_u$. Let $\Sigma_1 = (\Sigma \setminus \Sigma') \cup \{\mathsf{right}(\sigma) \mid \sigma \in \Sigma'\}$ and $\Sigma_2 = \{\mathsf{left}(\sigma) \mid \sigma \in \Sigma'\}$. We set $\pi_1$ and $\pi_2$ to be appropriate sequencings of macro moves $\bigcup_{\sigma \in \Sigma_1} \mathsf{init}(\sigma) \to \mathsf{fin}(\sigma)$ and $\bigcup_{\sigma \in \Sigma_2} \mathsf{init}(\sigma) \to \mathsf{fin}(\sigma)$, respectively, and set $\pi'$ to be the plan obtained from $\pi$ by removing $\pi[i]$ for every $i$ that appears in some MOT-chain belonging to $\Sigma$. It is easily verified that $\pi_1 \parallel \pi' \parallel \pi_2$ is equivalent to $\pi$ on $\gamma$. The cost condition is also obviously satisfied, because the multiset of obstacle moves of $\pi_1 \parallel \pi' \parallel \pi_2$ is a subset of that of $\pi$. $\Box$

In the following, we assume, without loss of generality owing to the above claim, that $\mathcal{M}_\pi$ does not have any sink in $T$. For each vertex $v$ of $P \cup T$ that has an obstacle in $\gamma$, i.e., $v \in V(P \cup T) \cap O_\gamma$, define a thread $\mathsf{evac}(v)$ of $\pi$, called the *evacuation thread* for $v$, as follows. We need auxiliary definitions. For move indices $i, j \in I_\pi^{ob}$, we say that $i$ *firmly connects to* $j$ if either
(1) $i$ obstacle-wise connects to $j$ at some vertex in $P \cup T$, or
(2) $i$ connects to $j$ at some vertex in $T$, $i$ is an ending index (i.e., does not obstacle-wise connect to any index), and $j$ is a starting index (i.e., no index obstacle-wise connects to $j$).
Note that at most one $i$ firmly connects to each $j$. For each $i \in I_\pi^{ob}$ such that $i$ involves a vertex in $P \cup T$, define a thread $\mathsf{othread}(i)$ inductively as follows: $\mathsf{othread}(i)$ is singleton $i$ if $i$ does not firmly connect to any index; $\mathsf{othread}(i)$ is $i$ followed by $\mathsf{othread}(j)$ if $i$ firmly connects to $j$. Note the the last element of $\mathsf{othread}(i)$ is to a vertex in $P$, a sidestep vertex of $P$, or a vertex adjacent to $s$.

Now, $\mathsf{evac}(v)$ consists of one or more *segments* possibly followed by a *coda*. Each segment is thread $\mathsf{othread}(i)$ for some move index $i$ and the coda is either empty or a hole-wise thread. The first segment of $\mathsf{evac}(v)$ is $\mathsf{othread}(i_v)$ where $i_v$ is the move index that is starting and is from $v$. Suppose the first $k$ segments of $\mathsf{evac}(v)$ has been defined for $k \geq 1$. Let $j$ be the last element of this $k$th segment. If $j$ is to a vertex on $P$ then $\mathsf{evac}(v)$ is already completed: it has $k$ segments and no coda. Otherwise, $j$ must be to a vertex $u$ that is either a sidestep vertex or a vertex adjacent to $s$ and not on $P$. Let $\tau_k$ be the longest hole-wise thread that starts with $j$. If no move index of $\tau_k$ involves $b_u$ (i.e., $\mathsf{walk}(\tau_k)$ stays within $T_u$), then we let $\mathsf{evac}(v)$ end with $\tau_k$ as the coda. Otherwise, let $j'$ be the first element of $\tau_k$ such that $j'$ is from $u$ to $b_u$. We take $\mathsf{othread}(j')$ as the $(k+1)$st segment of $\mathsf{evac}(v)$ and continue the inductive definition.

Note that (1) $\mathsf{evac}(v)$ is a finite sequence for every $v \in V(P \cup T) \cap O_\gamma$ and (2) $\mathsf{evac}(v)$ and $\mathsf{evac}(v')$ are disjoint for every distinct pair $v, v' \in V(P \cup T) \cap O_\gamma$. This follows easily from the fact that for each $i \in I_\pi$ there is at most one $i'$ (and there is none if $i$ is a starting move index from some $v \in V(P \cup T) \cap O_\gamma$) that can immediately precede $i$ in any evacuation thread.

In each segment of $\mathsf{evac}(v)$, all the move indices are either entirely ahead of the robot or entirely behind the robot. We call a segment in the first case an A-segment and in the second case a B-segment. Call an evacuation thread *critical* if its last move is to a vertex in $P \cup S$. Note that the last segment of a critical evacuation thread is a B-segment. Since any critical evacuation thread starts with an A-segment, it must contain an A-segment $\tau$ immediately followed by a B-segment $\tau'$. Let $i$ be the last element of $\tau$ and $i'$ the first element of $\tau'$. Then $(i', i)$ is a hole-critical pair.

**Claim 5** *Let* $\mathsf{evac}(v)$ *be an evacuation thread, let* $\tau$ *be an arbitrary A-segment of* $\mathsf{evac}(v)$ *and suppose a hole-critical pair* $(i, j)$ *at sidestep vertex* $u$ *appears in* $\mathsf{evac}(v)$*. Suppose furthermore that* $j$ *is either the last element of* $\tau$ *or appears after* $\tau$ *in* $\mathsf{evac}(v)$*. Suppose that* $\mathsf{walk}(\tau)$ *does not visit any vertex in the* $\mathrm{next}(\mathrm{next}(b_u))$*-side of* $\mathrm{next}(b_u)$*. Let* $l = |\tau|$*. Then,* $\tau[l] > i$ *implies that* $\tau$ *is not the first segment of* $\mathsf{evac}(v)$ *and that* $\tau[1] > i$*.*

**Proof:** Before anything, note that $\mathrm{next}(\mathrm{next}(b_u))$ is well defined because $u$ must be in $U'$. The proof is by contradiction. Suppose first that $\tau[l] > i$ and $\tau[1] < i$. By the assumption that $\mathsf{walk}(\tau)$ does not visit any vertex in the $\mathrm{next}(\mathrm{next}(b_u))$-side of $\mathrm{next}(b_u)$, move index $\tau[1]$ must be to a vertex in the $\mathrm{next}(b_u)$-side of $\mathrm{next}(\mathrm{next}(b_u))$. Then the following sequence of events happen in the application of $\pi$. Move $\pi[\tau[1]]$ puts an obstacle on $P$ in the $t$-side of the robot, which stays on $P \cup T$ until move $\pi[\tau[l]]$ takes if off $P \cup T$. Between these two moves, a robot move $\pi[i']$, for some $\tau[1] < i' < i$, moves the robot to $\mathrm{next}(b_u)$. But this is impossible because $\mathsf{walk}(\tau)$ stays in the $\mathrm{next}(b_u)$-side of $\mathrm{next}(\mathrm{next}(b_u))$. We get a similar contradiction if we suppose $\tau[l] > i$ and $\tau$ is the first segment of $\mathsf{evac}(v)$. □

The following claim is almost symmetric to the above and its proof is similar.

**Claim 6** *Let* $\mathsf{evac}(v)$ *be an evacuation thread that contains a hole-critical pair* $(i, j)$ *at some sidestep vertex* $u$*. Let* $\tau$ *be an arbitrary B-segment of* $\mathsf{evac}(v)$ *such that* $\tau$ *appears in* $\mathsf{evac}(v)$ *after* $i$*, possibly* $\tau[1]$ *being equal to* $i$*. Suppose that either (1)* $\mathrm{prev}(b_u) = s$ *or (2)* $\mathrm{prev}(b_u) \neq s$ *and* $\mathsf{walk}(\tau)$ *does not visit any vertex in the* $\mathrm{prev}(\mathrm{prev}(b_u))$*-side of* $\mathrm{prev}(b_u)$*. Let* $l = |\tau|$*. Then,* $\tau[1] < j$ *implies that* $\tau$ *is not the last segment of* $\mathsf{evac}(v)$ *and that* $\tau[l] < j$*.*

Suppose a hole-critical pair $(i, j)$ at $u$ appears in the evacuation sequence $\mathsf{evac}(v)$. We call the pair $(i, j)$ *forward-safe* if the part of the thread $\mathsf{evac}(v)$ before $i$ visits $\mathrm{next}(next(b_u))$. We call it *backward-safe* if either
(1) the final vertex of $\mathsf{evac}(v)$ is in $S$, or
(2) $b_u \neq \mathrm{prev}(s)$ and the part of the thread $\mathsf{evac}(v)$ after $j$ visits $\mathrm{prev}(\mathrm{prev}(b_u))$.

The following is a key claim in constructing our plan $\delta$.

**Claim 7** *Let $v \in O_\gamma$ be a vertex in $P \cup T$ and suppose the evacuation tread $\mathsf{evac}(v)$ is critical. Then, $\mathsf{evac}(v)$ contains a hole-critical pair that is backward- and forward-safe at the same time.*

**Proof:** We have already noted that any critical evacuation sequence contains a hole-critical pair. We first show that at least hole-critical pair must be backward-safe, namely the last one. Let $(i_1, j_1)$ be the last hole-critical pair and suppose it is not backward-safe. Let $\tau_1, \tau_2, \ldots,$ be the segments appearing after $i_1$ in $\mathsf{evac}(v)$, with $\tau_1$ starting with $i_1$. Since $(i_1, j_1)$ is the last hole-critical pair and $\mathsf{evac}(v)$ ends in $P \cup S$, these segments $\tau_1, \ldots$ are all $B$-segments. Let $i'_1$ be the last element of $\tau_1$. By Claim 6, $\tau_1$ is not the last segment of $\mathsf{evac}(v)$ and $i'_1 < j_1$. Let $i_2$ be the first element of $\tau_2$. Since there is a hole-wise thread starting with $i'_1$ and ending with $i_2$, we have $i_2 < i'_1 < j_1$. Proceeding by induction, we have a contradicting conclusion that $\tau_k$ is not the last segment for every $k$. Therefore, $(i_1, j_1)$ must be backward-safe.

Now, let $(i_0, j_0)$ be the first hole-critical pair in $\mathsf{evac}(v)$ that is backward-safe. We claim that it must also be forward-safe. Suppose to the contrary that it is not forward-safe. We first show by induction that, for every hole-critical pair $(i, j)$ preceding $(i_0, j_0)$ in $\mathsf{evac}(v)$ we have $i_0 < j$. The base case $(i, j) = (i_0, j_0)$ is trivial since $i < j$ for any critical pair. Suppose the claim holds for a hole-critical pair $(i, j)$ that precedes $(i_0, j_0)$ and let $(i', j')$ be the hole-critical pair that immediately precedes $(i, j)$. Then, between $j'$ and $i$ in $\mathsf{evac}(v)$, there is a consecutive series of B-segments and a consecutive series of A-segments. Let $\tau$ be the last of those B-segments and $\tau'$ be the first of those A-segments. Repeatedly applying Claim 5 to those A-segments based on the assumption that $(i_0, j_0)$ is not forward-safe, we obtain $i_0 < \tau'[1]$. On the other hand, repeatedly applying Claim 6 to those B-segments based on the assumption that $(i', j')$ is not backward-safe, we have $\tau[l] < j'$, where $l = |\tau|$. But clearly $\tau'[1] < \tau[l]$ because there is a hole-wise thread starting with $\tau[l]$ and ending with $\tau'[1]$. Combining all of these inequalities, we obtain $i_0 < j'$, completing the induction. Now, Claim 5 applied to the A-segments before the first critical pair implies, under our assumption that $(i_0, j_0)$ is not forward-safe, that none of those A-segments can be the first segment of $\mathsf{evac}(v)$, a contradiction. Therefore, the pair $(i_0, j_0)$ must be forward-safe. □

Let $V_o = V(P \cup T) \cap O_\gamma$ be the set of vertices in $P \cup T$ that have an obstacle initially. Let $I_{evac}$ be the subset of $I^{ob}_\pi$ defined by $I_{evac} = \bigcup_{v \in V_o} \mathsf{evac}(v)$, where we regard thread $\mathsf{evac}(v)$ as a set of move indices. Let $I_{rem}$ be the set of remaining move indices: $I_{rem} = I^{ob}_\pi \setminus I_{evac}$. Clearly, the sources of macro move $\mathcal{M}_\pi[I_{evac}]$ contain all the sources of macro move $\mathcal{M}_\pi$ in $T$. Moreover, from the definition of the evacuation thread, macro move $\mathcal{M}_\pi[I_{evac}]$ does not have a sink in $T$. Combined with the assumption that $\mathcal{M}_\pi$ does not have a sink in $T$, it follows that the macro move $\mathcal{M}_\pi[I_{rem}]$ has neither a source nor a sink in $T$. This means, in informal terms, that we may postpone the application of the moves in $I_{rem}$ until the robot reaches its destination $t$.

Let $V_c$ be the subset of $V_o$ consisting of all the vertices $v$ such that $\mathsf{evac}(v)$ is critical. For each $v \in V_c$, $\mathsf{evac}(v)$ has a backward- and forward-safe hole-critical pair, by Claim 7; fix one such hole-critical pair and let $u_v$ denote the sidestep vertex associated with it. Let $V'_c$ denote the subset of $V_c$ consisting of all the vertices $v$ such that $b_{u_v} = \mathrm{next}(s)$. For each $v \in V_c \setminus V'_c$, decompose $\mathsf{evac}(v)$ into four parts as $\mathsf{evac}(v) = \mathsf{evac}_1(v) \parallel \mathsf{evac}_2(v) \parallel \mathsf{evac}_3(v) \parallel \mathsf{evac}_4(v)$, so that $\mathsf{fin}(\mathsf{evac}_1(v)) = \mathrm{next}(\mathrm{next}(b_{u_v}))$, $\mathsf{fin}(\mathsf{evac}_2(v)) = u_v$, and $\mathsf{fin}(\mathsf{evac}_3(v)) = \mathrm{prev}(\mathrm{prev}(b_{u_v}))$. Similarly for each $v \in V'_c$, decompose $\mathsf{evac}(v)$ into three parts as $\mathsf{evac}(v) = \mathsf{evac}_1(v) \parallel \mathsf{evac}_2(v) \parallel \mathsf{evac}_3(v)$, so that $\mathsf{fin}(\mathsf{evac}_1(v)) = \mathrm{next}(\mathrm{next}(b_{u_v}))$, $\mathsf{fin}(\mathsf{evac}_2(v)) = u_v$. Note that $\mathsf{fin}(\mathsf{evac}_3(v)) = \mathsf{fin}(\mathsf{evac}(v))$ is in $S$ in this case, by the definition of backward-safety.

Our plan $\rho$ will consist of several parts we are going to define. Let $I_{pre}$ be the subset of $I_{evac}$ defined by $I_{pre} = \bigcup_{v \in V_o \setminus V_c} \mathsf{evac}(v) \cup \bigcup_{V \in V_c} \mathsf{evac}_1(v)$. Let $\rho_0$ be the plan consisting of the reduced form of the macro move $\mathcal{M}_\pi[I_{pre}]$. Plan $\rho_0$ is applicable to $\gamma$ because the sources of $\mathcal{M}_\pi[I_{pre}]$ are in $(P \setminus \{s\}) \cup T$ and its sinks are in $(P \setminus \{s\} \cup W)$; let $\delta_0$ denote the result of applying $\rho_0$ to $\gamma$. Let $V_c = \{v_1, \ldots, v_K\}$, let $u_i = u(v_i)$ and $b_i = b_{u_i}$, $1 \le i \le K$. Relabeling if necessary, we may assume that $b_1, \ldots, b_K$ appear on $P$ in this order when we scan from $s$ to $t$. Define a plan $\rho_i$ and a configuration $\delta_i$, $1 \le i \le K$, inductively as follows. Configuration $\delta_0$ is already defined above. Suppose $\delta_{i-1}$ has been defined. Let $\rho_i^1$ denote the minimal sequence of robot moves that brings the robot from its position $r_{\delta_{i-1}}$ in $\delta_{i-1}$ to $\mathrm{next}(b_i)$. Let $\rho_i^2$ denote the reduced form of the macro move $\mathcal{M}_\pi[\mathsf{evac}_2(v_i)]$. Let $\rho_i^3$ denote the reduced form of the macro move $\mathcal{M}_\pi[\mathsf{evac}_3(v_i)]$. Let $\rho_i^4$ denote the sequence of robot moves $\mathrm{next}(b_i) \Rightarrow b_i \parallel b_i \Rightarrow \mathrm{prev}(b_i)$. If $u_i$ has an obstacle in $\delta_{i-1}$, i.e., $u_i \in O_{\delta_{i-1}}$, then we put

$$\rho_i = \rho_i^1 \parallel \rho_i^3 \parallel \rho_i^4 \parallel \rho_i^2;$$

otherwise we put

$$\rho_i = \rho_i^2 \parallel \rho_i^1 \parallel \rho_i^3 \parallel \rho_i^4.$$

It can be easily verified by induction that $\rho_i$ is applicable to $\delta_{i-1}$; let $\delta_i$ be the result of applying $\rho_i$ to $\delta_{i-1}$. Note that, in $\delta_i$, no vertex on the path from $\mathrm{prev}(b_i)$ (or $s$ if $i = 0$) to $\mathrm{next}(b_{i+1})$ (or $t$ if $i = K$) has an obstacle.

Finally, let $\rho_{K+1}$ be the sequence of robot moves that brings the robot from $\mathrm{prev}(b_K)$ to $t$ and let $\rho_{K+2}$ be the reduced form of the macro move $\mathcal{M}_\pi[I'_{rem}]$ where $I'_{rem}$ is defined by $I'_{rem} = I_{rem} \cup \bigcup_{v \in V_c \setminus V'_c} \mathsf{evac}_4(v)$. Plan $\rho_{K+1}$ is applicable to $\delta_K$ since the path from $b_K$ to $t$ is clear of obstacles. To the result of this application, $\rho_{K+2}$ is applicable because (1) the macro move consisting of the obstacle moves in $\rho_1 \parallel \ldots \parallel \rho_K$ is equivalent to macro move $\mathcal{M}_\pi[I_\pi^{ob} \setminus I_r em']$ and (2) all the sources and sinks of $\mathcal{M}_\pi[I'_{rem}]$ are in the $s$-side of $t$. Our plan $\rho$ is defined by $\rho_0 \parallel \rho_1 \parallel \ldots \parallel \rho_{K+2}$. Clearly, $\rho$ is applicable to $\gamma$.

If we review how the macro moves of $\rho$ are derived, we see that macro move $\mathcal{M}_\pi$ is decomposed into disjoint macro moves and the reduced form of each of them is included in $\rho$. Therefore, macro move $\mathcal{M}_\rho$ is a subset of, and is equivalent to, $\mathcal{M}_\pi$. Since moreover $\rho$

is applicable to $\gamma$ and brings the robot from $s$ to $t$, $\rho$ is equivalent to $\pi$ on $\gamma$. Clearly the cost of obstacle moves of $\rho$ is no greater than that of $\gamma$, because $\mathcal{M}_\rho$ is a subset of $\mathcal{M}_\pi$. It remains to show that the cost of robot moves of $\rho$ is no greater than that of $\pi$. Suppose a backward robot move from $\mathrm{next}(v) \Rightarrow v$ for some vertex on $P \setminus t$ appears in $\rho$. Then, there is a critical pair at some $u$ with $b_u = v$ or $b_u = \mathrm{next}(v)$, which implies that $\pi$ must also contain the same backward robot move. Assume furthermore that $\mathrm{next}(v) \Rightarrow v$ appears twice in $\rho$. Then, there are critical pairs at some $u$ and $u'$, with $b_u = v$ and $b_{u'} = \mathrm{next}(v)$. This implies, by Claim 3 that the same robot move appears twice in $\pi$ as well. Since no backward robot move appears more than twice in $\rho$, we may conclude that the multiset of robot moves occurring in $\rho$ is the subset of that in $\pi$. Therefore, the cost of $\rho$ is no greater than that of $\pi$. This completes the proof of Lemma 4, except for the proofs of Claims 1, 2, and 3. □

In order to prove Claims 1, 2 and 3, we need the following lemma.

**Lemma 5** *Let $v$ be a vertex of tree $G$, let $v_1$, $v_2$, and $u$ be distinct vertices adjacent to $v$. Let $\gamma$ and $\delta$ be two configurations of $G$ such that*
*(1) None of $v_1$, $v_2$ and $v$ has an obstacle in $\gamma$ or in $\delta$, i.e., $\{v_1, v_2, v\} \subseteq H_\gamma \cap H_\delta$,*
*(2) $r_\gamma = v_1$ and $r_\delta = v_2$, and*
*(3) there is a matching from $\gamma$ to $\delta$.*
*Let $\mu$ be a minimum cost matching from $\gamma$ to $\delta$. Then there is a plan $\rho$ with robot's walk $v_1 \Rightarrow v \Rightarrow u \Rightarrow v \Rightarrow v_2$, such that applying $\rho$ to $\gamma$ results in $\delta$ and the total cost of obstacle moves in $\rho$ is $\mathsf{ocost}(\rho) = \mathsf{cost}(\mu) + 2\,\mathsf{cost}(v_1, v) + 2\,\mathsf{cost}(u, v) + 2\,\mathsf{cost}(v_2, v)$. Moreover, there is a plan $\rho$ as above with*
$\mathsf{ocost}(\rho) = \mathsf{cost}(\mu) + 2\,\mathsf{cost}(v_1, v)$ *if $u \in H_\gamma$;*
$\mathsf{ocost}(\rho) = \mathsf{cost}(\mu) + 2\,\mathsf{cost}(v_2, v)$ *if $u \in H_\delta$;*
$\mathsf{ocost}(\rho) = \mathsf{cost}(\mu)$ *if $u \in H_\gamma \cap H_\delta$.*

**Proof:** We may assume without loss of generality that $\mu$ is standard, i.e., $\mu(w) = w$ for every $w \in O_\gamma \cap O_\delta$. For each neighbor $v'$ of $v$, let $X(v')$ denote the set of vertices $w \in O_\gamma \setminus O_\delta$ such that there is a path from $w$ to $\mu(w)$ in $G \setminus \{v'\}$.

Our plan $\rho$ consists of robot moves $v_1 \Rightarrow v$, $v \Rightarrow u$, $u \Rightarrow v$, $v \Rightarrow v_2$ together with obstacle moves defined based on the matching $\mu$. Let $\rho_1$ ($\rho_u$, $\rho_2$, resp.) be an obstacle plan consisting of all the macro moves $w \to \mu(w)$, $w \in X(v_1)$ ($X(u)$, $X(v_2)$, resp.) in an arbitrary order. Note that we have $X(v_1) \cup X(u) \cup X(v_2) = O_\gamma$, so $\rho_1$, $\rho_u$, and $\rho_2$ together contain all the macro moves $w \to \mu(w)$, $w \in O_\gamma \setminus O_\delta$.

Let us start with the simplest case, where $u \in H_\gamma \cap H_\delta$. Then $\rho$ is simply defined as

$$\rho_1 \parallel v_1 \Rightarrow v \parallel v \Rightarrow u \parallel \rho_u \parallel u \Rightarrow v \parallel v \Rightarrow v_2 \parallel \rho_2,$$

which clearly satisfies the requirements and, in particular, satisfies $\mathsf{ocost}(\rho) = \mathsf{cost}(\mu)$.

Now suppose that $u \in H_\gamma \setminus H_\delta$. Since $v_2$ must be in $X(v_1) \cup X(u)$, the above plan still works unless $\mu(v_2) = u$. Suppose $\mu(v_2) = u$. Let $\rho_1'$ be obtained from $\rho_1$ by removing the macro move $v_2 \to u$. Our plan $\rho$ in this case is

$$\rho_1' \parallel v_1 \Rightarrow v \parallel v \Rightarrow u \parallel \rho_u \parallel v_2 \to v_1 \parallel u \Rightarrow v \parallel v \Rightarrow v_2 \parallel v_1 \to u \parallel \rho_2.$$

A straightforward tracing shows that $\rho$ is applicable to $\gamma$ and results in $\delta$. Moreover, $\mathsf{ocost}(\rho) = \mathsf{cost}(\mu) - \mathsf{cost}(v_2 \to u) + \mathsf{cost}(v_2 \to v_1) + \mathsf{cost}(v_1 \to u) = \mathsf{cost}(\mu) + 2\,\mathsf{cost}(v_1, v)$. The case where $u \in H_\delta \setminus H_\gamma$ is similar.

Finally, suppose $u \notin H_\delta \cup H_\gamma$. Our plan $\rho$ is

$$u \to v_2 \parallel \rho_1 \parallel v_1 \Rightarrow v \parallel v \Rightarrow u \parallel v_2 \to v_1 \parallel \rho_u \parallel v_2 \to v_1 \parallel u \Rightarrow v \parallel v \Rightarrow v_2 v_1 \to u \parallel \rho_2.$$

It is again straightforward to verify that $\rho$ is applicable to $\gamma$ and results in $\delta$. As for the cost, $\mathsf{ocost}(\rho) = \mathsf{cost}(\mu) + \mathsf{cost}(u \to v_2) + \mathsf{cost}(v_2 \to v_1) + \mathsf{cost}(v_1 \to u) = \mathsf{cost}(\mu) + 2\,\mathsf{cost}(v, v_1) + 2\,\mathsf{cost}(v, u) + 2\,\mathsf{cost}(v, v_2)$. □

**Proof of Claim 1:** Suppose $\pi$ is a path-plan that has an obstacle-critical pair $(i, j)$ at some sidestep vertex $u$. We use the notation in the proof of Lemma 4. Let $h$ be the smallest integer such that $\pi[h]$ is the robot move from $b_u$ to $\mathrm{next}(b_u)$. By Lemma 3 we may assume without loss of generality that $\pi[h-1]$ is the robot move from $\mathrm{prev}(b_u)$ to $b_u$. Similarly, let $k$ be the largest integer such that $\pi[k]$ is the robot move from $\mathrm{prev}(b_u)$ to $b_u$; we may assume that $\pi[k+1]$ is the robot move from $b_u$ to $\mathrm{next}(b_u)$. From the definition of an obstacle-critical pair, it is easily seen that $h < i < j < k$. We concentrate on the plan $\pi'$ consisting of the moves $\pi[h-1]$, $\pi[h]$, …, $\pi[k+1]$. Plan $\pi'$ is applied to $\gamma' = \gamma_{h-2}$ and results in $\delta' = \gamma_{k+1}$. We have a few cases to consider.

CASE 1: There are move indices $i'$ and $j'$, $h < i' \neq j' < k$, that are distinct from $i$ and $j$ such that $i'$ is from $b_u$ to $u$ and $j'$ is from $u$ to $b_u$. Let $\mu$ be an arbitrary minimum cost matching from $\gamma'$ to $\delta'$. Since macro move $\mathcal{M}_{\pi'}$ is reducible to a macro move in which two instances of $u \to b_u$ and two instances of $b_u \to u$ are removed, we have $\mathsf{cost}(\mu) \leq \mathsf{ocost}(\pi') - 4\,\mathsf{cost}(u, b_u)$. Applying Lemma 5, there is a plan $\rho'$ equivalent to $\pi'$, in which the robot's walk is $\mathrm{prev}(b_u) \Rightarrow b_u \Rightarrow u \Rightarrow b_u \Rightarrow \mathrm{next}(b_u)$, with

$$\begin{aligned}
\mathsf{ocost}(\rho') &= \mathsf{cost}(\mu) + 2\,\mathsf{cost}(\mathrm{prev}(b_u), b_u) + 2\,\mathsf{cost}(u, b_u) + 2\,\mathsf{cost}(\mathrm{next}(b_u), b_u) \\
&\leq \mathsf{ocost}(\pi') + 2\,\mathsf{cost}(\mathrm{prev}(b_u), b_u) + 2\,\mathsf{cost}(\mathrm{next}(b_u), b_u) - 2\,\mathsf{cost}(u, b_u).
\end{aligned}$$

On the other hand, we have $\mathsf{rcost}(\rho') \leq \mathsf{rcost}(\pi') + 2\,\mathsf{cost}(u, b_u) - 2\,\mathsf{cost}(\mathrm{prev}(b_u), b_u) - 2\,\mathsf{cost}(\mathrm{next}(b_u), b_u)$. Therefore, by replacing $\pi'$ by $\rho'$ in $\pi$ we obtain a plan $\rho$ equivalent to $\pi$ with no greater cost. Cut $\rho$ after the robot move $b_u \Rightarrow u$ and we obtain the required decomposition at $u$.

CASE 2: There is no move index $j' \neq i$, $h < j' < k$, such that $j'$ is from $u$ to $b_u$. Let $\tau$ denote the longest hole-wise thread of $\pi$ starting with $i$ and consisting solely of elements larger than $h$. Since $\tau$ cannot contain a move index that is from $u$ to $b_u$, $\mathsf{walk}(\tau)$ stays within $T_u$, except at its initial vertex $b_u$. Noting that $\mathsf{fin}(\tau)$ must have a hole in $\gamma'$, let $\gamma''$ be the result of applying macro move $u \to \mathsf{fin}(\tau)$ to $\gamma'$ and let $\mu'$ be an arbitrary minimum cost matching from $\gamma''$ to $\delta'$. Let $I \subseteq I_\pi^{ob}$ be defined by $I = \{l \in I_\pi^{ob} \mid h < l < k, l \notin \tau \cup \{j\}\}$. Then, macro move $\mathcal{M}_\pi[I]$ is applicable to $\gamma''$ and results in $\delta$. Therfore, we have $\mathsf{cost}(\mu') \leq \mathsf{ocost}(\pi') - \mathsf{cost}(u \to \mathsf{fin}(\tau)) - 2\,\mathsf{cost}(u, b_u)$. Applying Lemma 5 and obtain a plan $\rho''$ that sends $\gamma''$ to $\delta'$ such that the robot's walk in $\rho''$ is the same as in $\rho'$ above. Since $u$ has a hole in $\gamma'$, we may use the second case of the lemma, obtaining

$\mathsf{ocost}(\rho'') \leq \mathsf{cost}(\mu') + 2\,\mathsf{cost}(\text{prev}(b_u), b_u)$, and hence by the above inequality,

$$\mathsf{ocost}(\rho'') \leq \mathsf{ocost}(\pi') - \mathsf{cost}(u \to \mathsf{fin}(\tau)) - 2\,\mathsf{cost}(u, b_u) + 2\,\mathsf{cost}(\text{prev}(b_u), b_u).$$

Taking into account the difference of the robot cost, we have

$$\mathsf{cost}(\rho'') \leq \mathsf{cost}(\pi') - \mathsf{cost}(u \to \mathsf{fin}(\tau)).$$

By replacing $\pi'$ by $u \to \mathsf{fin}(\tau) \parallel \rho''$ in $\pi$, we obtain a plan equivalent to $\pi$ and with no greater cost than $\pi$. Cut this plan immediately after the robot move to $u$ to obtain the required decomposition of $\pi$.

CASE 3: There is no move index $i' \neq j$, $h < i' < k$, such that $i'$ is from $b_u$ to $u$. Let $\sigma$ denote the longest hole-wise thread ending with $j$ and consisting solely of elements smaller than $k$. Since $\sigma$ cannot contain a move index that is from $b_u$ to $u$, $\mathsf{walk}(\sigma)$ stays within $T_u$, except at its final vertex $b_u$. Noting that $\mathsf{init}(\sigma)$ must have a hole in $\delta'$, let $\delta''$ be the result of applying macro move $u \to \mathsf{init}(\sigma)$ to $\delta$. The rest of the proof proceeds similarly to Case 2, applying Lemma 5 to $\gamma'$ and $\delta''$. □

The proofs of Claim 2 and 3 are similar to each other. We prove Claim 3 and omit the simpler proof of Claim 2.

**Proof of Claim 3:** ¿From the assumptions that there is a hole-critical pair at each of $u$ and $u'$, that $b_{u'} = \text{next}(b_u)$, and that the backward robot move $b_{u'} \Rightarrow b_u$ appears at most once in $\pi$, it follows that the robot move immediately before $b_{u'} \Rightarrow b_u$ is $\text{next}(b_{u'}) \Rightarrow b_{u'}$ and the robot move immediately after $b_{u'} \Rightarrow b_u$ is $b_u \Rightarrow \text{prev}(b_u)$. Informally, the robot does not "turn around" at $b_u$ or $b_{u'}$. Let $h$ be the smallest integer such that $\pi[h]$ is the forward robot move $b_u \Rightarrow b_{u'}$ and $k$ be the largest such integer. By Lemma 3, we may assume that both $\pi[h-1] \parallel \pi[h] \parallel \pi[h+1]$ and $\pi[k-1] \parallel \pi[k] \parallel \pi[k+1]$ comprise a robot walk from $\text{prev}(b_u)$ to $\text{next}(b_{u'})$. Suppose first that $\mathsf{cost}(u, b_u) \leq \mathsf{cost}(u', b_{u'})$. Let $\pi'$ be the part of $\pi$ starting with the $(h-1)$st move and ending with the $k$th. Note $\pi'$ applies to $\gamma_{h-2}$ and results in $\gamma_k$. Let $\mu$ be an arbitrary minimum cost matching from $\gamma_{h-2}$ to $\gamma_k$. By our assumptions of the presence of critical pairs, we have $\mathsf{ocost}(\pi') \geq \mathsf{cost}(\mu) + 2(\mathsf{cost}(u, b_u) + \mathsf{cost}(u', b_{u'})) \geq \mathsf{cost}(\mu) + 4\,\mathsf{cost}(u, b_u)$. The rest of the proof proceeds similarly to Case 1 of the proof of Claim 1. The other case where $\mathsf{cost}(u, b_u) \geq \mathsf{cost}(u', b_{u'})$ is similar: instead of $\pi'$ defined above, deal with the part of $\pi$ starting with the $h$th move and ending with the $(k+1)$st. □

Let us turn our attention to the global structure of an optimal solution for a problem instance. The *robot trajectory* of a plan $\pi$ for $G$ is the subgraph of $G$ induced by the edges traversed by the robot in the application of $\pi$. Let $P$ be a path of $G$. A $P$*-caterpillar* of $G$ is a subgraph of $G$ consisting of $P$ and at most one vertex in $V(G) \setminus V(P)$ adjacent to each internal vertex of $P$. Thus, the degree of each vertex of a $P$-caterpillar is either 1, 2, or 3: we call a vertex with degree 3 a *joint* of the caterpillar. When $P$ is a path from $u$ to $v$, a $P$-caterpillar is also called a $u$-$v$ *caterpillar*. The *tail* of a $u$-$v$ caterpillar is defined to be the maximal subpath of the path from $u$ to $v$ that starts from $u$ and does not include a joint except possibly for the vertex adjacent to $u$.

**Lemma 6** *For any problem instance $(G, \gamma, t)$ where $G$ is a tree, there is a optimal solution for the instance such that its robot trajectory is an $s'$-$t$ caterpillar for some $s'$ that contains $s = r_\gamma$ in its tail.*

To prove this lemma, we need the following lemma that is used to "prune" long branches of a robot trajectory.

**Lemma 7** *Let $v$ be a vertex of tree $G$ with degree at least 3 and let $v_1$ and $v_2$ be distinct vertices adjacent to $v$. Let $\pi$ be a plan on $G$ of the form $v_1 \Rightarrow v \parallel \pi' \parallel v \Rightarrow v_2$, where $\pi'$ is a plan in which one of the robot moves is $v \Rightarrow u$ for some $u \notin \{v_1, v_2\}$, the first robot move is $v \Rightarrow u_1$ for some $u_1 \neq v_1$, the last robot move is $u_2 \Rightarrow v$ for some $u_2 \neq v_2$. Then, for each configuration $\gamma$ to which $\pi$ is applicable, there is a plan $\rho$ equivalent to $\pi$ on $\gamma$, with cost no greater than $\pi$, such that the robot's walk in $\rho$ is $v_1 \Rightarrow v \Rightarrow u \Rightarrow v \Rightarrow v_2$.*

**Proof:** Let $\delta$ be the result of applying $\pi$ to $\gamma$ and let $\mu$ be an arbitrary minimum cost matching from $\gamma$ to $\delta$. Define $X(v_1)$, $X(u)$, $X(v_2)$, $\rho_1$, $\rho_u$ and $\rho_2$ exactly in the same way as in the above proof of Lemma 5.

Suppose first that $u_1 = u$. There are several cases to consider. The simplest case is when $\mu(x) \neq u$ for every $x \in X(v_1)$. Then, our final plan $\rho$ is $\rho_1 \parallel v_1 \Rightarrow v \parallel v \Rightarrow u \parallel \rho_u \parallel u \Rightarrow v \parallel v \Rightarrow v_2 \parallel \rho_2$. It is straightforward to verify that $\rho$ is applicable to $\gamma$ and results in $\delta$. The cost of the obstacle moves of $\rho$ is exactly $\mathsf{cost}(\mu)$ in this case, and hence $\mathsf{cost}(\rho) \leq \mathsf{cost}(\pi)$. Next suppose that $\mu(u') = u$ for some $u' \in X(v_1)$. Note that $u_2 \neq u$ in this case, because $u_2 \in H_\delta$. Let $\rho_1'$ be obtained from $\rho_1$ by removing macro move $u' \to u$. If $u'$ is not in the $v_2$-side of $v$, then our plan $\rho$ is

$$\rho_1' \parallel v_1 \Rightarrow v \parallel v \Rightarrow u \parallel \rho_u \parallel u \Rightarrow v \parallel v \Rightarrow v_2 \parallel \rho_2 \parallel u' \to u,$$

with the cost condition satisfied similarly to the above. On the other hand, if $u'$ is in the $v_2$-side of $v$, our plan $\rho$ is

$$\rho_1' \parallel v_1 \Rightarrow v \parallel v \Rightarrow u \parallel \rho_u \parallel u' \to u_2 \parallel u \Rightarrow v \parallel v \Rightarrow v_2 \parallel \rho_2 \parallel u_2 \to u.$$

The cost of obstacle moves of $\rho$ in this case is $\mathsf{cost}(\mu) - \mathsf{cost}(u' \to u) + \mathsf{cost}(u' \to u_2) + \mathsf{cost}(u_2 \to u) = \mathsf{cost}(\mu) + 2\,\mathsf{cost}(v, u_2)$. On the other hand, the cost of the robot moves in $\rho$ is at least $2\,\mathsf{cost}(v, u_2)$ less than that of $\pi$. Therefore $\mathsf{cost}(\rho) \leq \mathsf{cost}(\pi)$ in this case as well.

The case $u = u_2$ is similar (in fact it is symmetric to the above case with respect to the reversal of the motion.)

Now suppose $u \notin \{u_1, u_2\}$. We may assume without loss of generality that $u_1 = v_2$. To see this, suppose $u_1 \neq v_2$. Let $\pi'$ be the suffix of $\pi$ starting with the first occurrence of the robot move from $u_1$ to $v$. We may inductively apply the lemma and obtain $\rho'$ equivalent to $\pi'$, in which the robot's walk is $u_1 \Rightarrow v \Rightarrow u \Rightarrow v \Rightarrow v_2$. Therefore, if we replace $\pi'$ by $\rho'$ in $\pi$, the problem now reduces to the case where $u = u_2$, which has already been dealt with. Similarly, we may assume without loss of generality that $u_2 = v_1$.

Let $i_u$ be an integer such that $\pi[i_u]$ is the robot move from $v$ to $u$. Let $\pi_u$ be the prefix of $\pi$ consisting of $i_u$ moves, and let $\gamma_u$ be the result of applying $\pi_u$ to $\gamma$. Clearly, $u$ has a hole in $\gamma_u$. Let $\mu_1$ be an arbitrary minimum cost matching from $\gamma$ to $\gamma_u$ and $\mu_2$ be an arbitrary minimum cost matching from $\gamma_u$ to $\delta$. Let $u' = u$ if $u \in H_\delta$ and $u' = \mu_1(u)$ otherwise. Suppose first that $u'$ is not in the $v_1$-side of $v$. Let $\gamma'$ be the result of applying macro move $u \rightarrow u'$ to $\gamma$ and let $\mu'$ be an arbitrary minimum cost matching from $\gamma'$ to $\delta$. Since $\mathsf{cost}(\mu_1) - \mathsf{cost}(u \rightarrow u') + \mathsf{cost}(\mu_2) \geq \mathsf{cost}(\mu')$, we have $\mathsf{cost}(\mu') \leq \mathsf{ocost}(\pi) - \mathsf{cost}(u \rightarrow u')$. Applying Lemma 5 to $\gamma'$ and $\delta$, obtain a plan $\rho'$ that sends $\gamma'$ to $\delta$ with its robot walk being as required. We have $\mathsf{ocost}(\rho') = \mathsf{cost}(\mu') + 2\,\mathsf{cost}(v_1, v) \leq \mathsf{ocost}(\pi) - \mathsf{cost}(u \rightarrow u') + 2\,\mathsf{cost}(v_1, v)$. Our plan $\rho$ is $u \rightarrow u' \parallel \rho'$, which is clearly applicable to $\gamma$ and yields $\delta$. Since $\mathsf{ocost}(\rho) \leq \mathsf{ocost}(\pi) + 2\,\mathsf{cost}(v_1, v)$ and $\mathsf{rcost}(\rho) \leq \mathsf{rcost}(\pi) - 2(\mathsf{cost}(v_1, v) + \mathsf{cost}(v_2, v))$, we have $\mathsf{cost}(\rho) \leq \mathsf{cost}(\pi)$. The case where $u'$ is not in the $v_2$-side of $v$ is similar. This completes the proof since $u'$ cannot be in the $v_1$-side of $v_2$ and in the $v_2$-side of $v$ at the same time. □

**Proof of Lemma 6:** Let $\pi$ be an arbitrary optimal solution for the instance and let $T$ be the robot trajectory of $\pi$. ¿From the optimality of $\pi$, we may assume that $t$ a leaf of $T$. Among the leaves of $T$ that are not in the $t$-side of $s$, choose $s'$ that is visited first by the robot in the application of $\pi$. Let $P$ denote the path from $s'$ to $t$. Note that $s$ is on $P$, with the possibility that $s = s'$. For each internal vertex $v$ of $P$, let $T_v$ denote the subtree of $T$ consisting of the vertices connected to $v$ without going through other vertices on $P$. Suppose there is some internal vertex $v$ of $P$ such that $|V(T_v)| > 2$. Let $u \in V(T_v)$ be a vertex adjacent to $v$ and and let $T'$ be the tree obtained from $T$ by replacing removing all the vertices in $V(T_v) \setminus \{u, v\}$. We claim that there is a plan $\pi'$ equivalent to $\pi$ on $\gamma$ without increased cost such that its robot trajectory is $T'$. This is shown by the application of Lemma 7 in the following manner. Let $v_1$ be the vertex adjacent to $v$ in the $s'$-side of $v$ and $v_2$ the vertex adjacent to $v$ in the $t$-side of $v$. Let $\pi_v$ denote the part of $\pi$ that (1) starts with the first occurrence of the robot move $v_1 \Rightarrow v$ such that the next robot move is not $v \Rightarrow v_1$, and (2) ends with the last occurrence of the robot move $v \Rightarrow v_2$ such that the preceding robot move is not $v_2 \Rightarrow v$. Clearly, $\pi_v$ contains an occurrence of the robot move $v \Rightarrow u$. Therefore, applying Lemma 7 to $\pi_v$ and substituting the result for $\pi_v$ in $\pi$, we obtain a plan $\pi'$ whose robot trajectory is $T'$. Repeating this process, we eventually obtain a plan equivalent to $\pi$ without increased cost, whose robot trajectory is an $s'$-$t$ caterpillar $C$. If $s$ is in the tail of $C$ then we are done. Otherwise, we can choose a leaf $s''$ of $C$ so that the maximal $s''$-$t$ caterpillar $C'$ that is a subgraph of $C$ contains $s$ in its tail. By a process similar to the above, we may obtain a plan whose robot trajectory is $C'$. □

We need one more lemma before proving the main result of this section.

**Lemma 8** *Let $\pi$ be a plan for tree $G$ applicable to a configuration $\gamma$ and let $\delta$ be the resulting configuration. Let $s = r_\gamma$ be the initial location of the robot and let $u \neq s$ be a vertex of $G$ such that (1) $u \in H_\gamma$ and (2) the robot's walk in $\pi$ does not visit any vertex in the $u$-side of $s$. Let $v$ be a vertex* not *in the $u$-side of $s$ such that $v \in O_\gamma$ and let $\gamma'$ be a configuration defined by $r_{\gamma'} = r_\gamma$ and $O_{\gamma'} = (O_\gamma \setminus \{v\}) \cup \{u\}$. Then, there exists a plan $\pi'$ with $\mathsf{cost}(\pi') \leq \mathsf{cost}(\pi) + \mathsf{cost}(u \rightarrow v)$ that is applicable to $\gamma'$ and results in delta.*

**Proof:** A simple transformation. □

**Theorem 9** *For any given problem instance $(G, \gamma, t)$, where $G$ is a tree, there is an optimal solution that is quasi-bitonic.*

**Proof:** By Lemma 6, there is an optimal solution $\pi$ whose robot trajectory is an $s'$-$t$ caterpillar $C$ that contains $s = r_\gamma$ in its tail. Let $u_1, \ldots, u_K$ be the list of leaves of $C$ excluding $s'$ and $t$, in the order of appearance along $P$. Let $u_0 = s'$ and $u_{K+1} = t$ for convenience. For each $i$, $1 \le i \le K$, let $b_i$ denote the internal vertex on $C$ that is adjacent to $u_i$. ¿From the proof of Lemma 6, we may assume that the robot's walk in $\pi$ is a concatenation of a walk from $s$ to $s'$ and a path-shaped walk from $u_i$ to $u_{i+1}$ for each $i$, $0 \le i \le K$. Moreover, by being careful in the application of Lemma 7 in the proof of Lemma 6, the path-shaped walk from $u_i$ to $u_{i+1}$ for each $i$, $1 \le i \le K$, may be assumed to consist of a move $u_i \Rightarrow b_i$ followed by a path-shaped walk from $b_i$ to $u_{i+1}$, i.e., the move $u_i \Rightarrow b_i$ is not repeated. By Lemma 4, we may replace the path-shaped walk from $s'$ to $u_1$ or from $b_i$ to $u_{i+1}$ for each $1 \le i \le K$ by a quasi-monotonic walk without increasing the cost of the plan. Moreover, from the proof of Lemma 4, we may assume that no wiggle or sidestep of the quasi-monotonic walk from $b_i$ to $u_{i+1}$ is at $b_{i+1}$. Therefore, the resulting robot's walk from $s'$ to $t$ is quasi-monotonic.

It remains to show that the robot's walk from $s$ to $s'$ is a path. We first assume that the length of $\pi$ is the smallest possible, among the optimal solutions; otherwise, we simply appeal to the induction on the length of the plan. By Lemma 4, we may assume that the robot's walk from $s$ to $s'$ is quasi-monotonic. Suppose it contains a wiggle. Let $v$ be the vertex closest to $s'$ such that the walk from $s$ to $s'$ contains a wiggle at $v$. Let $v_1$ and $v_2$ be the vertices on the path from $s$ to $s'$ that are adjacent to $v$, with $v_1$ on the $s$-side of $v$. Let $P'$ denote the path from $v_1$ to $s'$. Let $\pi$ be represented as $\pi = \pi_1 \parallel \pi_2 \parallel \pi_3 \parallel \pi_4$, so that
(1) the first move of $\pi_2$ is the first robot move of $\pi$ from $v_1$ to $v$,
(2) the first move of $\pi_3$ is the second robot move of $\pi$ from $v_1$ to $v$, and
(3) the first move of $\pi_4$ is the last robot move of $\pi$ from $v$ to $v_1$.
¿From the proof of Lemma 4, we may assume that $\pi_2$ is of the form $v_1 \Rightarrow v \parallel v \Rightarrow v_2 \parallel u \to x \parallel v_2 \Rightarrow v \parallel v \Rightarrow v_1 \parallel y \to u$, where $u$ is a vertex next to $v$ and not on $P$, $x$ is a vertex in the $v_1$-side of $v$, and $y$ is the vertex adjacent to $v_2$ on $P$ and in the opposite side to $v$. Let $\gamma_1$ denote the result of applying $\pi_1$ to $\gamma$, $\gamma_2$ the result of applying $\pi_2$ to $\gamma_1$, and $\gamma_3$ the result of applying $\pi_3$ to $\gamma_2$. Since the robot's walk from $v_1$ to $s'$ in $\pi_3$ is a path by the assumption, we may assume, by the proof of Lemma 4, that each vertex of $P'$ has a hole in $\gamma_2$. Turning our attention to $\pi_3$, we first note that macro move $\mathcal{M}_{\pi_3}$ has at least one connected source-sink pair with the source in the $v_1$-side of $v$ and with the sink in the $v_2$-side of $v$. To see this, suppose to the contrary macro move $\mathcal{M}_{\pi_3}$ does not have such a source-sink pair. Then, $\mathcal{M}_{\pi_3}$ can be decomposed into three macro moves $M_1$, $M_2$, and $M_3$ such that (1) all the sources and sinks of $M_1$ are in the $v_1$-side of $v$, (2) all the sources and sinks of $M_2$ are in the $v_2$-side of $v$, and (3) all the sources of $M_3$ are in the $v_2$-side of $v$ and all the sinks of $M_3$ are in the $v_1$-side of $v$. Let $\gamma_3'$ be the result of applying plan $v_1 \Rightarrow v \parallel M_1 \parallel M_2$ to $\gamma_2$. Applying Lemma 8 repeatedly to $\pi_4$ and $\gamma_3$, we obtain a plan $\pi_4'$ applicable to $\gamma_3'$ with $\mathsf{cost}(\pi_4') \le \mathsf{cost}(\pi_4) + \mathsf{cost}(M_3)$ such that $\pi' = \pi_1 \parallel \pi_2 \parallel \pi_3 \parallel \pi_4'$ is a solution to the given instance. Clearly, $\mathsf{cost}(\pi') \le \mathsf{cost}(\pi)$ and the length of $\pi'$ is strictly

smaller than the length of $\pi$, contradicting our assumption that $\pi$ is the shortest optimal solution.

Therefore, as claimed, macro move $\mathcal{M}_{\pi_3}$ has at least one connected source-sink pair $(z, w)$ such that $z$ is in the $v_1$-side of $v$ and $w$ is in the $v_2$-side of $v$. Clearly $z \neq v_1$ because $v_1 \in H_{\gamma_2}$. Moreover, from the proof of Lemma 4, we may choose such a pair $(z, w)$ so that $\pi_3$ contains either (a) a macro move $z \to w$ or (b) two macro moves $z \to u$ and $u \to w$, where $u$ is some sidestep vertex along $P'$. We may furthermore assume that $w$ is not on $P'$. This is because, if no such pair $(z, w)$ with $w$ not on $P'$ exists, then we can transform $\pi_3$ into a shorter equivalent plan by short-cutting the robot's visit to $s'$ without increasing the cost; again a contradiction with the assumption of the shortest solution.

Now, let $\pi_3''$ be the plan obtained from $\pi_3$ by removing macro move $z \to w$ (case (a) above) or macro moves $z \to u$ and $u \to w$ (case b). It can be easily verified that plan $y \to w \parallel z \to x \parallel \pi_3''$ is applicable to $\gamma_1$ and results in $\gamma_3$. Moreover, the cost of this plan is not greater than that of $\pi_2 \parallel \pi_3$. Therefore, $\pi$ can be transformed into an equivalent shorter plan without increasing the cost, a contradiction to our assumption. $\square$