Intermittent Cooperation in Path Planning for Selfish Agents

Supplementary Material

A PRELIMINARIES

We extend the notation used in the main paper to support the formal proofs provided in the supplementary material.

Let $\Pi_{v_1 \dots v_l}$ denote the set of all paths starting at v_1, ending at v_l, and visiting $v_2, \dots, v_{l-1}$ in order, possibly including additional nodes (for example, the path v_1, v_7, v_2, v_3, v_4 is in Π_{v_1, v_2, v_4}). The *path time* $T_{v,u}(\pi^1 \mid \pi^2)$ denotes the travel time incurred by an agent traversing path π^1 from node v to node u, given that the other agent follows path π^2. This includes edge traversal times as well as node travel delays induced by synchronization and cooperation, excluding the travel delays at the first and last nodes.

We use $D_{v,u}(\pi^1 \mid \pi^2)$ to denote the departure time from node u of an agent traveling along the path π^1 from node v, given that the other agent follows the path π^2. Formally, $D_{v,u}(\pi^1 \mid \pi^2)$ is given by the sum of the path time $T_{v,u}(\pi^1 \mid \pi^2)$ and the travel delay incurred at node u. The total path time of the full path π^1, from its starting node to its ending node, is denoted by $T(\pi^1 \mid \pi^2)$. The total departure time is represented by $D(\pi^1 \mid \pi^2)$.

Let $T_{v,u}(\pi)$ denote the independent travel time of an agent along path π from node v to node u, computed under the assumption that the agent traverses all nodes alone and incurs travel delay τ^1_w at each node $w \in \pi$. Let $D_{v,u}(\pi)$ denote the corresponding departure time of the agent from node u in this independent context.

A path π for agent a is considered *rational* if, for any two nodes v and u along the path, the condition $T_{v,u}(\pi|\pi) \leq T_{v,u}(SP^1_{v,u})$ holds. This implies that if the agent cooperates in all nodes along the path, the route between any two nodes is the shortest possible.

A.1 Path Time Calculation

Consider two agents a_1 and a_2 arriving at a cooperation node $c \in V_C$ at times t^1_c and t^2_c, respectively, where $t^2_c \in [t^1_c,\ t^1_c + \tau^1_c]$. This implies, without loss of generality, that agent a_2 arrives after a_1 while a_1 is still present at the node. The agents cooperate and incur the reduced travel delay at node c only if cooperation is beneficial (or at least not harmful) to both. While cooperation is always beneficial for the later-arriving agent (a_2 in this example), it is beneficial for the earlier-arriving agent (a_1) only if the induced waiting is advantageous.

Under these cooperation conditions, when agent a_i arrives at node v at time t^i_v, its waiting time for the other agent a_{-i}, denoted w^i_v, and its incurred travel delay δ^i_v, depend on the arrival time t^{-i}_v of a_{-i}:

$$w^i_v(t^i_v, t^{-i}_v) = \begin{cases} t^{-i}_v - t^i_v & \text{if } t^i_v \in [\, t^{-i}_v - (\tau^1_v - \tau^2_v),\ t^{-i}_v \,], \\ 0 & \text{otherwise.} \end{cases}$$

$$\delta^i_v(t^i_v, t^{-i}_v) = \begin{cases} \tau^2_v & \text{if } t^i_v \in [\, t^{-i}_v - (\tau^1_v - \tau^2_v),\ t^{-i}_v + (\tau^1_v - \tau^2_v) \,], \\ \tau^1_v & \text{otherwise.} \end{cases}$$

The total delay incurred at node v, combining waiting time and the travel delay, is denoted by

$$\lambda^i_v(t^i_v, t^{-i}_v) = w^i_v(t^i_v, t^{-i}_v) + \delta^i_v(t^i_v, t^{-i}_v)$$

Accordingly, the travel time of agent a_i from node v to node u along path π^i, while the other agent follows path π^{-i}, is given by

$$T_{v,u}(\pi^i \mid \pi^{-i}) = \begin{cases} 0 & \text{if } v = u, \\ \displaystyle\sum_{(x,y) \in \pi^i_{v,u}} \tau_{xy} + \sum_{x \in \pi^i_{v,u} \setminus \{v,u\}} \lambda^i_x\big(T_{v,x}(\pi^i \mid \pi^{-i}),\ T_{v,x}(\pi^{-i} \mid \pi^i)\big) & \text{otherwise.} \end{cases}$$

The corresponding departure time of agent a_i from node u, when traveling from node v to node u along path π^i while the other agent follows path π^{-i}, is given by

$$D_{v,u}(\pi^i \mid \pi^{-i}) = T_{v,u}(\pi^i \mid \pi^{-i}) + \lambda^i_u\big(T_{v,u}(\pi^i \mid \pi^{-i}),\ T_{v,u}(\pi^{-i} \mid \pi^i)\big)$$

B JOINING SEGMENT

B.1 Proof of Lemma 1

Lemma 1 (Early Cooperation)

Each agent prefers to initiate cooperation at the earliest possible cooperation node along the other agent's path. That is, given an agent a_i's path, the best response for the other agent, a_{-i}, is to join a_i as early as possible along its path, and together the optimal joint strategy from both agents' perspective is to initiate the cooperation as early as they can.

Proof. Consider a system with two agents, a_1 and a_2, where a_2's path is π^2. Let V_{C^*} be the set of all cooperation nodes in π^2 that a_1 can reach in a relevant time to cooperate:

$$V_{C^*} = \{v \in V_C \mid T(SIP_{s_1,v}) \leq T_{s_2,v}(\pi^2) + \tau^1_v - \tau^2_v\}$$

The path π^2 dictates a chronological order for the cooperation nodes. Let c_i^* denote the i-th cooperation node in V_{C^*} that a_2 visits according to the path π^2.

Then, to prove the lemma we prove that the *fastest* cooperation path starts at the earliest reachable cooperation node $c_1^* \in V_{C^*}$. Formally we prove that for any node $c_i^* \in V_{C^*}$ with $i > 1$, it holds that:

$$\forall \pi_{c_i^*,g_1} \in \Pi_{c_i^*,g_1}, \quad T_{s_1,g_1}(SIP_{s_1,c_1^*} \circ \pi'_{c_1^*,g_1} | \pi^2) \leq T_{s_1,g_1}(SIP_{s_1,c_i^*} \circ \pi_{c_i^*,g_1} | \pi^2)$$

where

$$\pi'_{c_1^*,g_1} = \pi^2_{c_1^*,c_i^*} \circ \pi_{c_i^*,g_1}$$

Given the path $\pi_{c_i^*,g_1}$, we construct the path $\pi'_{c_1^*,g_1} = \pi^2_{c_1^*,c_i^*} \circ \pi_{c_i^*,g_1}$ and examine the full paths $\hat{\pi} = SIP_{s_1,c_i^*} \circ \pi_{c_i^*,g_1}$ and $\hat{\pi}' = SIP_{s_1,c_1^*} \circ \pi'_{c_1^*,g_1}$. We need to show that:

$$T_{s_1,g_1}(\hat{\pi}' | \pi^2) \leq T_{s_1,g_1}(\hat{\pi} | \pi^2)$$

Since the path from c_i^* to g_1, denoted $\pi_{c_i^*,g_1}$, is the same in both cases, it suffices to show that the departure time from c_i^* using the path $SIP_{s_1,c_1^*} \circ \pi'_{c_1^*,g_1}$ is earlier than the departure time using the path $SIP_{s_1,c_i^*} \circ \pi_{c_i^*,g_1}$. When using SIP_{s_1,c_i^*}, agent a_1 arrives at c_i^* in a relevant time to cooperate, and its departure time from this node is given by:

$$D_{s_1,c_i^*}(SIP_{s_1,c_i^*} | \pi^2) = \max\left(T_{s_1,c_i^*}(SIP_{s_1,c_i^*} | \pi^2), T_{s_2,c_i^*}(\pi^2 | SIP_{s_1,c_i^*})\right) + \tau^2_{c_i^*}$$

However, using $\hat{\pi}'$, a_1 arrives at c_1^* at time to cooperate and therefore the delay of a_2 at c_1^* is reduced and the arrival time of a_1 at c_i^* (which is the same as the arrival time of a_2) is $T_{s_1,c_i^*}(\hat{\pi}' | \pi^2) = T_{s_2,c_i^*}(\pi^2 | \hat{\pi}') \leq T_{s_2,c_i^*}(\pi^2 | SIP_{s_1,c_i^*})$. The departure time of a_1 from c_i^* is then: $D_{s_1,c_i^*}(\hat{\pi}' | \pi^2) = T_{s_2,c_i^*}(\pi^2 | \hat{\pi}') + \tau^2_{c_i^*}$ and it holds that:

$$T_{s_1,c_i^*}(\hat{\pi}' | \pi^2) + \tau^2_{c_i^*} = T_{s_2,c_i^*}(\pi^2 | \hat{\pi}') + \tau^2_{c_i^*} \;\leq$$

$$\max(T_{s_1,c_i^*}(SIP_{s_1,c_i^*} | \pi^2), T_{s_2,c_i^*}(\pi^2 | \hat{\pi}')) + \tau^2_{c_i^*} \leq$$

$$\max(T_{s_1,c_i^*}(SIP_{s_1,c_i^*} | \pi^2), T_{s_2,c_i^*}(\pi^2 | SIP_{s_1,c_i^*})) + \tau^2_{c_i^*}$$

$$D_{s_1,c_i^*}(\hat{\pi}' | \pi^2) \leq D_{s_1,c_i^*}(SIP_{s_1,c_i^*} | \pi^2)$$

□

B.2 All Nodes Shortest Non-Cooperative Partial Paths

Algorithm 1 finds the shortest Non-Cooperative partial path from a given starting node s_i to all nodes in the graph. The algorithm is identical to Dijkstra's algorithm, except for one modification: if $c \in V_C$ can be reached by a_{-i} at a time relevant for cooperation, then once the shortest non-cooperative partial path to c is found, the node is removed from the graph. This ensures that c is not considered as part of the non-cooperative path to any other node in the graph. The algorithm initializes by setting all paths from s_i to infinity (except s_i, which is set to 0 with a trivial path) and defining $Q \leftarrow V$ as the set of unvisited nodes [lines 1–3]. The algorithm then evaluates all unvisited nodes v that are reachable from s_i in ascending order of their path time from s_i [lines 4-11]. For each node v, ignoring nodes that can be leveraged by the other agent to initiate an earlier cooperation [line 7], the algorithm iterates over its neighbors. For each neighbor, if the shortest Non-Cooperative path from s_i to it via v is shorter than its current path, the algorithm updates the path and its associated time [lines 8-11]. Once all reachable nodes have been evaluated, the algorithm returns a mapping of each node in the graph to its corresponding shortest Non-Cooperative partial path from s_i [line 12].

LEMMA 4. *Algorithm 1 computes the* ***shortest*** Non-Cooperative Partial Paths *from a given starting node to all nodes in the graph in polynomial time.*

The correctness of Lemma 4 follows from the optimality of Dijkstra's algorithm. Since the removal of nodes from the graph is done only when the shortest non-cooperative partial path to them is found, and the removed nodes are not part of any shortest non-cooperative path to any other node when this removal is performed, the path time and non-cooperation conditions are preserved. Similar to Dijkstra's algorithm, the complexity of Algorithm 2 is $O(|E| + |V| \log |V|)$.

C COOPERATION SEGMENT

C.1 Proof of Lemma 2

Lemma 2 (Cooperation Continuity)
Consider two cooperation nodes. If both agents cooperate between them, then neither agent can strictly improve its path time by deviating to an individual path and rejoining later.

PROOF. To formally prove the lemma, we consider a system with two agents, a_1 and a_2. Without loss of generality, assume that agent a_2 follows a rational path π, and that the agents initiate cooperation at a cooperation node $c_i \in \pi$. We show that the best response of the other agent, a_1, yielding the fastest path from c_i to any subsequent cooperation node $c_j \in \pi$, is to follow the subpath π_{c_i,c_j} of π. Formally, we prove that for any path

$$\pi' \in \Pi_{s_1,c_i,c_j} \quad \text{such that} \quad D_{s_1,c_i}(\pi' \mid \pi) = D_{s_2,c_i}(\pi \mid \pi'),$$

it holds that

$$T_{c_i,c_j}\left(\pi'_{s_1,c_i} \circ \pi_{c_i,c_j} \mid \pi\right) \;\leq\; T_{c_i,c_j}(\pi' \mid \pi).$$

Since a_1 and a_2 depart from node c_i at the same time, traveling together on the same path from c_i to any subsequent node v ensures they arrive at v simultaneously. If v is a cooperation node, this synchronization allows them to cooperate immediately upon arrival, minimizing the latency at that node.

Assume, by contradiction, that there exists a path $\pi' \in \Pi_{s_1,c_i,c_j}$ such that $D_{s_1,c_i}(\pi'|\pi) = D_{s_2,c_i}(\pi|\pi')$ and $T_{c_i,c_j}(\pi'_{s_1,c_i} \circ \pi_{c_i,c_j}|\pi) > T_{c_i,c_j}(\pi'|\pi)$. Since the partial path to c_i is the same in both paths π' and $\pi'_{s_1,c_i} \circ \pi_{c_i,c_j}$, the deviation between the two paths must occur along the partial path starting at c_i, that is, $\pi'_{c_i,c_j} \neq \pi_{c_i,c_j}$.

For any deviation between the paths, denote the last node before the deviation as d and the first node after the intersection of the paths as r. Because a_2 is rational, it holds that $T_{d,r}(\pi|\pi) \leq T_{d,r}(SIP_{d,r})$. Since there are only two agents, and their partial path from d to r doesn't intersect, no cooperation can occur along the path $\pi'_{d,r}$. Therefore,

$$T_{d,r}(\pi|\pi) \leq T_{d,r}(SIP_{d,r}) \leq T_{d,r}(\pi'|\pi)$$

This implies that

$$T_{c_i,r}(\pi|\pi) \leq T_{c_i,r}(\pi'|\pi)$$

for any deviation of π'_{c_i,c_j} from π_{c_i,c_j}. Thus,

$$T_{c_i,c_j}(\pi'_{s_1,c_i} \circ \pi_{c_i,c_j}|\pi) \leq T_{c_i,c_j}(\pi'|\pi)$$

which contradicts our initial assumption.
Therefore, for any node $c_j \in \pi$ visited after c_i, the *fastest path* for a_1 from c_i to c_j is achieved by following the path π_{c_i,c_j}. □

C.2 All Nodes Shortest Stable Cooperation Partial Paths

Algorithm 2 finds the shortest stable cooperation partial paths ending at node c_d from all nodes in the graph, working in reverse order from c_d to all nodes in the graph.

The algorithm initializes all paths to c_d as empty with infinite time, except c_d itself, which is set to time 0 with a trivial path, and defines $Q \leftarrow V$ as the set of unvisited nodes [lines 1–3]. The algorithm then evaluates all unvisited nodes v that can reach c_d in ascending order of their distance to c_d [lines 4-10]. For each node v, only its neighbors that preserve path stability are considered as its neighbors [line 7]. If the shortest stable cooperation path from a neighbor u of v via v to c_d is shorter than its current path, the algorithm updates the path and its associated time [lines 8-10]. Once all reachable nodes have been evaluated, the algorithm returns a mapping of each node in the graph to its corresponding *shortest* stable cooperation partial path to c_d [line 11].

LEMMA 5. *Given a cooperation ending node c_d, Algorithm 2 computes the* ***shortest*** Stable Cooperation Partial Path *concluding cooperation at c_d from every node in the graph in polynomial time.*

Similar to Lemma 4, the correctness of Lemma 5 follows from the optimality of Dijkstra's algorithm, with one adjustment. Since the neighbors of a node are filtered to maintain stability (line 7), we must also ensure that when a node v is pulled from Q, its set of stable neighbors, as filtered in line 7, is final and reflects all and only the stable neighbors of v. Specifically, we need to ensure that once node v is removed from Q, for all its neighbors $u \in V$ such that $(v, u) \in E$, the value $\tau_{u,v} + \tau_v^2 + T_{v,c_d}$ is fixed and will not change from that point on. The only element in this value that can change during the algorithm's execution is T_{v,c_d}, but once v is pulled from Q, its value remains fixed. Thus, the path times and stability conditions are correctly computed and finalized for each node upon removal from Q, ensuring the overall optimality of the algorithm. Similar to Dijkstra's algorithm, the complexity of Algorithm 2 is $O(|E| + |V| \log |V|)$.

D FINDING A BEST RESPONSE STRATEGY

We assume that the path π of agent a_2 is fixed and known to agent a_1, and seek the best response for a_1, i.e., the strategy that produces the *fastest path*, minimizing its arrival time at g_1. We assume that all m cooperation nodes are included in π (if not, m represents the number of cooperation nodes in π). For each cooperation node $c \in V_C$, we examine whether cooperation can and should be considered at that node. Gladly, from Lemmas 1 and 2 it follows that best response cooperation-based strategy has a simple structure: cooperation should start as early as possible, and it is continuous (that is, it is not beneficial to depart from cooperation and re-engage). This observation substantially reduces the strategy space for agent a_1, from 2^m possibilities to a number linear in m.

To find where it is possible to initiate cooperation, we find the shortest independent paths from s_1 to all m cooperation nodes: $SP^1_{s_1,c_1} \dots SP^1_{s_1,c_m}$, which determines the earliest arrival times $T^1_{s_1,c_1}(SP^1_{s_1,c_1}), \dots, T^1_{s_1,c_m}(SP^1_{s_1,c_m})$ for a_1 without considering cooperation with a_2. These arrival times are then compared against the arrival times of a_2 at these nodes. A cooperation node c is considered reachable by a_1 within *relevant* time for cooperation if a_1 can reach it within the following time frame, allowing it to cooperate with a_2 at c:

$$T^1(SP^1_{s_1,c}) \leq T^2_c(\pi) + \tau^1_c - \tau^2_c$$

While cooperating at c only improves a_1's departure time from that node when $T^2_c(\pi) - (\tau^1_c - \tau^2_c) \leq T^1(SP^1_{s_1,c}) \leq T^2_c(\pi) + \tau^1_c - \tau^2_c$, considering the potential for ongoing cooperation, a_1 may choose to wait upon early arrival (before $T^2_c(\pi) - (\tau^1_c - \tau^2_c)$) to cooperate with a_2, thereby aiding a_2 in improving its departure time, which could also benefit a_1 at subsequent nodes (see example in Figure 8). Denote the set of all nodes reachable by a_1 in relevant time for cooperation as V_{C^*}:

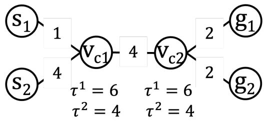


Figure 8: Agent a_1 reaches node v_{c_1} in time for cooperation. While it could depart earlier without cooperating ($t = 7$ vs. $t = 8$), cooperation reduces the total path time (18 vs. 19).

$$V_{C^*} = \{c \in V_C \mid T^1(SP^1_{s_1,c}) \leq T_c(\pi) + \tau^1_c - \tau^2_c\}$$

The path π dictates a chronological order for the cooperation nodes. Let c^*_i denote the i-th cooperation node in V_{C^*} that a_2 visits according to the path π, and let V'_{C^*} denote the set of all cooperation nodes that follow c^*_1 in π: $V'_{C^*} = \{c \in V_C \mid t^2_c(S^2) \geq t^2_{c^*_1}(S^2)\}$. If V_{C^*} is empty, the *fastest* path for a_1 is the independent shortest path to g_1. Otherwise, the following theorem follows directly from Lemmas 1 and 2:

THEOREM 4. *The* fastest *cooperation path is the path that starts a cooperation at c^*_1 and follows π to the optimal departure node d defined by*

$$d = \arg\min_{v \in V'_{C^*}} T(\pi_{c^*_1,v} \circ SP^1_{v,g_1} | \pi) \tag{3}$$

That is $\pi^* = SP^1_{s_1,c^*_1} \circ \pi_{c^*_1,d} \circ SP^1_{d,g_1}$.

This means that a_1 takes the shortest path to the earliest node in which it can cooperate with a_2, then travels with a_2 along its path, leveraging additional cooperation nodes. It departs from a_2 only once, at node d, to take the shortest independent path to its target node g_1.

Consequently, the best response strategy for a_1, assuming a_2's strategy is fixed, known in advance and rational, is either **(a)** an **Independent Shortest Path:** The path $SP^1_{s_1,g_1}$ from s_1 to g_1 without cooperation, computed by a shortest-path algorithm with node weights τ^1_v; or **(b)** a **Cooperation-Assisted Path:** The path $SP^1_{s_1,c^*_1} \circ \pi_{c^*_1,d} \circ SP^1_{d,g_1}$, where c^*_1 and d are earliest cooperation and optimal departure nodes, respectively, as defined above. This path can be constructed efficiently by using the following steps:

(1) Calculate all shortest paths from s_1 to all $v \in V_C$.
(2) Identify the first cooperation node in π that a_1 can reach at a relevant time for cooperation, denoted as c^*_1.
(3) Compute the shortest paths from every node along π to g_1. In an undirected graph, this can be achieved by finding the shortest paths from g_1 to all nodes.
(4) Determine the optimal node d along π for a_1 to leave the cooperation, minimizing the path time $T_{c^*_1,g_1}(\pi_{c^*_1,d} \circ SP^1_{d,g_1}, \pi)$: $d = \arg\min_{d \in \pi_{c^*_1,g_2}} T_{c^*_1,g_1}(\pi_{c^*_1,d} \circ SP^1_{d,g_1} | \pi)$

It is straightforward to verify that this procedure operates in polynomial time.

D.1 Best Response Algorithm

Algorithm BEST RESPONSE PATH Algorithm 3 identifies the best response strategy for agent a_1, given a_2's path π. The algorithm begins by computing SIP_{s_1} (the shortest independent paths from a_1's starting node to all other nodes) and SIP_{g_1} (the shortest independent paths from all nodes to a_1's target node) and initializes the optimal cooperation starting and departure nodes to NONE [lines 1-3]. It then iterates over all nodes in π to find the first cooperation node a_1 can reach at a relevant time for cooperation, denoted as c^*_1 [lines 4-7]. If no such node exists, since no cooperation can be established, the algorithm returns the independent shortest path as the best response to π [lines 8-9]. Otherwise, it tracks the cooperation path time along $\pi_{c^*_1,g_2}$ and determines the departure node d that optimizes a_1's arrival at its target [lines 10-18]. Finally, the algorithm compares the cooperation-assisted path time with the independent shortest path time and returns the strategy that minimizes travel time [lines 19-22].

---

[2] In the context of the pseudocode, $SP_{v,u}$ denotes both the shortest path between nodes v and u and the associated path time, depending on the context.

Complexity Analysis. The algorithm begins with two executions of Dijkstra's algorithm to compute the shortest paths from s_1 and g_1 to all nodes in the graph. Each run of Dijkstra's algorithm has a time complexity of $O(|E| + |V| \log |V|)$. Thus, the total complexity for these two runs is:

$$O(2 \cdot (|E| + |V| \log |V|)) = O(|E| + |V| \log |V|)$$

The algorithm then iterates over all nodes in π twice, first to find the optimal cooperation starting node and then to determine the optimal cooperation departure node. Since $|\pi| \leq |V|$, these two iterations contribute:

$$O(2 \cdot |V|) = O(|V|)$$

Combining these, the overall complexity of the algorithm remains:

$$O(|E| + |V| \log |V|)$$

E FULL PATH JOINT STRATEGY IN EQUILIBRIUM

E.1 Proof of Theorem 1

Theorem 1

A cooperation joint strategy (π^1, π^2) constitutes a PNE if and only if all the following conditions hold:

1. The agents cooperate along exactly one cooperation segment, $\pi^S_{c_s,c_d} \in \Pi^S_{c_s,c_d}$, which is stable.
2. Both agents reach the cooperation start node c_s via Mutually-Robust non-cooperative partial paths $\pi^{1(NC)}_{s_1,c_s} \in \Pi^{NC}_{s_1,c_s}$ and $\pi^{2(NC)}_{s_2,c_s} \in \Pi^{NC}_{s_2,c_s}$.
3. The cooperation end node c_d is jointly optimal for departure: $c_d = v^1_{d^*}(\pi^2_{c_s,g_2}) = v^2_{d^*}(\pi^1_{c_s,g_1})$.
4. From c_d onward, both agents follow their shortest independent paths to their targets, SIP_{c_d,g_1} and SIP_{c_d,g_2}.
5. Cooperation weakly dominates the shortest independent path for both agents: $T(\pi^1 \mid \pi^2) \leq T(SIP_{s_1,g_1})$, $\quad T(\pi^2 \mid \pi^1) \leq T(SIP_{s_2,g_2})$.

Consequently, (π^1, π^2) can be formally expressed as follows:

$$\pi^1 = \pi^{1(NC)}_{s_1,c_s} \circ \pi^S_{c_s,c_d} \circ SIP_{c_d,g_1}, \quad \pi^2 = \pi^{2(NC)}_{s_2,c_s} \circ \pi^S_{c_s,c_d} \circ SIP_{c_d,g_2}$$

PROOF. **Direction 1: $(\pi^1, \pi^2) \neq (SIP_{s_1,g_1}, SIP_{s_2,g_2})$ is an ECJS $\implies$ (π^1, π^2) is a PNE:**
Since (π^1, π^2) is an ECJS, it can be expressed as:

$$\pi^1 = \pi^{1(NC)}_{s_1,c_s} \circ \pi^S_{c_s,c_d} \circ SIP_{c_d,g_1}, \quad \pi^2 = \pi^{2(NC)}_{s_2,c_s} \circ \pi^S_{c_s,c_d} \circ SIP_{c_d,g_2}.$$

Where $c_s, c_d \in V_C$, $\pi^{1(NC)}_{s_1,c_s} \in \Pi^{NC}_{s_1,c_s}$, $\pi^{2(NC)}_{s_2,c_s} \in \Pi^{NC}_{s_2,c_s}$, and $\pi^S_{c_s,c_d} \in \Pi^S_{c_s,c_d}$.

By Theorem 4, the optimal path for a_i, assuming π_{-i}'s is fixed, known in advance, and rational, is one of the following:

1. The shortest independent path from s_i to g_i.
2. The path $SIP_{s_i,c^*_1} \circ \pi^{-i}_{c^*_1,d} \circ SIP_{d,g_i}$, where c^*_1 is the first cooperation node along π^{-i} that a_i can reach in a relevant time for cooperation, and d is the optimal departure node for a_i along $\pi^{-i}_{c^*_1,g_i}$

Since (π^1, π^2) is an ECJS, it holds that:

$$T(\pi^i \mid \pi^{-i}) \leq T(SIP_{s_i,g_i})$$

implying that SIP_{s_i,g_i} is not the best response to π^{-i}. We therefore examine the cooperation-assisted path $SIP_{s_i,c^*_1} \circ \pi^{-i}_{c^*_1,d} \circ SIP_{d,g_i}$. Since a_{-i}'s path to c_s is non-cooperative, it follows that c_s is the first cooperation node along π^{-i} that a_i can reach in a relevant time for cooperation. Furthermore, c_d is the optimal departure node for a_i along $\pi^{-i}_{c_s,g_{-i}}$. Therefore, a_i's best response to π^{-i} is the path

$$\pi^{i^*} = SIP_{s_i,c_s} \circ \pi^{-i}_{c_s,c_d} \circ SIP_{c_d,g_i} = SIP_{s_i,c_s} \circ \pi^S_{c_s,c_d} \circ SIP_{c_d,g_i}$$

The arrival time of a_i at its target node using this path can be expressed as:

$$T(\pi^{i^*} | \pi^{-i}) = \max(T(SIP_{s_i,c_s}), T(\pi^{-i(NC)}_{s_{-i},c_s})) + \tau^2_{c_s} + D(\pi^S_{c_s,c_d} | \pi^S_{c_s,c_d}) + T(SIP_{c_d,g_i})$$

We now analyze a_i's path $\pi^i = \pi^{i(NC)}_{s_i,c_s} \circ \pi^S_{c_s,c_d} \circ SIP_{c_d,g_i}$. The arrival time of a_i using π^i is:

$$T(\pi^i | \pi^{-i}) = \max(T(\pi^{i(NC)}_{s_i,c_s}), T(\pi^{-i(NC)}_{s_{-i},c_s})) + \tau^2_{c_s} + D(\pi^S_{c_s,c_d} | \pi^S_{c_s,c_d}) + T(SIP_{c_d,g_i})$$

Since $\pi^{1(NC)}_{s_1,c_s}$ and $\pi^{2(NC)}_{s_2,c_s}$ are *Mutually-Robust Non-Cooperative Partial Paths*, there are two possible cases for $T(\pi^{i(NC)}_{s_i,c_s})$:

1. If $T(\pi^{i(NC)}_{s_i,c_s}) \leq T(\pi^{-i(NC)}_{s_{-i},c_s})$, then $T(SIP_{s_i,c_s}) \leq T(\pi^{i(NC)}_{s_i,c_s}) \leq T(\pi^{-i(NC)}_{s_{-i},c_s})$ Thus:

$$\max(T(\pi^{i(NC)}_{s_i,c_s}), T(\pi^{-i(NC)}_{s_{-i},c_s})) = \max(T(SIP_{s_i,c_s}), T(\pi^{-i(NC)}_{s_{-i},c_s})) = T(\pi^{-i(NC)}_{s_{-i},c_s})$$

Hence:

$$T(\pi^i | \pi^{-i}) = T(\pi^{i^*} | \pi^{-i})$$

(2) If $T(\pi^{i(NC)}_{s_i,c_s}) > T(\pi^{-i(NC)}_{s_{-i},c_s})$, then, since $\pi^{1(NC)}_{s_1,c_s}$ and $\pi^{2(NC)}_{s_2,c_s}$ are Mutually-Robust Non-Cooperative Partial Paths, it follows that $\pi^{i(NC)}_{s_i,c_s} = SIP_{s_i,c_s}$. Thus:

$$\max(T(\pi^{i(NC)}_{s_i,c_s}), T(\pi^{-i(NC)}_{s_{-i},c_s})) = \max(T(SIP_{s_i,c_s}), T(\pi^{-i(NC)}_{s_{-i},c_s}))$$

Hence:

$$T(\pi^i|\pi^{-i}) = T(\pi^{i^*}|\pi^{-i})$$

In both cases, a_i's path π^i serves as the best response to π^{-i}. Therefore, the joint strategy (π^1, π^2) constitutes a PNE.

Direction 2: $(\pi^1, \pi^2) \neq (SIP_{s_1,g_1}, SIP_{s_2,g_2})$ is a PNE $\implies$ (π^1, π^2) is an ECJS:

Given a Pure Nash Equilibrium (PNE) joint strategy (π^1, π^2), we assume, by contradiction, that one or more of the five conditions for an *ECJS* is violated. Each condition is analyzed individually:

(1) Condition 1: We consider both parts of Condition 1: (a) the agents cooperate along exactly one cooperation segment, and (b) this cooperation segment is stable.

(a) We assume, by contradiction, that the joint strategy (π^1, π^2) contains more than one cooperation segment. In that case, it can be expressed as:

$$\pi^1 = \pi^1_{s_1,c_{s_1}} \circ \pi^C_{c_{s_1},c_{d_1}} \circ \pi^1_{c_{d_1},c_{s_2}} \circ \pi^C_{c_{s_2},c_{d_2}} \circ \pi^1_{c_{d_2},c_{g_1}}$$
$$\pi^2 = \pi^2_{s_2,c_{s_1}} \circ \pi^C_{c_{s_1},c_{d_1}} \circ \pi^2_{c_{d_1},c_{s_2}} \circ \pi^C_{c_{s_2},c_{d_2}} \circ \pi^2_{c_{d_2},c_{g_2}}$$

However, according to Lemma 2 (Cooperation Continuity), we have:

$$T(\pi^1_{s_1,c_{d_1}} \circ \pi^2_{c_{d_1},c_{s_2}} | \pi^2_{s_2,c_{s_2}}) \leq T(\pi^1_{s_1,c_{s_2}} | \pi^2_{s_2,c_{s_2}})$$

Therefore, it follows that:

$$T(\pi^1_{s_1,c_{d_1}} \circ \pi^2_{c_{d_1},c_{s_2}} \circ \pi^1_{c_{s_2},g_1} | \pi^2) \leq T(\pi^1|\pi^2)$$

If equality holds, since the joint path $(\pi^1_{s_1,c_{d_1}} \circ \pi^2_{c_{d_1},c_{s_2}} \circ \pi^1_{c_{s_2},g_1}, \pi^2)$ involves a single continuous cooperation segment, a_1 would prefer the strategy $\pi^1_{s_1,c_{d_1}} \circ \pi^2_{c_{d_1},c_{s_2}} \circ \pi^1_{c_{s_2},g_1}$ over π^1, contradicting the assumption that (π^1, π^2) is a PNE.

(b) We assume, by contradiction, that the cooperation partial path π_{c_s,c_d} is not stable. In that case, for one of the agents, without loss of generality a_1, there exists a better departure node $c_{d'} = v^1_{d^*}(\pi_{c_s,c_d})$, where $c_{d'} \neq c_d$, along the cooperation path:

$$T(\pi_{c_s,c_{d'}} \circ SIP_{c_{d'},g_1} | \pi_{c_s,c_d}) < T(\pi_{c_s,c_d} \circ SIP_{c_d,g_1} | \pi_{c_s,c_d})$$

Therefore, it follows that:

$$T(\pi^1_{s_1,c_s} \circ \pi_{c_s,c_{d'}} \circ SIP_{c_{d'},g_1} | \pi^2) < T(\pi^1, \pi^2)$$

which contradicts the assumption that (π^1, π^2) is a PNE.

(2) Condition 2: We assume, by contradiction, that the paths of the two agents from their starting nodes to c_s are not *Mutually-Robust Non-Cooperative Partial Paths*. We consider two options:

(a) The path of one of the agents, without loss of generality a_2, to c_s, $\pi^2_{s_2,c_s}$, is not non-cooperative. In this case, there exists a cooperation node along this path, $c \in \pi^2_{s_2,c_s}$, $c \neq c_s$, that a_1 can reach in a relevant time for cooperation (if there is more than one such node, we take c to be the first one along $\pi^2_{s_2,c_s}$). From Lemma 1 (Early Cooperation), it follows that:

$$T(SIP_{s_1,c} \circ \pi^2_{c,c_s} \circ \pi^1_{c_s,g_1}, \pi^2) \leq T(SIP_{s_1,c_s} \circ \pi^1_{c_s,g_1}, \pi^2)$$

Additionally, since SIP_{s_1,c_s} is the shortest path from s_1 to c_s without involving cooperation, it follows that:

$$T(SIP_{s_1,c_s} \circ \pi^1_{c_s,g_1}, \pi^2) \leq T(\pi^1_{s_1,c_s} \circ \pi^1_{c_s,g_1}, \pi^2) = T(\pi^1_{s_1,g_1} | \pi^2_{s_2,g_2})$$

If equality holds, since the joint path $(SIP_{s_1,c} \circ \pi^2_{c,c_s} \circ \pi^1_{c_s,g_1}, \pi^2)$ starts cooperation earlier than (π^1, π^2), a_1 would prefer the strategy $SIP_{s_1,c} \circ \pi^2_{c,c_s} \circ \pi^1_{c_s,g_1}$ over π^1, contradicting the assumption that (π^1, π^2) is a PNE.

(b) One of the agents, without loss of generality a_1, can modify its path toward c_s to a new path $\pi^{1'}_{s_1,c_s}$, allowing cooperation to start earlier at time

$$t' = \max(T(\pi^{1'}_{s_1,c_s}), T(\pi^2_{s_2,c_s})) < \max(T(\pi^1_{s_1,c_s}), T(\pi^2_{s_2,c_s}))$$

The agent can then use the modified full path $\pi^{1'}_{s_1,g_1} = \pi^{1'}_{s_1,c_s} \circ \pi^1_{c_s,g_1}$ and reach its target node at an improved time:

$$T(\pi^{1'}_{s_1,g_1} | \pi^2_{s_2,g_2}) = t' + \tau^2_{c_s} + T(\pi^1_{c_s,g_1} | \pi^2_{c_s,g_2}) <$$
$$\max(T(\pi^1_{s_1,c_s}), T(\pi^2_{s_2,c_s})) + \tau^2_{c_s} + T(\pi^1_{c_s,g_1} | \pi^2_{c_s,g_2}) = T(\pi^1, \pi^2)$$

contradicting the assumption that (π^1, π^2) is a PNE.

(3) Condition 3: We assume, by contradiction, that the optimal departure node for one of the agents, without loss of generality a_1, along the other agent's path $\pi^2_{c_s,g_2}$ is $c_{d'} \neq c_d$. This implies that:

$$T(\pi^1_{s_1,c_s} \circ \pi^2_{c_s,c_{d'}} \circ SIP_{c_{d'},g_1} | \pi^2) < T(SIP_{s_1,c_s} \circ \pi^2_{c_s,c_d} \circ SIP_{c_d,g_1} | \pi^2) = T(\pi^1, \pi^2)$$

which contradicts the assumption that (π^1, π^2) is a PNE.

(4) Condition 4: We assume, by contradiction, that for one of the agents, without loss of generality a_1, the partial path $\pi^1_{c_d,g_1}$ from the departure node c_d to its target node g_1 differs from the shortest non-cooperative path SIP_{c_d,g_1}. Since (π^1, π^2) involves a single continuous cooperation segment (Condition 1), and c_d is the cooperation departure node, the joint path $(\pi^1_{c_d,g_1}, \pi^2_{c_d,g_2})$ contains no cooperation. Therefore:

$$T(\pi^1_{c_d,g_1}|\pi^2_{c_d,g_2}) = T(\pi^1_{c_d,g_1}) > T(SIP_{c_d,g_1}),$$

which implies:

$$T(\pi^1|\pi^2) > T(\pi^1_{s_1,c_d} \circ SIP_{c_d,g_1}|\pi^2),$$

contradicting the assumption that (π^1, π^2) is a PNE.

(5) Condition 5: We assume, by contradiction, that one of the agents, without loss of generality a_1, prefers to take its independent path directly from its starting node to its target node, SIP_{s_1,g_1}. In this case, the independent shortest path for a_1 would be a better response to the π^2 than π^1:

$$T(SIP_{s_1,g_1}, \pi^2) < T(\pi^1, \pi^2).$$

This contradicts the assumption that the joint strategy (π^1, π^2) is a PNE.

Therefore, it holds that a cooperation joint strategy $(\pi^1, \pi^2) \neq (SIP_{s_1,g_1}, SIP_{s_2,g_2})$ constitutes a *Pure Nash Equilibrium (PNE)* if and only if (π^1, π^2) is an *ECJS*. □

E.2 Dominance of Extended Stable Partial Paths

LEMMA 6. *Given two cooperation nodes* $c_{d_1}, c_{d_2} \in V_C$*, where* $c_{d_1} \neq c_{d_2}$*, if there exists a stable cooperation partial path from* c_{d_1} *to* c_{d_2}*, denoted* $\pi^S_{c_{d_1},c_{d_2}}$*, then for any stable cooperation partial path* $\pi_{c_s,c_{d_1}}$ *starting at* c_s *and ending at* c_{d_1}*, the concatenated path*

$$\pi'_{c_s,c_{d_2}} = \pi_{c_s,c_{d_1}} \circ \pi^S_{c_{d_1},c_{d_2}}$$

is also stable and dominates $\pi_{c_s,c_{d_1}}$*, satisfying:*

$$\begin{aligned}\forall i \in \{1,2\}, \quad & T(\pi_{c_s,c_{d_1}} \circ SIP_{c_{d_1},g_i} \mid \pi_{c_s,c_{d_1}} \circ SIP_{c_{d_1},g_{-i}}) \geq \\ & T(\pi'_{c_s,c_{d_2}} \circ SIP_{c_{d_2},g_i} \mid \pi'_{c_s,c_{d_2}} \circ SIP_{c_{d_2},g_{-i}})\end{aligned}$$

PROOF. Given a stable path $\pi_{c_s,c_{d_1}}$, since $\pi^S_{c_{d_1},c_{d_2}}$ is stable, the path $\pi'_{c_s,c_{d_2}} = \pi_{c_s,c_{d_1}} \circ \pi^S_{c_{d_1},c_{d_2}}$ is also stable.
For an agent a_i, the path time to its target node via a cooperation along $\pi_{c_s,c_{d_1}}$ can be described as:

$$T(\pi_{c_s,c_{d_1}} \circ SIP_{c_{d_1},g_i}|\pi_{c_s,c_{d_1}} \circ SIP_{c_{d_1},g_{-i}}) = T(\pi_{c_s,c_{d_1}}|\pi_{c_s,c_{d_1}}) + \tau^2_{c_{d_1}} + T(SIP_{c_{d_1},g_i}|SIP_{c_{d_1},g_{-i}})$$

Similarly, the path time to its target node via a cooperation along $\pi'_{c_s,c_{d_2}}$ is:

$$T(\pi'_{c_s,c_{d_2}} \circ SIP_{c_{d_2},g_i}|\pi'_{c_s,c_{d_2}} \circ SIP_{c_{d_2},g_{-i}}) = T(\pi_{c_s,c_{d_1}}|\pi_{c_s,c_{d_1}}) + \tau^2_{c_{d_1}} + T(\pi^S_{c_{d_1},c_{d_2}} \circ SIP_{c_{d_2},g_i}|\pi^S_{c_{d_1},c_{d_2}} \circ SIP_{c_{d_2},g_{-i}})$$

Since $\pi^S_{c_{d_1},c_{d_2}}$ is stable, it holds that:

$$T(SIP_{c_{d_1},g_i}|\pi^S_{c_{d_1},c_{d_2}} \circ SIP_{c_{d_2},g_{-i}}) \geq T(\pi^S_{c_{d_1},c_{d_2}} \circ SIP_{c_{d_2},g_i}|\pi^S_{c_{d_1},c_{d_2}} \circ SIP_{c_{d_2},g_{-i}})$$

Additionally, since c_{d_1} is the last cooperation node along $\pi_{c_s,c_{d_1}}$, $(SIP_{c_{d_1},g_i}, SIP_{c_{d_1},g_{-i}})$ involves no cooperation, and it follows that:

$$T(SIP_{c_{d_1},g_i}|SIP_{c_{d_1},g_{-i}}) \geq T(SIP_{c_{d_1},g_i}|\pi^S_{c_{d_1},c_{d_2}} \circ SIP_{c_{d_2},g_{-i}})$$

Hence,

$$T(\pi_{c_s,c_{d_1}}|\pi_{c_s,c_{d_1}}) + \tau^2_{c_{d_1}} + T(SIP_{c_{d_1},g_i}|SIP_{c_{d_1},g_{-i}}) \geq$$

$$T(\pi_{c_s,c_{d_1}}|\pi_{c_s,c_{d_1}}) + \tau^2_{c_{d_1}} + T(\pi^S_{c_{d_1},c_{d_2}} \circ SIP_{c_{d_2},g_i}|\pi^S_{c_{d_1},c_{d_2}} \circ SIP_{c_{d_2},g_{-i}})$$

which implies that:

$$T(\pi_{c_s,c_{d_1}} \circ SIP_{c_{d_1},g_i}|\pi_{c_s,c_{d_1}} \circ SIP_{c_{d_1},g_{-i}}) \geq T(\pi'_{c_s,c_{d_2}} \circ SIP_{c_{d_2},g_i}|\pi'_{c_s,c_{d_2}} \circ SIP_{c_{d_2},g_{-i}})$$

and the lemma holds. □

The following corollary follows directly from Lemma 6. Since the segment $SIP_{c_d,c_{d'}}$ is stable, appending it to the original cooperation segment preserves stability. Moreover, this extension does not violate Conditions 1, 2, 4, or 5 in Definition 4.

COROLLARY 1. *Given a cooperation joint strategy*

$$(\pi^1_{s_1,c_s} \circ \pi^S_{c_s,c_d} \circ SIP_{c_d,g_1}, \pi^2_{s_2,c_s} \circ \pi^S_{c_s,c_d} \circ SIP_{c_d,g_2})$$

that satisfies all conditions of Definition 4 except Condition 3, i.e., for some agent a_i, the optimal departure node along $\pi^S_{c_s,c_d} \circ SIP_{c_d,g_{-i}}$ is $c_{d'} \neq c_d$:

$$c_{d'} = v^i_{d^*}(\pi^S_{c_s,c_d} \circ SIP_{c_d,g_{-i}}) \neq c_d$$

then the joint strategy

$$(\pi^1_{s_1,c_s} \circ \pi^S_{c_s,c_d} \circ SIP_{c_d,c_{d'}} \circ SIP_{c_{d'},g_1}, \pi^2_{s_2,c_s} \circ \pi^S_{c_s,c_d} \circ SIP_{c_d,c_{d'}} \circ SIP_{c_{d'},g_2})$$

satisfies the same conditions of Definition 4 and dominates the original joint strategy.

E.3 Proof of Theorem 2

We aim to show that in any scenario, there always exists a joint strategy that constitutes a PNE. If the independent joint strategy $(SIP_{s_1,g_1}, SIP_{s_2,g_2})$ is already a PNE, the claim holds. Otherwise, one of the agents necessarily has an incentive to deviate from its independent shortest path in favor of cooperation with the other agent. In this case, we show that the resulting cooperative path is stable and can serve as the basis for constructing a Pure Nash Equilibrium.

As a first step, we show that if there exists a cooperative joint strategy with a stable cooperation segment that both agents prefer over the independent strategy, then a PNE exists.

LEMMA 7. *Consider a cooperative joint strategy of the form:*

$$\Big(SIP_{s_1,c_s} \circ \pi^S_{c_s,c_d} \circ SIP_{c_d,g_1},\ SIP_{s_2,c_s} \circ \pi^S_{c_s,c_d} \circ SIP_{c_d,g_2}\Big),$$

where c_s is the first cooperation node and c_d is the last cooperation node in which the agents cooperate, and $\pi^S_{c_s,c_d} \in \Pi_{c_s,c_d}$ is a stable partial path. If the following condition holds:

$$\forall i \in \{1,2\}, \quad T\Big(SIP_{s_i,c_s} \circ \pi^S_{c_s,c_d} \circ SIP_{c_d,g_i} \,|\, SIP_{s_{-i},c_s} \circ \pi^S_{c_s,c_d} \circ SIP_{c_d,g_{-i}}\Big) \leq T\Big(SIP_{s_i,g_i}\Big)$$

then a Pure Nash Equilibrium (PNE) exists.

PROOF. Following Theorem 1, to show that a cooperative joint strategy constitutes a PNE, it is sufficient to show that it is an ECJS (by verifying the five conditions of Definition 4).

Considering the joint strategy:

$$\Big(SIP_{s_1,c_s} \circ \pi^S_{c_s,c_d} \circ SIP_{c_d,g_1},\ SIP_{s_2,c_s} \circ \pi^S_{c_s,c_d} \circ SIP_{c_d,g_2}\Big)$$

We show that this joint strategy is either already an ECJS or can serve as the foundation for constructing one. The cooperation segment $\pi^S_{c_s,c_d}$ is stable by the Lemma's assumption, satisfying **Condition 1**. Additionally, since both agents follow their respective shortest independent paths from c_d to their target nodes, **Condition 4** is satisfied. Furthermore, because both agents prefer this cooperative joint strategy over their independent shortest-path strategy, **Condition 5** also holds.

We show that if Conditions 2 and 3 do not hold, the cooperation segment $\pi^S_{c_s,c_d}$ can be extended to satisfy them without violating the other conditions.

Condition 2: If the paths SIP_{s_1,c_s} and SIP_{s_2,c_s} toward the cooperation starting node are non-cooperative, then, since neither agent can shorten its path to c_s, these paths are mutually robust, satisfying the condition. However, if one agent (without loss of generality, a_1) can reach a cooperation node $c_e \in SIP_{s_2,c_s}$ at a time relevant for cooperation, we show that the partial path $\pi_{c_e,c_d} = SIP_{c_e,c_s} \circ \pi^S_{c_s,c_d}$ is also stable, and the resulting joint strategy

$$(SIP_{s_1,c_e} \circ \pi_{c_e,c_d} \circ SIP_{c_d,g_1},\ SIP_{s_2,c_e} \circ \pi_{c_e,c_d} \circ SIP_{c_d,g_2})$$

continues to satisfy Conditions 1, 4, and 5.

We assume, by contradiction, that for one of the agents, without loss of generality a_1, $v^1_{d^*}(\pi_{c_e,c_d}) \neq c_d$, and denote the optimal departure node as $c_{e'}$. Then:

$$T(SIP_{s_1,c_e} \circ \pi_{c_e,c_{e'}} \circ SIP_{c_{e'},g_1} \mid SIP_{s_2,c_e} \circ \pi_{c_e,c_d} \circ SIP_{c_d,g_2}) < T(SIP_{s_1,c_e} \circ \pi_{c_e,c_d} \circ SIP_{c_d,g_1} \mid SIP_{s_2,c_e} \circ \pi_{c_e,c_d} \circ SIP_{c_d,g_2})$$

Since both agents start cooperating at c_e, the above inequality can be rewritten as:

$$\max(T(SIP_{s_1,c_e} \mid SIP_{s_2,c_e}), T(SIP_{s_2,c_e} \mid SIP_{s_1,c_e})) + \tau^2_{c_e} + D(\pi_{c_e,c_{e'}} \mid \pi_{c_e,c_{e'}}) + T(SIP_{c_{e'},g_1} \mid \pi_{c_{e'},c_d} \circ SIP_{c_d,g_2})$$

$$< \max(T(SIP_{s_1,c_e} \mid SIP_{s_2,c_e}), T(SIP_{s_2,c_e} \mid SIP_{s_1,c_e})) + \tau^2_{c_e} + D(\pi_{c_e,c_{e'}} \mid \pi_{c_e,c_{e'}}) + T(\pi_{c_{e'},c_d} \circ SIP_{c_d,g_1} \mid \pi_{c_{e'},c_d} \circ SIP_{c_d,g_2})$$

Simplifying:

$$T(SIP_{c_{e'},g_1} \mid \pi_{c_{e'},c_d} \circ SIP_{c_d,g_2}) < T(\pi_{c_{e'},c_d} \circ SIP_{c_d,g_1} \mid \pi_{c_{e'},c_d} \circ SIP_{c_d,g_2})$$

Adding the arrival time of a_1 at $c_{e'}$ (without cooperation) to both sides of the in-equation:

$$D(SIP_{s_1,c_{e'}}) + T(SIP_{c_{e'},g_1} \mid \pi_{c_{e'},c_d} \circ SIP_{c_d,g_2}) < D(SIP_{s_1,c_{e'}}) + T(\pi_{c_{e'},c_d} \circ SIP_{c_d,g_1} \mid \pi_{c_{e'},c_d} \circ SIP_{c_d,g_2})$$

Since $c_{e'}$ is the optimal departure node for a_1, $(SIP_{c_{e'},g_1}, \pi_{c_{e'},c_d} \circ SIP_{c_d,g_2})$ does not involve cooperation. Moreover, as SIP_{s_1,g_1} denotes the shortest path to g_1 without cooperation:

$$D(SIP_{s_1,c_{e'}}) + T(SIP_{c_{e'},g_1} \mid \pi_{c_{e'},c_d} \circ SIP_{c_d,g_2}) = D(SIP_{s_1,c_{e'}}) + T(SIP_{c_{e'},g_1}) \geq T(SIP_{s_1,g_1})$$

Thus:

$$D(SIP_{s_1,c_{e'}}) + T(\pi_{c_{e'},c_d} \circ SIP_{c_d,g_1} \mid \pi_{c_{e'},c_d} \circ SIP_{c_d,g_2}) > T(SIP_{s_1,g_1})$$

Since the path time $T(\pi_{c_{e'},c_d} \circ SIP_{c_d,g_1} \mid \pi_{c_{e'},c_d} \circ SIP_{c_d,g_2})$ assumes cooperation along $\pi_{c_{e'},c_s}$, it follows that:

$$T(SIP_{s_1,c_s} \circ \pi^S_{c_s,c_d} \circ SIP_{c_d,g_1} \mid SIP_{s_2,c_s} \circ \pi^S_{c_s,c_d} \circ SIP_{c_d,g_2}) \geq D(SIP_{s_1,c_{e'}}) + T(\pi_{c_{e'},c_d} \circ SIP_{c_d,g_1} \mid \pi_{c_{e'},c_d} \circ SIP_{c_d,g_2}) > T(SIP_{s_1,g_1})$$

contradicting our assumption. Thus, we conclude that $\pi_{c_e,c_d} = SIP_{c_e,c_s} \circ \pi^S_{c_s,c_d}$ constitutes a *Stable Cooperation Partial Path*. Since the joint strategy $\left(SIP_{s_1,c_e} \circ \pi_{c_e,c_d} \circ SIP_{c_d,g_1},\ SIP_{s_2,c_e} \circ \pi_{c_e,c_d} \circ SIP_{c_d,g_2}\right)$ initiates cooperation at node c_e, which precedes c_s, it dominates the original joint strategy and therefore preserves the conditions of Definition 4 that were satisfied by the original joint strategy.

Since the set of cooperation nodes is finite, the cooperation segment can be repeatedly extended until reaching a cooperation starting node $c_s \in V_C$ for which the paths SIP_{s_1,c_s} and SIP_{s_2,c_s} are mutually robust and non-cooperative, satisfying Condition 2.

Condition 3: By Corollary 1, if there exists a cooperation joint strategy that satisfies all conditions of Definition 4 except Condition 3 (i.e., one of the agents, a_i, prefers to continue cooperating with a_{-i} beyond c_d along the path to a_{-i}'s target node), then a cooperation joint strategy extending this cooperation satisfies the same conditions of Definition 4 and dominates the original one.

Since the set of cooperation nodes is finite, the cooperation segment can be repeatedly extended until reaching a cooperation ending node $c_{d'} \in V_C$ that satisfies Condition 3.

Thus, the joint strategy

$$\left(SIP_{s_1,c_{s'}} \circ SCP^S_{c_{s'},c_{d'}} \circ SIP_{c_{d'}}, \quad SIP_{s_2,c_{s'}} \circ SCP^S_{c_{s'},c_{d'}} \circ SCP_{c_{d'}}\right)$$

where $c_{s'}$ is the cooperation starting node that satisfies Condition 2, found by repeatedly extending the cooperation segment from the beginning, and $c_{d'}$ is the cooperation ending node that satisfies Condition 3, found by repeatedly extending the segment from the end, is an ECJS and, by Theorem 1, constitutes a PNE. □

Leveraging Lemma 7, we establish the existence of a PNE in any scenario.

Theorem 2.
In any setting involving two self-interested agents and multiple cooperation nodes ($m > 1$), at least one Pure Nash Equilibrium (PNE) joint strategy exists.

PROOF. Consider the *Independent Joint Strategy* $(SIP_{s_1,g_1}, SIP_{s_2,g_2})$. If this strategy is already a PNE, the claim follows directly. Otherwise, at least one agent, without loss of generality, a_1, has an incentive to deviate and leverage cooperation with a_2.

By Theorem 4, given that a_2 follows its independent shortest path SIP_{s_2,g_2}, the optimal path for a_1 adheres the structure $SIP_{s_1,c_s} \circ SIP_{c_s,c_d} \circ SIP_{c_d,g_1}$, where c_s is the first cooperation node along a_2's path that a_1 can reach in time for cooperation, and c_d is its optimal departure node. Since c_d lies on the shortest independent path from c_s to g_2, it is also a_2's optimal departure node along SIP_{c_s,c_d}. Thus, SIP_{c_s,c_d} forms a *Stable Cooperation Partial Path*.

Since this strategy is a_1's best response to a_2 shortest independent path, we have:

$$T(SIP_{s_1,c_s} \circ SIP_{c_s,c_d} \circ SIP_{c_d,g_1} \mid SIP_{s_2,c_s} \circ SIP_{c_s,c_d} \circ SIP_{c_d,g_2}) \leq T(SIP_{s_1,g_1})$$

Additionally, since it involves cooperation along a_2's shortest independent path, it also improves a_2's arrival time at its target node:

$$T(SIP_{s_2,c_s} \circ SIP_{c_s,c_d} \circ SIP_{c_d,g_2} \mid SIP_{s_1,c_s} \circ SIP_{c_s,c_d} \circ SIP_{c_d,g_1}) \leq T(SIP_{s_2,g_2})$$

By Lemma 7, this implies the existence of a PNE. □

F OPTIMAL STABLE JOINT STRATEGY ENDING COOPERATION AT A GIVEN NODE

Given a cooperation-ending node c_d, Algorithm 4 computes the stable joint strategy ending cooperation at c_d that dominates all other stable joint strategies with the same cooperation-ending node.

The algorithm first computes the shortest independent paths from each agent's starting node to all nodes in the graph, as well as from all nodes to each agent's respective target node [lines 1-4], It then computes the shortest stable partial paths that end cooperation at c_d from all nodes in the graph using Algorithm 2 [lines 5], and the shortest non-cooperative partial paths from each agent's starting node to all nodes in the graph using Algorithm 1 [lines 6-7].

Next, for each cooperation node $c_s \in V_C$, the algorithm evaluates the agents' arrival times at c_d under the joint strategy

$$\left(SIP^{NC}_{s_1,c_s} \circ SCP^S_{c_s,c_d} \circ SIP_{c_d,g_1},\ SIP^{NC}_{s_2,c_s} \circ SCP^S_{c_s,c_d} \circ SIP_{c_d,g_2}\right),$$

and records the cooperation-starting node that minimizes this arrival time [lines 8-12].

Finally, if the resulting stable cooperative joint strategy ending cooperation at c_d improves both agents' performance relative to following their shortest independent paths to their respective target nodes, it is returned as the optimal stable joint strategy ending cooperation at c_d [lines 13-14. Otherwise, the algorithm returns `None` [line 15].

Lemma 3.
Algorithm 4 computes, in polynomial time, the optimal stable joint strategy ending cooperation at c_d, which dominates all other stable joint strategies ending cooperation at c_d.

The correctness of Lemma 3 follows from the observation that, if there exists a stable joint strategy that ends cooperation at c_d, then the joint strategy

$$(\pi^1, \pi^2) = \left(SIP^{NC}_{s_1,c_s} \circ SCP^{S}_{c_s,c_d} \circ SIP_{c_d,g_1},\ SIP^{NC}_{s_2,c_s} \circ SCP^{S}_{c_s,c_d} \circ SIP_{c_d,g_2}\right),$$

where

$$c_s = \arg\min_{c \in V_C} \left\{ \max\left(T(SIP^{NC}_{s_1,c}), T(SIP^{NC}_{s_2,c})\right) + \tau_c^2 + T\left(SCP^{S}_{c,c_d} \mid SCP^{S}_{c,c_d}\right) \right\},$$

dominates all other stable joint strategies that end cooperation at c_d.

The algorithm iterates over all cooperation nodes, evaluating each as a potential starting node and selecting the one that minimizes the arrival time at c_d.

Notably, if a beneficial cooperation opportunity exists for one of the agents along the departure segment, thereby violating Condition 3 of Definition 4, the resulting joint strategy does not constitute a full-path ECJS. However, as shown in Lemma 6, such strategies are disregarded by Algorithm 5 when evaluating all cooperation nodes as potential departure nodes.

Complexity Analysis. The algorithm begins with four executions of Dijkstra's algorithm to determine the shortest paths from s_1, s_2, g_1, and g_2. Each execution of Dijkstra's algorithm has a time complexity of $O(|E| + |V| \log |V|)$. Consequently, the total complexity for these four runs is:

$$O(4 \cdot (|E| + |V| \log |V|)) = O(|E| + |V| \log |V|)$$

Next, the algorithm runs Algorithm 2, which operates similarly to Dijkstra's algorithm, with a complexity of:

$$O(|E| + |V| \log |V|)$$

Subsequently, the algorithm executes Algorithm 1 twice, each with the same complexity:

$$2 \cdot O(|E| + |V| \log |V|) = O(|E| + |V| \log |V|)$$

Finally, the algorithm iterates over all cooperation nodes in the graph to find the one that optimizes the Nash equilibrium overall path time. This iteration has a complexity of:

$$O(|V|)$$

Combining all these components, the total complexity of the algorithm can be expressed as:

$$O(|E| + |V| \log |V| + |V|)$$

Which concludes to:

$$O(|E| + |V| \log |V|)$$

G OPTIMAL ECJS

The algorithm begins by initializing a dictionary that maps each cooperation node in the graph to its optimal stable joint strategy ending cooperation at that node, using Algorithm 4. The dictionary is initialized with the independent shortest-paths joint strategy, and an empty set is initialized to track cooperation nodes that admit a stable cooperation partial path to another cooperation node [line 1]. The algorithm then iterates over all cooperation nodes $c_d \in V_C$, excluding nodes that already have a stable cooperation path to another cooperation node. For each node, it determines the optimal stable joint strategy ending cooperation at that node using Algorithm 4, and updates the list of dominated nodes with those that admit a stable cooperation partial path to c_d [lines 2-6]. Next, the algorithm verifies whether the independent shortest paths joint strategy constitutes a pure Nash equilibrium (PNE). If it does not, the strategy is removed [lines 9-10]. Finally, the algorithm removes all joint strategies that terminate cooperation at nodes dominated by others (i.e., nodes with stable cooperation partial paths to another node) and returns the set of all non-dominated ECJS [line 1].

Theorem 3.
Algorithm 5 returns a set of joint strategies satisfying the following properties:

(1) **Nash Equilibrium Guarantee:** Every joint strategy (π^1, π^2) returned by the algorithm is a PNE.
(2) **Dominance:** For every joint strategy $(\pi^1, \pi^2) \in \mathbb{ECJS}$, the algorithm returns a joint strategy (π'^1, π'^2) that dominates it.

PROOF. **Nash Equilibrium Guarantee:**
Every joint strategy (π^1, π^2) that Algorithm 5 returns is either the shortest independent joint strategy or a cooperation joint strategy returned by Algorithm 4. If $(\pi^1, \pi^2) = (SIP_{s_1,g_1}, SIP_{s_2,g_2})$, the algorithm explicitly verifies that it is a PNE in lines 9-13. Otherwise, if $(\pi^1, \pi^2) \neq (SIP_{s_1,g_1}, SIP_{s_2,g_2})$ and is returned by Algorithm 4, it may fail to be a PNE only if the shortest independent path from its cooperation ending node c_d to one of the agents' target nodes g_i contains a stable cooperation partial path, violating Condition 3 for all cooperation joint strategies

that end cooperation at c_d. However, since the algorithm returns only joint strategies that end cooperation at nodes without stable cooperation partial paths to other nodes in the graph, this scenario cannot occur. Consequently, all conditions of Definition 4 are satisfied, implying that $(\pi^1, \pi^2) \in ECJS = PNE$. **Dominance:**

If $(\pi^1, \pi^2) = (SIP_{s_1,g_1}, SIP_{s_2,g_2})$, then the algorithm implicitly adds this joint strategy to its result set. Otherwise, if $(\pi^1, \pi^2) \neq (SIP_{s_1,g_1}, SIP_{s_2,g_2})$, then by Theorem 1, $(\pi^1, \pi^2) \in \mathbb{ECJS}$ and can be represented as:

$$\pi^1 = \pi^{1(NC)}_{s_1,c_s} \circ \pi^S_{c_s,c_d} \circ SIP_{c_d,g_1}, \quad \pi^2 = \pi^{2(NC)}_{s_2,c_s} \circ \pi^S_{c_s,c_d} \circ SIP_{c_d,g_2}$$

where c_s is the cooperation starting node and c_d is the cooperation ending node. This joint strategy is dominated by the optimal joint strategy in $\mathbb{ECJS}^{c_d}$:

$$(\pi'^1, \pi'^2) = \left(SIP^{NC}_{s_1,c_s} \circ SCP^S_{c_s,c_d} \circ SIP_{c_d,g_1},\ SIP^{NC}_{s_2,c_s} \circ SCP^S_{c_s,c_d} \circ SIP_{c_d,g_2}\right)$$

If there is no stable cooperation partial path from c_d to any other cooperation node in the graph, the algorithm uses Algorithm 4 to identify (π'^1, π'^2), ensuring that (π'^1, π'^2) is returned. If, however, there exists a cooperation node $c_{d'}$ such that a stable cooperation partial path $\pi^S_{c_d,c_{d'}}$ exists (among multiple such nodes, we select $c_{d'}$ as one that has no stable cooperation partial path leading to any other cooperation node in the graph), then by Lemma 6, the following joint strategy:

$$(\pi''^1, \pi''^2) = \left(SIP^{NC}_{s_1,c_s} \circ SCP^S_{c_s,c_d} \circ \pi^S_{c_d,c_{d'}} \circ SIP_{c_{d'},g_1},\ SIP^{NC}_{s_2,c_s} \circ SCP^S_{c_s,c_d} \circ \pi^S_{c_d,c_{d'}} \circ SIP_{c_{d'},g_2}\right)$$

dominates (π'^1, π'^2). Since $(\pi''^1, \pi''^2) \in \mathbb{ECJS}^{c_{d'}}$ and the algorithm returns the optimal joint strategy in $\mathbb{ECJS}^{c_{d'}}$, which dominates all other joint strategies in $\mathbb{ECJS}^{c_{d'}}$, it follows that the returned strategy also dominates (π''^1, π''^2), which in turn dominates (π^1, π^2).

Consequently, for any given joint strategy $(\pi^1, \pi^2) \in PNE$, Algorithm 5 either returns (π^1, π^2) or an equilibrium cooperation joint strategy (π'^1, π'^2) that dominates it. □

EXPERIMENTS AND RESULTS

H SOCIAL WELFARE OPTIMIZATION

We aim to determine the joint strategy (π^{1^*}, π^{2^*}) for agents a_1 and a_2 that minimizes their total path time:

$$(\pi^{1^*}, \pi^{2^*}) = \arg\min_{\pi^1,\pi^2} \left[T_{g_1}(\pi^1|\pi^2) + T_{g_2}(\pi^2|\pi^1)\right]$$

Considering $V_C = \{c_1, \ldots, c_m\}$ for some $m > 0$, a naive approach might involve evaluating all combinations of possible cooperation paths. However, Lemma 2 provides key insights into the structure of the optimal social welfare joint strategy, significantly streamlining the search process. According to Lemma 2, the most efficient cooperation path, minimizing the total path time between some initial cooperation node (denoted with c_s) and a final cooperation node (denoted with c_d) for both agents, involves continuous cooperation along the same path SCP_{c_s,c_d}. The joint strategy that optimizes social welfare is therefore:

$$(\pi^{1^*}, \pi^{2^*}) = (SIP_{s_1,c_s} \circ SCP_{c_s,c_d} \circ SIP_{c_d,g_1}, SIP_{s_2,c_s} \circ SCP_{c_s,c_d} \circ SIP_{c_d,g_2})$$

Thus, our objective is to identify the cooperation starting node c_s and the cooperation departure node c_d that optimize social welfare (see illustration in Figure 9):

$$\begin{aligned} c_s, c_d = \arg\min_{c_s,c_d} \Big[& \\ & T_{g_1}(SIP_{s_1,c_s} \circ SCP_{c_s,c_d} \circ SIP_{c_d,g_1} | SIP_{s_2,c_s} \circ SCP_{c_s,c_d} \circ SIP_{c_d,g_2}) + \\ & T_{g_2}(SIP_{s_2,c_s} \circ SCP_{c_s,c_d} \circ SIP_{c_d,g_2} | SIP_{s_1,c_s} \circ SCP_{c_s,c_d} \circ SIP_{c_d,g_1}) \\ & \Big] \end{aligned}$$

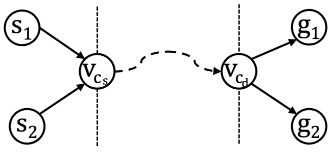


Figure 9: Social welfare optimization occurs when both agents follow the shortest path from their initial nodes to a cooperation node v_{c_s}, then continuously cooperate along a shared path to the final cooperation node v_{c_d}. After reaching v_{c_d}, the agents depart and take the shortest paths to their respective target nodes.

Using a shortest path algorithm from g_1 and g_2, we can determine the shortest path time from each cooperation node $c_i \in V_C$ to g_1, denoted as SIP_{c_i,g_1}, and to g_2, denoted as SIP_{c_i,g_2}. The total path time of these paths, $T_{c_i,g_1}(SIP_{c_i,g_1}) + T_{c_i,g_2}(SIP_{c_i,g_2})$, is referred to as the *departure value* of node c_i and is denoted by $d^*(c_i)$:

$$d^*(c_i) = T_{c_i,g_1}(SIP_{c_i g_1}) + T_{c_i,g_2}(SIP_{c_i,g_2})$$

To find the *optimal departure node* for a cooperation that starts at a given node c_s, we identify the departure node that minimizes the combined path time of reaching it cooperatively from c_s and subsequently reaching the target nodes from it:

$$v_d(c_s) = \arg\min_{c\in V_C} \left(2\cdot \left(T_{c_s,c}(SCP_{c_s,c}|SCP_{c_s,c}) + \tau_c^2\right) + d^*(c)\right)$$

However, determining the optimal node to initiate cooperation, c_s, may still require evaluating all potential cooperation nodes. While Lemma 1 demonstrates that starting cooperation earlier along a specific path generally results in a shorter overall path time, it cannot be applied to globally compare cooperation nodes, as different nodes may result in distinct subsequent paths. Consequently, a cooperation that begins later but follows a different path may achieve a better overall path time than one that begins earlier (see example in Figure 10).

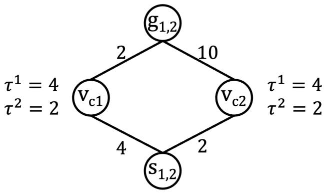


Figure 10: Although a_1 and a_2 can begin cooperation at v_{c_2} earlier than at v_{c_1}, the optimal social welfare is achieved when the cooperation starts at v_{c_1}.

To improve the performance of finding the optimal cooperation starting node c_s we propose a pruning approach that leverages Lemma 1 to reduce the number of evaluated cooperation nodes while ensuring all possible cooperation paths are considered.

Using a shortest path algorithm from s_1 and s_2, we can determine the shortest path time to each cooperation node $c_i \in V_C$, denoted as SIP_{s_1,c_i} and SIP_{s_2,c_i}, respectively. We denote the *earliest cooperation time* of node c_i as $t^*(c_i)$, which represents the earliest shared departure time for two agents starting to cooperate at this node:

$$t^*(c_i) = \max\left(T_{s_1,c_i}(SIP_{s_1,c_i}), T_{s_2,c_i}(SIP_{s_2,c_i})\right) + \tau_{c_i}^2$$

We sort the cooperation nodes in ascending order based on their *earliest cooperation times* . For each potential cooperation starting node c_i, we determine its associated *optimal departure node* $v_{d_i} = v_d(c_i)$ and calculate the social welfare of the path dictated by the two nodes:

$$2\cdot\left(t^*(c_i) + T_{c_i,d_i}(SCP_{c_i,d_i}|SCP_{c_i,d_i}) + \tau_{d_i}^2\right) + d^*(v_{d_i})$$

With the goal of finding the optimal cooperation starting node c_s that minimizes social welfare:

$$c_s = \arg\min_{c_i\in V_C}\left(2\cdot\left(t^*(c_i)+T_{c_i,d_i}(SCP_{c_i,d_i}|SCP_{c_i,d_i}) + \tau_{v_{d_i}}^2\right) + d^*(v_{d_i})\right)$$

Since evaluating a cooperation starting node c_i as a starting node involves finding the shortest cooperative path to every other cooperation node $c_j \in V_C$, we compare the arrival time at c_j through c_i with $t^*(c_j)$. If

$$t^*(c_j) \ge t^*(c_i) + D_{c_i,c_j}(SCP_{c_i,c_j}|SCP_{c_i,c_j})$$

then we can prune c_j and exclude it as a potential cooperation starting node, thereby reducing the number of nodes to be evaluated (see example in Figure 11).

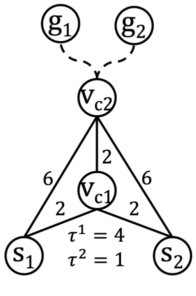


Figure 11: Although a_1 and a_2 can individually reach v_{c_2} by $t = 6$, starting cooperation earlier at v_{c_1} allows them to reach v_{c_2} by $t = 5$, improving path efficiency. As a result, v_{c_2} can be excluded from the set of potential cooperation starting nodes, as initiating cooperation earlier at v_{c_1} yields a more efficient outcome.

Algorithm 6 is designed to identify the optimal joint strategy (π_1^*, π_2^*) that minimizes social welfare for two agents navigating a graph with m cooperation nodes. The algorithm starts by finding SIP_{s_1} and SIP_{s_2} (all shortest non-cooperative paths from the initial nodes to other graph nodes), and SIP_{g_1} and SIP_{g_2} (all shortest non-cooperative paths to the target nodes from other graph nodes) [lines 1-4]. These paths are then used to compute the *departure value* $d^*(c)$ and the *earliest cooperation time* $t^*(c)$ for each cooperation node $c \in V_C$ [lines 5-8]. The algorithm iterates over all cooperation nodes $c_s \in V_C$, in ascending order of their t^* values, evaluating each as a potential cooperation starting node within a continuous cooperation joint strategy $(SIP_{s_1,c_s} \circ SCP_{c_s,c_d} \circ SIP_{c_d,g_1}, SIP_{s_2,c_s} \circ SCP_{c_s,c_d} \circ SIP_{c_d,g_2})$. It first computes the shortest paths from the selected node to all other nodes in the graph, assuming cooperation at all nodes. Then, it evaluates the remaining cooperation nodes as potential cooperation departure nodes by calculating the total path time for both agents when these nodes are designated as the cooperation endpoints [lines 9-26]. If a cooperation node can be reached earlier through cooperation than individually, it is pruned from consideration as a cooperation starting node [lines 16-17]. Finally, the algorithm returns the social welfare optimal joint strategy [line 27].

Algorithm 6 SOCIAL WELFARE OPTIMAL PATH$(G, V_C, s_1, s_2, g_1, g_2)$

```
 1: $SIP_{s_1} \leftarrow$ ALL SHORTEST PATHS FROM$(G, \tau^1, s_1)$          ▹ Find shortest paths from starting nodes
 2: $SIP_{s_2} \leftarrow$ ALL SHORTEST PATHS FROM$(G, \tau^1, s_2)$
 3: $SIP_{g_1} \leftarrow$ ALL SHORTEST PATHS TO$(G, \tau^1, g_1)$            ▹ Find shortest paths to target nodes
 4: $SIP_{g_2} \leftarrow$ ALL SHORTEST PATHS TO$(G, \tau^1, g_2)$
 5: for all $c \in V_C$ do                                                  ▹ Determine $t^*$ and $d^*$ for all node
 6:     $t_c^* \leftarrow \max(SIP_{s_1,c}, SIP_{s_2,c}) + \tau_c^2$
 7:     $d_c^* \leftarrow SIP_{g_1,c} + SIP_{g_2,c}$
 8: $coopStartNodes \leftarrow$ SORT$(V_C$, by ascending $t^*)$              ▹ Sort cooperation nodes by $t^*$
 9: $\pi_1^* \leftarrow SIP_{s_1,g_1}, \pi_2^* \leftarrow SIP_{s_2,g_2}$
10: $minimalJointPathTime \leftarrow SIP_{s_1,g_1} + SIP_{s_2,g_2}$
11: while $coopStartNodes \neq \emptyset$ do
12:     $c_s \leftarrow coopStartNodes$.POP()                                ▹ Select a starting node
13:     $SCP_{c_s} \leftarrow$ ALL SHORTEST PATHS FROM$(G, \tau^2, c_s)$      ▹ Find shortest paths from $c_s$
14:     for all $c \in V_C$ do
15:         if $t_{c_s}^* + SCP_{c_s,c} + \tau_c^2 \leq t_c^*$ then
16:             $coopStartNodes$.REMOVE$(c)$                             ▹ Prune non-optimal cooperation starting nodes
17:             $totalPathTime \leftarrow 2 \cdot (t_{c_s}^* + SCP_{c_s,c} + \tau_c^2) + d_c^*$
18:             if $totalPathTime < minimalJointPathTime$ then             ▹ Update optimal joint path time
19:                 $minimalJointPathTime \leftarrow totalPathTime$
20:                 $\pi_1^* \leftarrow SIP_{s_1,c_s} \circ SCP_{c_s,c} \circ SIP_{c,g_1}$
21:                 $\pi_2^* \leftarrow SIP_{s_2,c_s} \circ SCP_{c_s,c} \circ SIP_{c,g_2}$
22: return $(\pi_1^*, \pi_2^*)$                                              ▹ Return social welfare optimal path
```

To establish the optimality of the algorithm, we verify two key lemmas, analogous to Lemmas 1 and 2 presented in the Fixed Strategy Assumption scenario:

(1) **Cooperation Continuity:** The joint strategy that optimizes social welfare through cooperation is one in which a_1 and a_2 independently follow the shortest paths to a cooperation starting node c_s $(SIP_{s_1,c_s}, SIP_{s_2,c_s})$, cooperate along the joint shortest path to a cooperation departure node c_d (SCP_{c_s,c_d}), and then each independently follows the shortest path to their respective target nodes $(SIP_{c_d,g_1}, SIP_{c_d,g_2})$.

(2) **Early Cooperation:** If agents a_1 and a_2 can reach a cooperation node $c \in V_C$ earlier through cooperation rather than individually, then c is not the first cooperation node in the joint strategy that optimizes social welfare.

LEMMA 8 (COOPERATION CONTINUITY). *Let $\pi^1 \in \Pi_{s_1,g_1}$ and $\pi^2 \in \Pi_{s_2,g_2}$ be two paths that begin cooperation at node c_s and finish the cooperation at node c_d. Then, it holds that:*

$$T_{g_1}(\pi'^1|\pi'^2) + T_{g_2}(\pi'^2|\pi'^1) \leq T_{g_1}(\pi^1|\pi^2) + T_{g_2}(\pi^2|\pi^1)$$

Where:

$$\pi'^1 = SIP_{s_1,c_s} \circ SCP_{c_s,c_d} \circ SIP_{c_d,g_1}$$
$$\pi'^2 = SIP_{s_2,c_s} \circ SCP_{c_s,c_d} \circ SIP_{c_d,g_2}$$

PROOF. Since a_1 and a_2 cooperate at nodes c_s and c_d, the social welfare induced by π'^1 and π'^2 is given by:

$$T_{g_1}(\pi'^1|\pi'^2) + T_{g_2}(\pi'^2|\pi'^1) =$$
$$2 \cdot \left(\max\left(T_{s_1,c_s}(SIP_{s_1,c_s}), T_{s_2,c_s}(SIP_{s_2,c_s})\right) + \tau_{c_s}^2 + T_{c_s,c_d}(SCP_{c_s,c_d}|SCP_{c_s,c_d}) + \tau_{c_d}^2\right)$$
$$+T_{c_d,g_1}(SIP_{c_d,g_1}) + T_{c_d,g_2}(SIP_{c_d,g_2})$$

The social welfare induced by π^1 and π^2 is:

$$T_{g_1}(\pi^1|\pi^2) + T_{g_2}(\pi^2|\pi^1) =$$
$$2 \cdot \left(\max\left(T_{s_1,c_s}(\pi^1), T_{s_2,c_s}(\pi^2)\right) + \tau_{c_s}^2 + \max(T_{c_s,c_d}(\pi^1|\pi^2), T_{c_s,c_d}(\pi^2|\pi^1)) + \tau_{c_d}^2\right)$$

$$+T_{c_d,g_1}(\pi^1) + T_{c_d,g_2}(\pi^2)$$

Since the execution time for a_1 at each node v visited by π_1 before c_s or after c_d is τ_i^1, Then by the definitions of SIP_{s_1,c_s} and SIP_{c_d,g_1} we have:

$$T_{s_1,c_s}(SIP_{s_1,c_s}) \leq T_{s_1,c_s}(\pi^1)$$

$$T_{c_d,g_1}(SIP_{c_d,g_1}) \leq T_{c_d,g_1}(\pi^1)$$

Similarly:

$$T_{s_2,c_s}(SIP_{s_2,c_s}) \leq T_{s_2,c_s}(\pi^2)$$

$$T_{c_d,g_2}(SIP_{c_d,g_2}) \leq T_{c_d,g_2}(\pi^2)$$

Therefore:

$$\max\left(T_{s_1,c_s}(SIP_{s_1,c_s}), T_{s_2,c_s}(SIP_{s_2,c_s})\right) \leq \max\left(T_{s_1,c_s}(\pi^1|\pi^2), T_{s_2,c_s}(\pi^2|\pi^1)\right)$$

Additionally, since a_1 and a_2 depart from node c_s simultaneously, traveling together ensures they arrive at subsequent cooperation nodes at the same time, enabling them to cooperate at those nodes. Therefore, by the definition of SCP_{c_s,c_d}:

$$T_{c_s,c_d}(SCP_{c_s,c_d}|SCP_{c_s,c_d}) \leq T_{c_s,c_d}(\pi^1|\pi^2)$$

$$T_{c_s,c_d}(SCP_{c_s,c_d}|SCP_{c_s,c_d}) \leq T_{c_s,c_d}(\pi^2|\pi^1)$$

Thus:

$$T_{c_s,c_d}(SCP_{c_s,c_d}|SCP_{c_s,c_d}) \leq \max(T_{c_s,c_d}(\pi^1|\pi^2), T_{c_s,c_d}(\pi^2|\pi^1))$$

Combining these results:

$$T_{g_1}(\pi'^1|\pi'^2) + T_{g_2}(\pi'^2|\pi'^1) \leq T_{g_1}(\pi^1|\pi^2) + T_{g_2}(\pi^2|\pi^1)$$

□

LEMMA 9 (EARLY COOPERATION). *Assume two cooperation nodes $c_i, c_j \in V_C$ such that a_1 and a_2 can both reach c_i by cooperating at c_j no later than they would by traveling directly, i.e.,*

$$t^*(c_i) \geq t^*(c_j) + T_{c_j,c_i}(SCP_{c_j,c_i}|SCP_{c_j,c_i}) + \tau_{c_i}^2.$$

Then, the minimal joint path time possible by a path starting cooperation at c_j (achieved by departing from $v_{d_j} = v_d(c_j)$) is at least as low as the minimal joint path time possible by a path starting cooperation at c_i (achieved by departing from $v_{d_i} = v_d(c_i)$), i.e.,

$$2 \cdot \left(t^*(c_j) + T_{c_j,v_{d_j}}(SCP_{c_j,v_{d_j}}|SCP_{c_j,v_{d_j}}) + \tau_{v_{d_j}}^2\right) + d^*(v_{d_j}) \leq$$

$$2 \cdot \left(t^*(c_i) + T_{c_i,v_{d_i}}(SCP_{c_i,v_{d_i}}|SCP_{c_i,v_{d_i}}) + \tau_{v_{d_i}}^2\right) + d^*(v_{d_i}).$$

THEOREM 5. *Given an* IC2PP *instance with two agents a_1 and a_2, Algorithm 6 finds the joint strategy (S^{1^*}, S^{2^*}) that optimizes the social welfare of a_1 and a_2.*

Outline. This result follows directly from Lemmas 8 and 9. Lemma 8 establishes that the joint strategy that optimizes social welfare is of the following form:

$$(\pi^{1^*}, \pi^{2^*}) = (SIP_{s_1,c_s} \circ SCP_{c_s,c_d} \circ SIP_{c_d,g_1}, SIP_{s_2,c_s} \circ SCP_{c_s,c_d} \circ SIP_{c_d,g_2}),$$

which implies that finding the joint strategy that optimizes social welfare is equivalent to finding the nodes c_s and c_d that minimize the total path time.

Lemma 9 demonstrates that if a cooperation node can be reached earlier through cooperation rather than individually, then this node is not the starting cooperation node in the optimal social welfare joint strategy and can be disregarded.

By combining these results, we conclude that Algorithm 6, which evaluates all possible cooperation nodes $c \in V_C$ as potential starting cooperation nodes (excluding those that can be pruned by Lemma 9), identifies their corresponding optimal departure nodes $v_d(c)$, and returns the paths π^{1^*}, π^{2^*}

$$(\pi^{1^*}, \pi^{2^*}) = (SIP_{s_1,c_s} \circ SCP_{c_s,c_d} \circ SIP_{c_d,g_1}, SIP_{s_2,c_s} \circ SCP_{c_s,c_d} \circ SIP_{c_d,g_2}),$$

corresponding to:

$$c_s = \arg\min_{c \in V_C} \left(2 \cdot \left(t^*(c) + T_{c,c_d}(SCP_{c,c_d}|SCP_{c,c_d}) + \tau_{c_d}^2\right) + d^*(v_d(c)\right)$$

$$c_d = v_d(c_s)$$

successfully finds the joint strategy (π^{1^*}, π^{2^*}) that optimizes the social welfare of a_1 and a_2.

Complexity Analysis. Every run of the algorithm starts with four executions of Dijkstra's algorithm to determine the shortest paths from s_1, s_2, g_1, and g_2. Each run of Dijkstra's algorithm has a time complexity of $O(|E| + |V| \log |V|)$. Consequently, the total complexity for these four runs is:

$$O(4 \cdot (|E| + |V| \log |V|)) = O(|E| + |V| \log |V|)$$

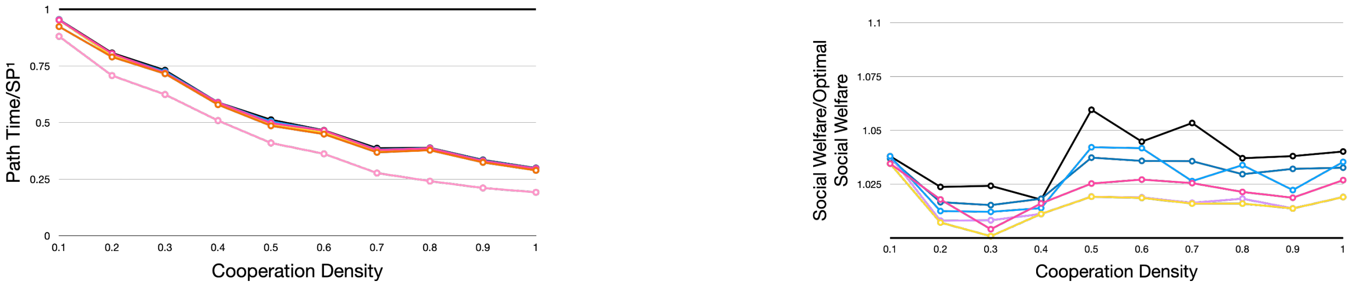
The algorithm then evaluates at most m potential starting cooperation nodes. Thus, the overall complexity for evaluating these nodes is:

$$O(m \cdot (|E| + |V| \log |V|))$$

Combining these, the total complexity of the algorithm is:

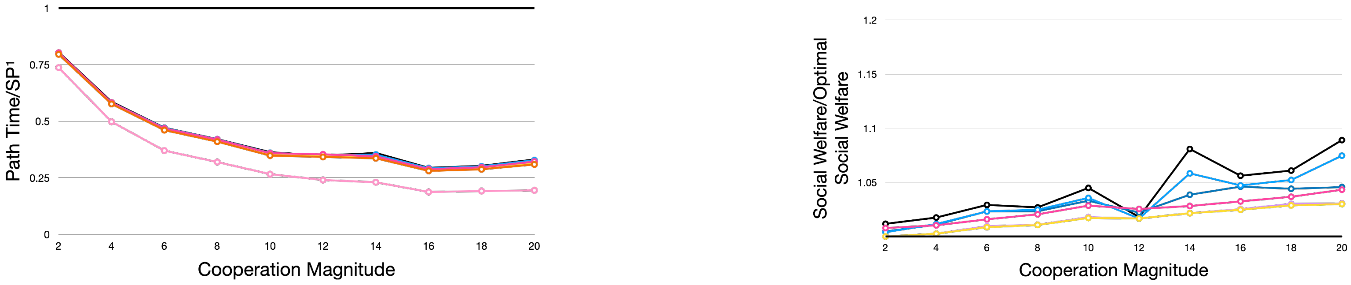
$$O(m \cdot (|E| + |V| \log |V|))$$

I EXPERIMENTAL RESULTS



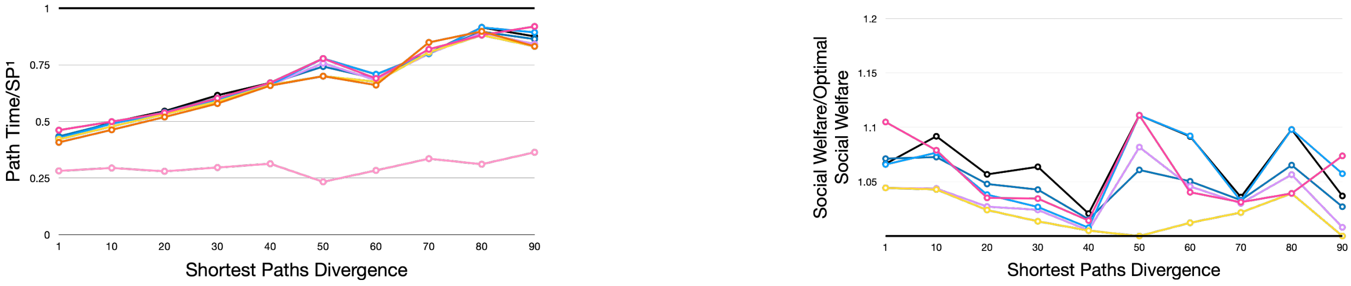
(a) Individual Path Time vs. Cooperation Density

(b) Social Welfare vs. Cooperation Density



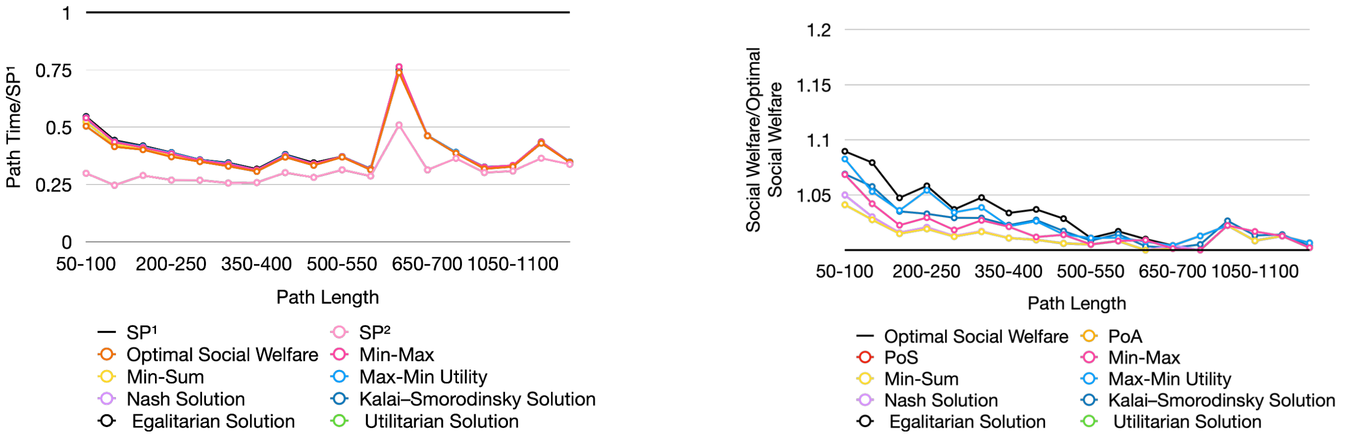
(c) Individual Path Time vs. Cooperation Magnitude

(d) Social Welfare vs. Cooperation Magnitude



(e) Individual Path Time vs. Shortest Paths Divergence

(f) Social Welfare vs. Shortest Paths Divergence



(g) Individual Path Time vs. Paths Lengths

(h) Social Welfare vs. Path Lengths

Figure 12: Effects of cooperation magnitude, density, path length, and path divergence on individual path times and social welfare.