A New Model for Multi-Robot Path Planning on Graphs with Using Network Flow

Samira Kashefi Biroon, Bahram Sadeghi Bigham and Salman Khodayifar

November 25, 2019

A New Model for Multi-Robot Path Planning on Graphs with Using Network Flow

Samira Kashefi Biroon* Bahram Sadeghi Bigham† Salman Khodayifar‡

Abstract

In this study, we address the problem of optimal multi-robot path planning (MPP) on graphs, focusing on the makespan criteria (minimizing the maximum arrival time). We use network flows concept and present an Integer Linear Programming (ILP) approach to solve the problem. In contrast to previous studies that used gadgets to avoid the head-on collision, our model applies some constrains on the edges and vertices instead of using gadgets. These constraints significantly reduce the number of edges and vertices in our designed network in compared with the presented networks flow in related studies. More precisely, the number of edges and vertices in our network are $(nT)+(2m-1)T$ and $n(T+1)$, respectively, where T, m, and n are time-horizon, the number of main graph edges and vertices, respectively. Whereas these numbers in the case of using the gadgets are $5mT + 2nT$ and $n(2T + 1) + 2T(n - 1)$.

1 Introduction

Multi-robot Path Planning (MPP) is one of the most important problems in robotics. The MPP problem should find a safe and feasible path set for robots in order to reach initial goals without the occurrence of collision. We study a specific type of graph where every edge has a unit weight as distance, and the robots can move simultaneously with unit speed through the edges. A distinguishing feature is that robots can move on fully occupied cycles to rotate synchronously. We say that there is collision between two paths when two robots want to change their configurations with each other (head-on collision or collision in edge), or when two robots want to enter a common vertex at the same time (collision in vertex). The MPP problem includes important applications in real world such as autonomous warehouse systems [1], evacuation problem [2, 3], object transportation [4, 5], and formation control [6, 7, 8]. The MPP problem consist of four distinct minimization objectives: the makespan (minimizing the maximum arrival time), the maximum distance (single-robot traveled), the total arrival time, and the total distance. We use the network flows concept and present an ILP approach to solve the makespan objective of the MPP problem. (Section 3 discusses the network flow in more detail).

Yu et al. [9] with inspired of “escape problem” in [10] established the link between multi robot path planning on graph and network flow. They presented a complete algorithm for the planning problem in which robots have no pre-specificed goals. The algorithm finds a solution to the problem in $O(k|V||E|)$ time, where k is the number of robots, $|V|$ is the number of vertices, and $|E|$ is the number of edges. In addition, they showed that a longest finish time is not more than $k + |V| - 1$ steps. In the study of [11], they indicated that objectives are distinct for pre-specific goals, and optimizing the problem is NP-hard. They firstly established a one-to-one solution mapping between the MPP and network flow. Then, based on this equivalence and ILP, they designed a complete algorithm for optimizing over each of the four objectives. In order to change the main graph to time-expanded network, they used a gadget for splitting an undirected edge through time steps for dealing with head-on collision constraint. However, the authors provided a solution to compute minimum makespan with robot-vertex ratio up to 100%. This is because using gadgets in the time-expanded network causes to dramatically enlarge the network in terms of the number of edges and vertex. In the another paper [12] Yu and et al. demonstrated that each pair of the four objectives induces a Pareto and cannot always be optimized simultaneously. Then, through reductions from 3SAT, they further established that computation over each objective is an NP-hard task, providing evidence that solving MPP optimally is generally intractable.

In this paper, we present a new ILP model to minimize the makespan objective. As apposed to the provided model in [11], we use no extra gadgets because it enlarges the time-expanded network in terms of the number of the edges and vertices. We apply some constrains that do not allow to robots for entering common edges or changing their location at the same time. This enable the model to handle head-on collision in

*Department of Computer Science and Information Technology, Institute of advance studies in basic science, Zanjan, Iran, s.kashefi67@gmail.com

†Department of Computer Science and Information Technology, Institute of advance studies in basic science, Zanjan, Iran, b_sadeghi_b@iasbs.ac.ir

‡Department of Mathematics, Institute of advance studies in basic science, Zanjan, Iran, s.khodayifar@iasbs.ac.ir

the time-expanded network without the need for gadgets. In our model, the number of edges and vertices are $(nT)+(2m-1)T$ and $n(T+1)$, respectively. Where T is time-horizon, m is number of main graph edges, and n is number of main graph vertices. Whereas the number of edged and vertices by using the gadgets are $n(2T+1)+2T(n-1)$, and $5mT+2nT$. In fact, we minimize the makespan objective with normal size of time-expanded network.

The rest of the paper is organized as follows. In Sect. 2, we define multi-robot path planning on a graph. Sect. 3 reviews dynamic network flow. Sect. 4 derives a new ILP model, then compares our time-expanded network and the presented network in [11]. Finally, we conclude our study in Sect. 5, and mention open problems that will tackle in upcoming works.

2 Multi Robot Path Planning on a Graph

Let $G=(V,E)$ be a connected, undirected, simple graph where V and E are the vertex and edge sets, respectively. Let $R=\{r_1,...,r_k\}$ denotes a set including k robots on the graph such that during a discrete time step, each robot may either remain stationary or move to an adjacent vertex. The initial and final configuration of the robots are denoted as X_I and X_G, respectively. To formally describe a plan, an scheduled path defines as a map $p_i:\mathbb{Z}^+\rightarrow V$ in which $\mathbb{Z}^+ := \{N\cup 0\}$. The scheduled path p_i is feasible if it satisfies the following requirements:

- $p_i(0)=x_I(r_i)$
- For each i, there exists a smallest $t_i\in\mathbb{Z}^+$ such that $p_i(t_i)=x_G(r_i)$
- For any $t>t_i$, $x_g(r_i)\equiv p_i(t)$
- For any $0\le t<t_i^{min}$, $(p_i(t),p_i(t+1))\in E$ or $p_i(t)=p_i(t+1)$. We say that two paths p_i,p_j are in collision if:
  - $\forall t\in Z^+, p_i(t)=p_j(t)$ (meet collision)
  - $((p_j(t+1),p_j(t))=((p_i(t),p_i(t+1))$ (head-on collision).

2.1 Multi-robot Path Planning on Graph problem

Given a 4-tuple (G,R,x_G,x_I), find a set of feasible paths $P=\{p_1,...,p_n\}$ so that each path p_i connect two $(X_I(i),X_G(i))$ pre-specified, which there is not collision between two paths p_i,p_j. Goal (optimal paths) is to minimize the following four objectives:

Objective 1 (Total arrival time): Compute a path set P that minimizes $\sum_{i=1}^{n} t_i$.

Objective 2 (Total distance): Compute a path set P that minimizes $\sum_{i=1}^{n} len(p_i)$.

Objective 3 (Makespan): Compute a path set P that minimize $Max_{1\leqslant i\leqslant n} t_i$.

Objective 4 (Maximum Distance): Compute a path set P that minimize $Max_{1\leqslant i\leqslant n} len(p_i)$.

At the present study we focus on the objective 3 (makespan). Fig.1 shows the undirected graph G, with start vertices $\{s_i\}$, $i=1,2$ and goal vertices $\{g_i\}$, $i=1,2$. An instance of MPP is given by $(G,\{r_1,r_2\},x_I : r_i\longmapsto s_i, x_G: r_i\longmapsto g_i)$.

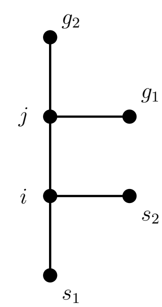

Figure 1: A simple graph G, $\{s_i\}$ and $\{g_i\}$ are start and goal vertices, respectively, $(i=1,2)$, also vertices i and j are intermediate vertices.

3 Multicommodity Flows

Let $G=(V,E)$ be a directed network with vertex set V ($|V|=n$), edge set E ($|E|=m$), and set of commodities $K=\{1,2,\ldots,h\}$ that must be routed through the same network. Every commodity $k\in K$ has only one source $s_k^+\in N$ and one sink $s_k^-\in N$. R_k is the amount of supply or demand of commodity $k\in K$.

Each edge $(i,j)\in A$ has a capacity u_{ij} that restricts the total flow of all commodities on that edge. For commodity k, let $\mathbf{x}^k=(x_{ij}^k)_{(i,j)\in A}$, $\mathbf{d}^k=(d_i^k)_{i\in N}$, and $\mathbf{c}^k=(c_{ij}^k)_{(i,j)\in A}$ present the flow vector, supply-demand vector, and per unit cost vector. Using defined notations, the multicommodity flow (for short, MCF) problem can be characterized as follows:

$$\text{MCF : min} \sum_{k=1}^{h}\sum_{(i,j)\in A} c_{ij}^k x_{ij}^k \tag{1}$$

s.t.

$$\sum_{k=1}^{h} x_{ij}^k \le u_{ij}, \forall (i,j)\in A, \tag{2}$$

$$\sum_{\{j:(i,j)\in A\}} x_{ij}^k - \sum_{\{j:(j,i)\in A\}} x_{ji}^k = \begin{cases} R_k & if\ i=s_k^+ \\ -R_k & if\ i=s_k^- \\ o & o.w., \end{cases}$$

$$\forall i\in N,\ k\in K, \tag{3}$$

$$x_{ij}^k \ge 0, \forall (i,j)\in A,\ k\in K. \tag{4}$$

The objective function (1) minimizes the total cost of the multicommodity flow. Constraints (2) implement the bundle constraint on each vertex $(i,j) \in A$. Constraints (3) are separate flow conservation constraints for each commodity $k \in K$. Constraints (4) are the continuous restrictions on the decision variables.

3.1 Dynamic Multicommodity Flow Problem

We consider a directed network $G = (V, E, K, c, u, \tau, \mathcal{T})$ with set of vertices $|V| = n$, set of edges $|E| = m$, and set of commodities $K = \{1, 2, \ldots, h\}$ that must be routed through the same network. We consider the discrete time model, in which all times are integral and bounded by time horizon T, which defines the set $\mathcal{T} = \{0, 1, \ldots, T\}$ of time moments. We define $c_{ij}^k(t)$ and $c_i^k(t)$ as the cost for sending one unit of flow kth on edge (i,j) in time t and the cost for storing one unit of flow kth at vertex i from time $t-1$ to t, respectively. Moreover, $u_{ij}(t)$, $u_i^k(t)$, and $\tau_{ij}(t)$ are an upper bound on the amount of flow that can enter to edge (i,j) at time $t \in \mathcal{T}$, an upper bound on the amount of flow that can be stored in vertex i from time $t-1$ to t, and positive transmit time on edge (i,j) at time $t \in \mathcal{T}$, respectively.

Time is measured in discrete steps, so that if one unit of flow of commodity k leaves vertex i at time t, one unit of flow arrives at vertex j at time $t + \tau_{ij}(t)$. For commodity k, let $\mathbf{x}^k = (x_{ij}^k(t))_{(i,j)\in A}$ denote the dynamic flow vector and flow variables $x_{ij}^k(t)$ present the flow of the commodity k on edge (i,j) at time t. We assume that the flow variables $x_{ij}^k(t)$ have no individual flow bounds; that is, each $u_{ij}^k(t) = +\infty$. For commodity k, let $\mathbf{y}^k = (y_i^k(t))_{i\in N}$ denote the storage vector and $y_i^k(t)$ gives the amount of commodity flow k stored at vertex i from time $t-1$ to t.

The dynamic multi-commodity flow (for short, DMF) problem with storage at intermediate vertices can be modeled as follows:

$$\text{DMF : min } \sum_{t=0}^{T} \Big(\sum_{k\in K} \sum_{(i,j)\in A} c_{ij}^k(t) x_{ij}^k(t) + \sum_{k\in K} \sum_{i\in N} c_i^k(t) y_i^k(t) \Big) \tag{5}$$

s.t.

$$\sum_{k\in K} x_{ij}^k(t) \le u_{ij}(t), \quad \forall (i,j) \in A,\, t \in \mathcal{T} \tag{6}$$

$$\sum_{\{j:(i,j)\in A\}} x_{ij}^k(t) - \sum_{\{j:(j,i)\in A\}} \sum_{\{t':t'+\tau_{ji}(t')=t\}} x_{ji}^k(t') +$$

$$y_i^k(t) - y_i^k(t-1) = 0, \quad \forall k \in K,\, i \in N - \{s_k^+, s_k^-\},\, t \in \mathcal{T} \tag{7}$$

$$\sum_{t=0}^{T} \Big(\sum_{\{j:(i,j)\in A\}} x_{ij}^k(t) - \sum_{\{j:(j,i)\in A\}} \sum_{\{t':t'+\tau_{ji}(t')=t\}} x_{ji}^k(t') \Big) =$$

$$\begin{cases} R_k & if\ i = s_k^+ \\ -R_k & if\ i = s_k^- \\ o & o.w. \end{cases}, \quad \forall i \in N,\, k \in K, \tag{8}$$

$$x_{ij}^k(t) \ge 0,\ 0 \le y_i^k(t) \le u_i^k(t), \quad \forall (i,j) \in A,\, i \in N,\, k \in K,\, t \in \mathcal{T}. \tag{9}$$

The objective function (5) minimizes the total cost of the dynamic flow vector $\mathbf{x}^k$ and storage flow vector $\mathbf{y}^k$ in time horizon T. Constraints (6) implement the bundle constraint on each edge $(i,j) \in A$ at any time step t. For commodity k, constraints (7) indicate the flow conservation constraints and amount of stored flow at intermediate vertex i in each time step t. For commodity k, constraints (8) indicate the flow conservation constraints in time horizon T. That is, after all the time steps have passed, the dynamic flow vectors $\mathbf{x}^k$, $\forall k \in K$ should be satisfy the supply/demand of every commodity and there are no additional flows in the intermediate vertices. Constraints (9) indicate capacity constraint and storage capacity constraint for dynamic flow vectors $\mathbf{x}^k$, $\forall k \in K$ and storage vectors $\mathbf{y}^k$, $\forall k \in K$, respectively.

4 An ILP Model for Makespan

As mentioned earlier, we present an ILP model to solve the makespan objective in order to prevent collision of edges and vertices. In the given model, $c_{ij}^k(t)$ is considered 1 for all the robots and $x_{ij}^k(t)$ is 1 if the robot k in the time t is on the edge (i,j) and otherwise it is equal zero. Transmit time on an edge is considered 1 for all the robots. The suggested ILP model is as follows:

$$\min \sum_{t=0}^{T} \sum_{k=1}^{|L|} \sum_{(i,j)\in A} c_{i,j}^k x_{ij}^k(t) + \sum_{t=0}^{T} \sum_{k=1}^{|L|} \sum_{i\in A} y_i^k(t) \tag{10}$$

s.t.

$$\sum_{\{j|(i,j)\in A\}} x_{ij}^k(t) - \sum_{\{j|(j,i)\in A\}} x_{ji}^k(t-1) + y_i^k(t) = 0,$$

$$\forall i \in N,\ k \in K, t \in \{0, 1, ..., T\} \tag{11}$$

$$\sum_{t=0}^{T} \sum_{\{j|(i,j)\in A\}} x_{ij}^k(t) - \sum_{t=0}^{T} \sum_{\{j|(j,i)\in A\}}$$

$$x_{ji}^k(t-1) = \begin{cases} 1 & i = O(k) \\ -1 & i = D(k) \\ 0 & \text{otherwise} \end{cases}, \forall i \in N,\ \forall k \in L \tag{12}$$

$$\sum_{k\in K} x_{ij}^k(t) \le 1, \qquad \forall t \in \{0, 1.., T\}, \quad \forall (i,j) \in A \tag{13}$$

$$\sum_{k\in K} \sum_{i\in N} x_{ij}^k(t) \le 1, \qquad \forall t \in \{0, 1, ..., T\},\ \forall (i,j) \in A \tag{14}$$

$$\sum_{k\in K} \sum_{i\in N} y_i^k(t) \le 1, \qquad \forall t \in \{0, 1, ..., T\}, \tag{15}$$

$$x_{ij}^k(t) \in \{0, 1\}, \qquad \forall (i,j) \in A,\ \forall k \in K\ \forall t \in \{0, 1, ..., T\} \tag{16}$$

$$0 \leqslant y_i^k(t) \leqslant 1, \qquad \forall i \in N,\ \forall k \in K,\ \forall t \in \{0, 1, ..., T\} \tag{17}$$

Constrains (11) show that the robots are able to move in place and intermediate vertices, which is equal to

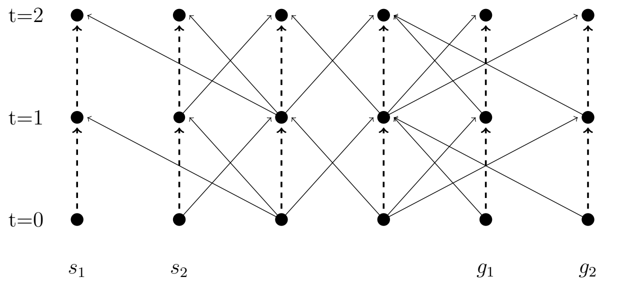

Figure 2: The time-expanded network based on proposed model.

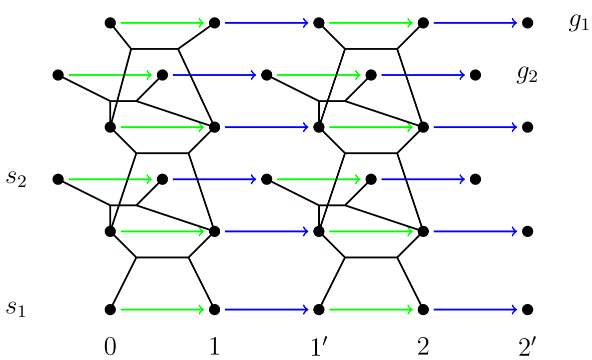

Figure 3: The time-expanded network based on [11].

the constrain in the storage of networks. (12) are balance constrains for initial, goal and intermediate vertices. Constraint (13) expresses that the head-on collision does not occur. Constraint (14) prevents to meet collision. Constraints (15) show that at each vertex and each given moment, one robot can be moved in place. Constraints (16), (17) are the continuous restrictions on the decision variables.

In order to compare our model and the provided model in [11], Fig.(2) and Fig.(3) illustrate the time-expanded networks with an expansion time horizon of $T = 2$. In other words, Fig.(2) shows that without the need for gadget we are able to handle head-on collision, where the number of vertices and edges is equal to 18 and 30, respectively. Fig.(3), on the other hand, depicts the time-expanded network with 50 vertices and 74 edges. As a result, as we mentioned, it is clear that the number of vertices and edges in our model is more less than the study of [11]. Note that, in [11] the number of vertices and edges would be significantly increased for bigger graphs.

5 Conclusion and Future works

In this paper, we have investigated the minimizing the maximum arrival time of optimal MPP problems based on network flow for presenting a new ILP model. We show that our new ILP model in contrast to the previous model notably reduces the number of edges and vertices in designed network. There is still a open problem that we would like to tackle in our future works. We shall focus on a ILP model to minimize the maximum distance.

References

[1] Wurman, Peter R., Raffaello D'Andrea, and Mick Mountz Coordinating hundreds of cooperative, autonomous vehicles in warehouses. *AI magazine*, 2008.

[2] Chalmet, L. G., R. L. Francis, and P. B. Saunders Network models for building evacuation. *Management science*, 1982.

[3] Rodriguez, Samuel, and Nancy M. Amato Behavior-based evacuation planning. *Robotics and Automation (ICRA),IEEE International Conference on. IEEE*, 2010.

[4] Mataric, Maja J., Martin Nilsson, and Kristian T. Simsarin Cooperative multi-robot box-pushing. *iros. IEEE*, 1995.

[5] Rus, Daniela, Bruce Donald, and Jim Jennings. Moving furniture with teams of autonomous robots. *Intelligent Robots and Systems 95.'Human Robot Interaction and Cooperative Robots', Proceedings. 1995 IEEE/RSJ International Conference on. IEEE*, 1995.

[6] Shucker, Brian, Todd Murphey, and John K. Bennett Switching rules for decentralized control with simple control laws. *American Control Conference, IEEE*, 2007.

[7] Smith, Brian S., Magnus Egerstedt, and Ayanna Howard Automatic deployment and formation control of decentralized multi-agent networks. *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on. IEEE*, 2008.

[8] Tanner, Herbert G., George J. Pappas, and Vijay Kumar Leader-to-formation stability. *IEEE Transactions on robotics and automation*, 2004.

[9] Yu, Jingjin, and Steven M. LaValle Multi-agent path planning and network flow *Algorithmic foundations of robotics*, 2013.

[10] Cormen, Thomas H., et al. Introduction to algorithms. *MIT press*, 2009.

[11] Yu, Jingjin, and Steven M. LaValle Optimal multirobot path planning on graphs: Complete algorithms and effective heuristics. *IEEE Transactions on Robotics*, 2016.

[12] Yu, Jingjin, and Steven M. LaValle Optimal multi-robot path planning on graphs: Structure and computational complexity. *arXiv preprint arXiv:1507.03289*, 2015).