

The Quickest Delivery on the Super Pipeline Based Logistic System

Zhongyuan Lai

Institute for Interdisciplinary Research
Jiangnan University
Wuhan, Hubei Provinces, China

Hao Li*

Institute for Interdisciplinary Research,
Jiangnan University,
Wuhan, Hubei Provinces, China
L.R.I, UMR 8623,
CNRS and Université Paris-Sud 11
F-91405 Orsay, France
*Corresponding author

Tonghui Qian and Bo Wang

Institute for Interdisciplinary Research
Jiangnan University
Wuhan, Hubei Provinces, China

Abstract—In this paper, we consider the problem of delivering replenishments from source to sink nodes on a delivery line in minimum time interval, where the halfway stopping and intercrossing are forbidden. We call this problem the quickest delivery problem, and define it in a mathematical way. We point out that this problem is equivalent to the problem of moving time-position segments along the time axis such that any two segments are not intersected with each other and the total time is minimized. Based on this point, we formulate the quickest delivery problem as a minimization problem, with the non-intercrossing constraints as non-intersection constraints among time-position segments in four cases. We can solve this minimization problem by using optimization toolboxes. Experiments confirm the effectiveness of our model.

Keywords-logistic system; quickest delivery; delivery line; time-position segment

I. INTRODUCTION

With the rapid development of electronic commerce, delivery problem has increasingly become the bottleneck of the development of electronic commerce. How to improve the efficiency and velocity of deliveries becomes an urgent problem that should be solved in electronic commerce [1].

It is a good starting point to solve the above delivery problem by using automation and hypervelocity. Recently, both industry and academia have proposed a logistics solution using super pipeline [2] [3]. Because eighty percent of our daily necessities are of small sizes, they can be delivered through pipelines. The radii of these pipelines range from 20 centimeters to 1.5 meters, so as to satisfy different logistic requirements. The necessities of small sizes can be sorted and encapsulated in the capsule containers using automatic sorting technology, and then delivered to different destinations through pipeline-based logistic system. In this way, the automation and

hypervelocity are implemented on the delivery of small daily necessities.

The total efficiency and average velocity of a super pipeline-based logistic system depend not only on the automation sorting technology and hyper velocity delivery technology, but also on the container delivery strategy. Because there are many logistic nodes along a pipeline, there may be many deliveries among different nodes in the same time interval. How to arrange these deliveries, so that the total delivery time is minimized provided that each delivery can be directly transported to their destinations once they enter the super pipeline, is the key to ensure the total efficiency and velocity of the super pipeline based logistic system.

The above delivery problem is essentially a quickest multi commodity flow problem, which is demonstrated to be NP-hard [4]. With the increasing number of deliveries, it is infeasible to find the optimal solution from a computational perspective. However, compared with general quickest multi commodity flow problem, the particularity of our quickest delivery problem is that there only exists a single path for a given delivery. This particularity allows us to use the time-position segment to describe the path of each delivery, and to transform the quickest delivery problem to the optimal time sequencing problem of the time-position segments. Meanwhile, we can use the knowledge of the plane geometry to describe the non-intercrossing constraints in the delivery processes. In this way, we can establish the corresponding optimization model, and find an optimal solution through the existing optimization toolboxes [5]. The simulation results confirm the effectiveness of our model.

II. THE QUICKEST DELIVERY MODEL

We study the discrete quickest delivery problem on a line. Our aim is to find the starting time of each delivery, so as to

minimizing the total delivery time provided that the halfway stopping and intercrossing are forbidden. Now we describe this problem in a mathematical way.

Assume that there are K deliveries, numbered with $0, 1, \dots, \text{and } K-1$, needed to be delivered on a line with N nodes, numbered with $0, 1, \dots, \text{and } N-1$. For the i th delivery, let t_i and l_i be its starting time and delivery duration, so its ending time is $t_i + l_i$. Let s_i and v_i be its starting position and velocity, where v_i can only take the value of 1 or -1, so its ending position is $e_i = s_i + v_i l_i$. Let $p_i(t)$ be the position at the time t . It can be expressed as

$$p_i(t) = \begin{cases} s_i, & t < t_i \\ s_i + v_i(t - t_i), & t_i \leq t \leq t_i + l_i \\ s_i + v_i l_i, & t \geq t_i + l_i \end{cases} \quad (1)$$

Here we can use the time-position coordinates to describe the process of the i th delivery. As shown in Figure I, this

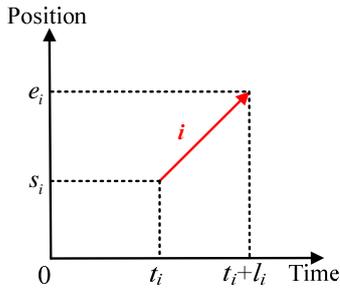


FIGURE I. ILLUSTRATION OF THE TIME-POSITION SEGMENT OF THE i TH DELIVERY

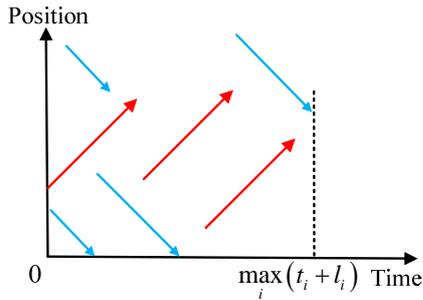


FIGURE II. ILLUSTRATION OF THE TIME-POSITION SEGMENTS OF ALL THE DELIVERIES.

delivery can be regarded as a line segment i in the time-position plane, with its slope v_i . Its starting position s_i and ending position e_i are fixed, but its starting time t_i is adjustable. The goal of the quickest delivery is to optimally adjust the starting time $\{t_i | t_i \in \mathbb{Q}, k=0, \dots, K-1\}$, so as to minimize the total delivery time as shown in Figure II. Therefore, the objective function is

$$\min_{\{t_i\}} \max_i (t_i + l_i). \quad (2)$$

The constraint of the above min-max problem is that the distance of any two deliveries should be no less than one at any time, i.e., the vertical distance between any two time-position segments should be no less than one at any time. Now we discuss this constraint in detail.

Assume that there are two deliveries numbered i and j , which should be transported with the velocity v_i and v_j . If the paths of these two deliveries do not overlap each other, then the above constraint is satisfied. Otherwise, there are two cases needed to be discussed.

Case 1: i and j have the same velocity, i.e., $v_i = v_j$. In this case, their time-position segments i and j are parallel to each other, as shown in Figure III. Thus, the vertical distance between them being no less than one at any time is equivalent to the distance between their vertical intercepts (double headed arrow in green in Figure III) being no less than one. The vertical intercepts of i and j are $s_i - v_i t_i$ and $s_j - v_j t_j$

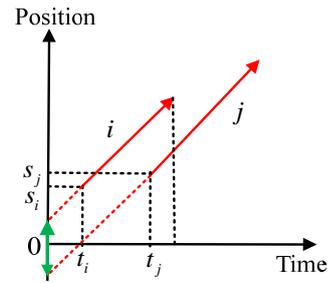


FIGURE III. ILLUSTRATION OF NON-INTERCROSSING CONSTRAINTS WHEN TWO DELIVERIES i AND j HAVE THE SAME DIRECTION.

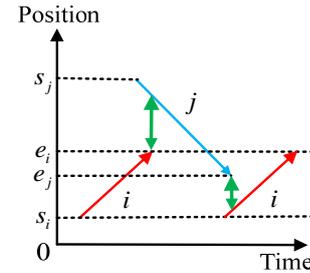


FIGURE IV. ILLUSTRATION OF NON-INTERCROSSING CONSTRAINTS WHEN TWO DELIVERIES i AND j HAVE OPPOSITE DIRECTIONS WITH $s_i < e_j < e_i < s_j$.

respectively. Therefore, the above equivalent constraint can be written as

$$v_i t_i - v_j t_j \leq s_i - s_j - 1 \quad \text{or} \quad v_i t_i - v_j t_j \geq s_i - s_j + 1. \quad (3)$$

Case 2: i and j have different velocities, i.e., $v_i \neq v_j$. In this case, their time-position segments i and j are perpendicular to each other. Without loss of generality, assume that $v_i = 1$ and $v_j = -1$. According to the relationships among their starting positions s_i, s_j and ending positions e_i, e_j , there are four subcases needed to be discussed.

Subcase 1: $s_i < e_j < e_i < s_j$. In this subcase, the vertical distance between i and j being no less than one at any time is equivalent to both the vertical distances between e_i and j and between e_j and i (double headed arrows in green in Figure IV) being no less than 1. This equivalent constraint can be written as

$$p_j(t_i + l_i) - p_i(t_i + l_i) \geq 1 \text{ or } p_j(t_j + l_j) - p_i(t_j + l_j) \geq 1, \quad (4)$$

which can be simplified to

$$t_i - t_j \leq s_j - s_i - 2l_i - 1 \text{ or } t_j - t_i \leq s_j - s_i - 2l_j - 1. \quad (5)$$

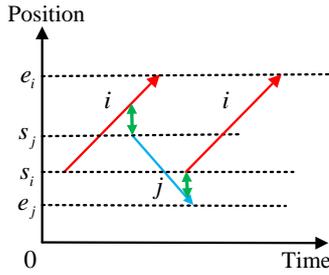


FIGURE V. ILLUSTRATION OF NON-INTERCROSSING CONSTRAINTS WHEN TWO DELIVERIES i AND j HAVE OPPOSITE DIRECTIONS WITH $e_j < s_i < s_j < e_i$.

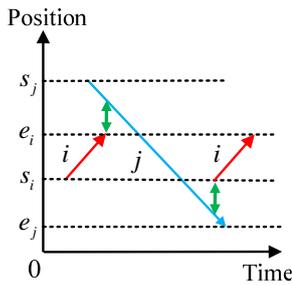


FIGURE VI. ILLUSTRATION OF NON-INTERCROSSING CONSTRAINTS WHEN TWO DELIVERIES i AND j HAVE OPPOSITE DIRECTIONS WITH $e_j < s_i < e_i < s_j$.

Subcase 2: $e_i < s_j < s_i < e_j$. In this subcase, the vertical distance between i and j being no less than one at any time is equivalent to both the vertical distances between s_i and

j and between s_j and i (double headed arrows in green in Figure V) being no less than 1. This equivalent constraint can be written as

$$p_i(t_j) - p_j(t_j) \geq 1 \text{ or } p_i(t_i) - p_j(t_i) \geq 1, \quad (6)$$

which can be simplified to

$$t_j - t_i \geq 1 - s_i + s_j \text{ or } t_i - t_j \geq 1 - s_i + s_j. \quad (7)$$

Subcase 3: $e_j < s_i < e_i < s_j$. In this subcase, the vertical distance between i and j being no less than one at any time is equivalent to both the vertical distances between s_i and j and between e_i and j (double headed arrows in green in Figure VI) being no less than 1. This equivalent constraint can be written as

$$p_j(t_i + l_i) - p_i(t_i + l_i) \geq 1 \text{ or } p_i(t_i) - p_j(t_i) \geq 1, \quad (8)$$

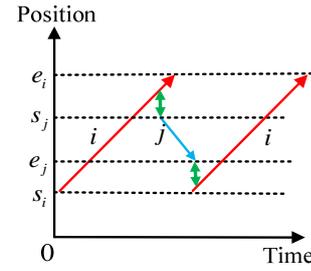


FIGURE VII. ILLUSTRATION OF NON-INTERCROSSING CONSTRAINTS WHEN TWO DELIVERIES i AND j HAVE OPPOSITE DIRECTIONS WITH $s_i < e_j < s_j < e_i$.

which can be simplified to

$$t_i - t_j \leq s_j - s_i - 2l_i - 1 \text{ or } t_i - t_j \geq 1 - s_i + s_j. \quad (9)$$

Subcase 4: $s_i < e_j < s_j < e_i$. In this subcase, the vertical distance between i and j being no less than one at any time is equivalent to both the vertical distances between s_j and i and between e_j and i (double headed arrows in green in Figure VII) being no less than 1. This equivalent constraint can be written as

$$p_i(t_j) - p_j(t_j) \geq 1 \text{ or } p_j(t_j + l_j) - p_i(t_j + l_j) \geq 1, \quad (10)$$

which can be simplified to

$$t_j - t_i \geq 1 - s_i + s_j \text{ or } t_j - t_i \leq s_j - 2l_j - s_i - 1. \quad (11)$$

Now we have formulated the quickest delivery problem as a min-max problem with the objective function given by Equation (2) and the corresponding constraints given by Equations (3), (5), (7), (9), and (11). We can efficiently solve it by using the library of YALMIP [5].

III. THE EXPERIMENTAL RESULTS

In this section, we conducted an experiment to verify the effectiveness of our model. We randomly generated K pairs of

natural numbers ranging from 0 to $N-1$ as the starting and ending position pairs of K deliveries needed to be transported along a line with N nodes. Then, we built the quickest delivery model as a min-max problem and found the optimal starting time solutions via MATLAB +YALMIP+CPLEX.

In order to display a more intuitive result, we set $N = K = 100$. The generated starting and ending position pairs of 100 deliveries are listed in Table I. The optimal starting times of these deliveries are listed in Table II.

TABLE I. THE GENERATED STARTING AND ENDING POSITION PAIRS OF 100 DELIVERIES

(60, 87)	(47, 26)	(70, 32)	(70, 12)	(64, 94)	(3, 65)	(6, 48)	(32, 64)	(53, 55)	(66, 65)
(41, 54)	(82, 72)	(72, 52)	(97, 100)	(53, 22)	(32, 10)	(10, 11)	(61, 6)	(78, 40)	(42, 45)
(9, 36)	(26, 77)	(15, 63)	(28, 77)	(44, 94)	(53, 98)	(46, 19)	(88, 14)	(52, 70)	(95, 9)
(64, 53)	(96, 53)	(24, 86)	(68, 48)	(29, 39)	(70, 74)	(6, 52)	(25, 35)	(22, 15)	(67, 59)
(85, 26)	(34, 4)	(78, 76)	(68, 24)	(0, 44)	(60, 69)	(39, 36)	(92, 74)	(0, 39)	(46, 69)
(42, 71)	(46, 44)	(77, 1)	(32, 33)	(79, 42)	(47, 27)	(3, 19)	(17, 82)	(72, 43)	(47, 89)
(15, 39)	(34, 77)	(61, 40)	(19, 81)	(74, 76)	(24, 38)	(92, 21)	(27, 29)	(77, 95)	(19, 33)
(29, 67)	(9, 44)	(58, 84)	(69, 77)	(55, 16)	(42, 87)	(65, 99)	(65, 51)	(68, 89)	(64, 59)
(95, 15)	(21, 20)	(71, 41)	(23, 75)	(12, 83)	(61, 79)	(45, 32)	(46, 53)	(66, 9)	(77, 11)
(35, 13)	(66, 68)	(42, 50)	(85, 19)	(84, 49)	(25, 14)	(61, 5)	(58, 85)	(54, 56)	(93, 70)

TABLE II. THE OPTIMAL STARTING TIMES OF THESE DELIVERIES

20	65	62	60	116	0	0	28	0	3	8	0	61	31	62	55	0	59	75	3
0	0	0	3	10	16	67	72	14	60	2	66	0	60	1	0	1	2	20	0
71	77	0	61	21	17	0	0	22	14	11	1	66	0	75	67	1	1	63	11
1	4	60	1	1	3	69	0	24	2	0	1	17	30	61	7	19	0	23	0
62	0	63	1	0	17	65	0	65	67	93	2	2	74	74	23	61	16	0	0

According to the above data, the minimum total time is 146.

IV. CONCLUSION

In this paper, we investigated the quickest delivery problem for the super pipeline based logistic system, where each delivery should be transported to its destination without halfway stopping and intercrossing. We mapped each delivery into the time-position coordinates, and pointed out that this problem is equivalent to minimizing the maximum ending time among the deliveries, where the time-position segments of any two deliveries should not be intersected with each other. Based on this point, we modeled the quickest delivery problem as a min-max problem, and solved it using the existing optimization toolboxes. Experiment results confirm the effectiveness of our model.

ACKNOWLEDGMENT

The authors wish to thank Chun Wang and Xia Liu for their suggestion. This work is supported by the Hubei Provincial

Natural Science Foundation of China (Grant No. 2013CFB214), the Wuhan Municipal Key Technologies Research and Development Programs (Grant Nos. 2013011001010484 and 2013011001010463), the Industry-Academy-Research Program of Wuhan Municipal Universities and Colleges (Grant No. CXY04), and the Wuhan Innovative Talent Development Fund to Dr. Tonghui Qian.

REFERENCES

- [1] N. Malhotra, E-Commerce Boom Triggers Transformation In Retail Logistics. Res. Rep., Jones Lang LaSalle, Chicago, IL, 2013.
- [2] E. Musk, Hyperloop Alpha. SpaceX, Hawthorne, NE, 2013.
- [3] B. Dodson, Beyond the Hype of Hyperloop: An Analysis of Elon Musk's Proposed Transit System. Res. Rep., GizMag, 2013.
- [4] A. Hall, S. Hippler, and M. Skutella, "Multicommodity flows over time: Efficient algorithms and complexity," Theoretical Computer Science, vol. 379, no. 3, pp. 387-404, 2007.
- [5] J. Löfberg, "YALMIP: A toolbox for modeling and optimization in MATLAB," in Proc. CACSD Conf., Taipei, Taiwan, 2004.