The Tenth International Symposium on Operations Research and Its Applications (ISORA 2011)
Dunhuang, China, August 28–31, 2011

Studies on the Heuristic Algorithm for the N-Vehicle Exploration Problem (NVEP)[1]

Ruo-Yang Liu[2] Huai-Bao Lu[3] Jin-Chuan Cui[4] Xiao-Bing Li[5]

2,4 Institute of Applied Mathematics
Academy of Mathematics and Systems Science, CAS, Beijing 100190
3,5 Daqing Oilfield Limited Company, Daqing, Heilongjiang Province, 163453

Abstract This paper studies some properties of the exploration problem with N vehicles, which consist a necessary condition for the optimal solution. Based on this condition, the authors design a heuristic algorithm, into which the greedy idea is introduced. The computational complexity of the algorithm is $O(n^4)$. After that, by calculating 10 numerical examples selected from paper [4], we compare the effectiveness of this algorithm with two kinds of heuristic algorithm proposed in paper [4], and the simulation results show that even if it costs a little more computing time which could be controlled in polynomial time, the new algorithm we design in this paper has a much higher approximation ratio.

Keywords N-Vehicle Exploration Problem(NVEP); The Greedy Heuristic Algorithm; Paired Comparison Matrix; Approximation Ratio

1 Introduction

It is a usual problem that how to effectively utilize limited resource to support an exploration for an optimal purpose such as achieving the farthest or the highest. These kinds of problem can be abstracted into a class of N-Vehicle Exploration Problem (NVEP). NVEP can be generally described as follows ([1]): there are a fleet of n vehicles with oil capacities a_i liters and oil efficiencies b_i liters per unit distance $(i = 1,2,\cdots,n)$. The fleet starts from the same position at the same time, and move in the same direction. They can't get oil from outside, but can share oil with each other on the path, and that any of the vehicle may stop and wait the others at any stage. All of them must return back over the course from which they have come. The question is how to arrange the fleet to ensure one of vehicles reach the farthest, and all of them finally return to the starting position?

NVEP dates back to [1], in which it shows that the optimal solution of the problem is a replacement of sequence of 1 to n. We find that the essence of NVEP

---

[1]This paper is supported by Daqing Oilfield Limited Company Program and Key Laboratory of Management, Decision and Information Systems, CAS.
[2]Email Address: liuruoyang@amss.ac.cn
[4]Corresponding Author. Email Adress: cjc@amss.ac.cn

is a permutation problem, of which the computational complexity is $O(n!)$, so it is assigned to the NP class. For a special case, Li and Cui ([2]) designed a polynomial algorithm with $O(n^2)$ computational complexity to find the optimal solution. However, for most of the general NVEP, the computational complexity is still an exponential size. Li and Cui ([4]) proposed a real scheme for computing the approximate solution of NVEP with two kinds of heuristic algorithms. Xia and Cui ([5]) analyzed the possible complexity under different input conditions; Liang ([3]) gave a necessary condition of optimal arrangement; Xu and Cui ([6]) constructed a nonlinear 0-1 mixed integer programming model and designed a kind of ε – approximation algorithm with the method of penalizing function; in paper [7], the authors extended the single task NVEP to multi-task NVEP and proved it is strongly NP-hard.

By analyzing the properties of NVEP, we find a necessary condition for the optimal solution. Based on this condition, we design a heuristic algorithm which contains the greedy idea.

The paper is organized as follows. In section 2, we illustrate the general description of NVEP and propose two conclusions about its properties. We design the greedy heuristic algorithm in section 3 and analyze its computational complexity, which is about $O(n^4)$. In section 4, 10 numerical examples are calculated to compare the effectiveness of greedy heuristic algorithm with the two kinds of heuristic algorithm in paper [4] through the value of approximation ratio.

2 NVEP

2.1 Mathematical Models

Given a fleet of n vehicles, respectively denoted by $V_1, V_2, \cdots, V_n$, the oil capacity of V_i is a_i liters, and the oil efficiency of V_i is b_i liters per unit distance $(i = 1,2,\cdots,n)$. Let $\pi(1) \Rightarrow \pi(2) \Rightarrow \cdots \Rightarrow \pi(n)$ denote the order of vehicle number in terms of distance traveled from near to far, where $\pi(i)$ indicates the index of vehicle which traveled i th farthest, and $\pi(i) \Rightarrow \pi(j)$ shows that $V_{\pi(i)}$ provides oil to $V_{\pi(j)}$, which can be seen in *Fig.1*.

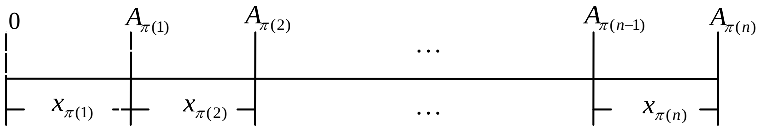

Figure 1.Order of oil supply relation of NVEP

In *Fig.1*, 0 is the starting and returning point. $A_{\pi(i)}$ denotes the farthest point that $V_{\pi(i)}$ can reach, $x_{\pi(i)}$ denotes the more distance that $V_{\pi(i+1)}$ runs than $V_{\pi(i)}$. During the exploration, $V_{\pi(i)}$ stops at point $A_{\pi(i)}$ and wait until other vehicles $V_{\pi(i+1)}, \cdots, V_{\pi(n)}$ return from farthest points. Then the vehicles share the left oil for their following back traveling. Set S_π be the range of the n vehicles fleet under the traveling order of $\pi(1) \Rightarrow \pi(2) \Rightarrow \cdots \Rightarrow \pi(n)$, then, we have:

$$S_\pi = x_{\pi(n)} + \cdots + x_{\pi(2)} + x_{\pi(1)}$$

$$\text{s.t.}\begin{cases} 2x_{\pi(n)}b_{\pi(n)} = a_{\pi(n)} \\ 2x_{\pi(n-1)}(b_{\pi(n-1)} + b_{\pi(n)}) + 2x_{\pi(n)}b_{\pi(n)} = a_{\pi(n-1)}a_{\pi(n)} \\ \cdots\cdots \\ 2x_{\pi(1)}(b_{\pi(1)} + b_{\pi(2)} + \cdots + b_{\pi(n)}) + 2x_{\pi(2)}(b_{\pi(2)} + \cdots + b_{\pi(n)}) + \cdots + 2x_{\pi(n)}b_{\pi(n)} = a_{\pi(1)} + a_{\pi(2)} + \cdots a_{\pi(n)} \end{cases}$$

By simplifying,

$$S_\pi = \frac{1}{2}\left(\frac{a_{\pi(n)}}{b_{\pi(n)}} + \frac{a_{\pi(n-1)}}{b_{\pi(n-1)} + b_{\pi(n)}} + \cdots + \frac{a_{\pi(1)}}{b_{\pi(1)} + b_{\pi(2)} + \cdots + b_{\pi(n)}}\right)$$

Hence, NVEP can be formulated as follows:

Definition.1 (The Permutation Model of NVEP)
Consider a set of n bi-vectors $AB = \{(a_1, b_1), (a_2, b_2), \cdots, (a_n, b_n)\}$, if a permutation $\pi(1), \pi(2), \cdots, \pi(n)$ of $\{1,2,\cdots,n\}$ could maximize the value of

$$S_\pi = \frac{1}{2}\left(\frac{a_{\pi(n)}}{b_{\pi(n)}} + \frac{a_{\pi(n-1)}}{b_{\pi(n-1)} + b_{\pi(n)}} + \cdots + \frac{a_{\pi(1)}}{b_{\pi(1)} + b_{\pi(2)} + \cdots + b_{\pi(n)}}\right)$$

Then the permutation $\pi(1), \pi(2), \cdots, \pi(n)$ is the optimal solution of NVEP.

From *Definition.1*, we conclude that the number of the feasible solution of NVEP is $n!$. The *brute-force search algorithm* can be used to find the optimal solution when n is small, however, when the size of problem is large, this method for exact solution becomes an exponential function of n.

2.2 Some Properties of NVEP

Property.1 (A Necessary Condition)
If the optimal solution of NVEP is $\pi = (\pi(1), \pi(2), \cdots, \pi(n))$, the optimal order of oil supply is $\pi(1) \Rightarrow \pi(2) \Rightarrow \cdots \Rightarrow \pi(n)$, then any two vehicles $V_{\pi(i)}$ and $V_{\pi(j)}$ which are adjacent must satisfy $\frac{a_{\pi(i)}}{(b_{\pi(i)} + B)b_{\pi(i)}} \le \frac{a_{\pi(i+1)}}{(b_{\pi(i+1)} + B)b_{\pi(i+1)}}$, in which a_i and b_i respectively denotes the oil capacity and efficiency of V_i, $B = \sum_{k=\pi(i+2)}^{\pi(n)} b_k$.

Proof. Given the optimal solution $\pi = (\pi(1), \pi(2), \cdots, \pi(n))$, the farthest distance is

$$S_\pi = \frac{1}{2}\left(\frac{a_{\pi(1)}}{b_{\pi(1)} + b_{\pi(2)} + \cdots + b_{\pi(n)}} + \cdots + \frac{a_{\pi(i)}}{b_{\pi(i)} + b_{\pi(i+1)} + B} + \frac{a_{\pi(i+1)}}{b_{\pi(i+1)} + B} + \cdots + \frac{a_{\pi(n)}}{b_{\pi(n)}}\right)$$

Change the order of any two vehicles $V_{\pi(i)}$ and $V_{\pi(j)}$ which are adjacent, we have $\pi(1) \Rightarrow \cdots \Rightarrow \pi(i+1) \Rightarrow \pi(i) \Rightarrow \cdots \Rightarrow \pi(n)$.

Then the farthest distance under this order is

$$S_0 = \frac{1}{2}\left(\frac{a_{\pi(1)}}{b_{\pi(1)}+b_{\pi(2)}+\cdots+b_{\pi(n)}}+\cdots+\frac{a_{\pi(i+1)}}{b_{\pi(i+1)}+b_{\pi(i)}+B}+\frac{a_{\pi(i)}}{b_{\pi(i)}+B}+\cdots+\frac{a_{\pi(n)}}{b_{\pi(n)}}\right)$$

for π is the optimal solution, then we have $S_\pi \ge S_0$, that is

$$S_\pi - S_0 = \frac{1}{2}\left(\frac{a_{\pi(i+1)}b_{\pi(i)}}{\left(b_{\pi(i+1)}+B\right)\left(b_{\pi(i+1)}+b_{\pi(i)}+B\right)}-\frac{a_{\pi(i)}b_{\pi(i+1)}}{\left(b_{\pi(i)}+B\right)\left(b_{\pi(i)}+b_{\pi(i+1)}+B\right)}\right)\ge 0$$

By simplifying, we obtain

$$\frac{a_{\pi(i+1)}}{\left(b_{\pi(i+1)}+B\right)b_{\pi(i+1)}} \ge \frac{a_{\pi(i)}}{\left(b_{\pi(i)}+B\right)b_{\pi(i)}} \qquad \square$$

Corollary.1 Under the optimal order of the oil supply $\pi(1)\Rightarrow\pi(2)\Rightarrow\cdots\Rightarrow\pi(n)$, the vehicle which runs the farthest and the vehicle which last supply the oil satisfy $\frac{a_{\pi(n-1)}}{b^2_{\pi(n-1)}} \le \frac{a_{\pi(n)}}{b^2_{\pi(n)}}$.

3 The Greedy Heuristic Algorithm

3.1 Designing the Algorithm

Based on *Property.1*, we construct the heuristic algorithm with the greedy idea through n-1 phases.

Phase.1 Given a fleet of n vehicles, of which respectively have a_i oil capacity and b_i efficiency per unit distance. Considering the *Property.1*, we construct the paired comparison matrix M_n for any two vehicles of the fleet, and choose the index of the smaller value as the element of M_n, that is,

$$M_n = \begin{pmatrix} (1,1) & (1,2) & \cdots & (1,n) \\ (2,1) & (2,2) & \cdots & (2,n) \\ \vdots & \vdots & \ddots & \vdots \\ (n,1) & (n,2) & \cdots & (n,n) \end{pmatrix}$$

In which, the element $(i,j)=\begin{cases} i & if\ \dfrac{a_i}{(b_i+B_n)b_i} \le \dfrac{a_j}{(b_j+B_n)b_j} \\ j & if\ \dfrac{a_i}{(b_i+B_n)b_i} > \dfrac{a_j}{(b_j+B_n)b_j} \end{cases}$, $B_n = \sum_{k\in S_n} b_k$, $S_n = N_1 \setminus \{i,j\}$, $N_1 = \{1,2,\cdots,n\}$. We can see that M_n is a symmetric matrix. For the i th vehicle, we set $I_n(i)$ as the frequency that i appears in the row i ($i=1,2,\cdots,n$). Set $\pi(1)=i_1$, which satisfies $I_n(i_1)=\max\{I_n(i) \mid i\in N_1\}$, it means that we choose the vehicle which maximizes I_n in set N_1 as the vehicle stopping firstly.

Phase.*k* We construct M_{n-k+1} for the rest of $n-k+1$ vehicles, and $N_k = N_{k-1}\setminus\{i_{k-1}\}$, in which $2 \le k \le n-2$. For the the i th vehicle, we set $I_{n-k+1}(i)$ as the frequency that i appears in the row i, where $i \in N_k$. Set $\pi(k) = i_k$, which satisfies $I_{n-k+1}(i_k) = \max\{I_{n-k+1}(i) | i \in N_k\}$, as the index of the stopping vehicle at this step.

Phase.*n*-1 For the rest of two vehicles, we use *Corollary.1* to decide the order of supplying oil. Then we can obtain the approximate solution, which is displayed as $i_1 \Rightarrow i_2 \Rightarrow \cdots \Rightarrow i_n$.

The algorithm scheme is described as follows:

The Greedy Heuristic Algorithm

Input: n, a_i, b_i, $i = 1,2,\cdots,n$

Step1: Initially $N_1 = \{1,2,\cdots,n\}$ and $k = 1$ .If $n = 1$ stop; Else if $n = 2$ goto step3.

Step2: If $k = n-1$ then goto step3;else if $k = n$,stop;

Else compute M_k, and

for each $i \in N_k$,compute $I_{n-k+1}(i)$, set $i_k = \arg\{\max\{I_{n-k+1}(i) | i \in N_k\}\}$.

$k = k+1$, then goto step2;

Step3: For i, j, compute a_i / b_i^2 and a_j / b_j^2.

If $a_i / b_i^2 \le a_j / b_j^2$, set $i_{n-1} = i$, $i_n = j$;else set $i_{n-1} = j$, $i_n = i$.

Output: $i_1 \Rightarrow i_2 \Rightarrow \cdots \Rightarrow i_n$

3.2 Complexity Analysis

According to the *Property.1* and the constructing phases, the heuristic algorithm needs to be circulated about n times. With the greedy idea, we choose the vehicle that maximizes the frequency of satisfying the necessary condition as the stopping vehicle at each phase. By this way, we could obtain an approximate solution for NVEP, however, it cannot ensure the approximation ratio of solution.

As discussed above, we can see that the complexity is mainly in constructing the comparison matrix. Assume one time of addition or multiplication as a basic operation. Determining the elements of M_n needs n^2 operations, and each element needs $n+5$ operations (including n times of addition, four times of multiplication and once comparison), meanwhile, we can use some technologies to neglect the cost of determining the largest I_n, therefore, it needs $n^2(n+5)$ times of basic operation at the first step. Successively, the k th phase requires $(n-k+1)^2(n+6-k)$ times of basic operation. The operation times at the last phase are five. To sum up, the total computation times are $5 + \sum_{k=1}^{n-2}(n-k+1)^2(n+6-k) \approx O(n^4)$. Therefore, the computation complexity of the algorithm is $O(n^4)$.

4 Numerical Example

In this section, we compare our greedy heuristic algorithm with two kinds of *heuristic algorithm* in paper [4], using 10 numerical examples selected from [4]. In *Table1*, we list the optimal solution found by *brute-force search algorithm*, and the approximate solution with the greedy heuristic algorithm. Then the compared results of effectiveness are showed in *Table2*.

Table1.The exact and the approximate solution with greedy heuristic algorithm

		1	2	3	4	5	6	7	8	9	10
Ex.1	*a*	70	76	84	70	70	79	87	71	74	72
	b	7.5	8.6	9.6	6.4	7.2	9	10	7.8	7	8
	Optimal.	$7 \Rightarrow 3 \Rightarrow 6 \Rightarrow 2 \Rightarrow 10 \Rightarrow 8 \Rightarrow 1 \Rightarrow 5 \Rightarrow 9 \Rightarrow 4$									
	Approx.	$7 \Rightarrow 3 \Rightarrow 6 \Rightarrow 2 \Rightarrow 10 \Rightarrow 8 \Rightarrow 1 \Rightarrow 5 \Rightarrow 9 \Rightarrow 4$									
Ex.2	*a*	6	9	12	15	18	21	24	32	38	45
	b	4	5	6	7	8	9	10	13	15	17
	Optimal.	$1 \Rightarrow 3 \Rightarrow 9 \Rightarrow 10 \Rightarrow 8 \Rightarrow 7 \Rightarrow 6 \Rightarrow 5 \Rightarrow 4 \Rightarrow 2$									
	Approx.	$1 \Rightarrow 2 \Rightarrow 3 \Rightarrow 4 \Rightarrow 5 \Rightarrow 9 \Rightarrow 10 \Rightarrow 8 \Rightarrow 7 \Rightarrow 6$									
Ex.3	*a*	20	30	40	89	78	67	46	38	90	35
	b	7	9	27	8	11	16	10	20	13	15
	Optimal.	$3 \Rightarrow 8 \Rightarrow 10 \Rightarrow 1 \Rightarrow 2 \Rightarrow 6 \Rightarrow 7 \Rightarrow 9 \Rightarrow 5 \Rightarrow 4$									
	Approx.	$3 \Rightarrow 8 \Rightarrow 10 \Rightarrow 1 \Rightarrow 2 \Rightarrow 6 \Rightarrow 7 \Rightarrow 9 \Rightarrow 5 \Rightarrow 4$									
Ex.4	*a*	51	60	62	65	120	135	140	150	155	
	b	9	7	10	8	30	26	28	32	27	
	Optimal.	$5 \Rightarrow 8 \Rightarrow 7 \Rightarrow 6 \Rightarrow 9 \Rightarrow 1 \Rightarrow 3 \Rightarrow 4 \Rightarrow 2$									
	Approx.	$5 \Rightarrow 8 \Rightarrow 7 \Rightarrow 6 \Rightarrow 9 \Rightarrow 1 \Rightarrow 3 \Rightarrow 4 \Rightarrow 2$									
Ex.5	*a*	51	60	62	65	120	135	140	150	155	68
	b	9	7	10	8	30	26	28	32	27	8
	Optimal.	$5 \Rightarrow 8 \Rightarrow 7 \Rightarrow 6 \Rightarrow 9 \Rightarrow 1 \Rightarrow 3 \Rightarrow 4 \Rightarrow 10 \Rightarrow 2$									
	Approx.	$5 \Rightarrow 8 \Rightarrow 7 \Rightarrow 6 \Rightarrow 9 \Rightarrow 1 \Rightarrow 3 \Rightarrow 4 \Rightarrow 10 \Rightarrow 2$									
Ex.6	*a*	51	60	62	65	120	135	140	150	155	55
	b	9	7	10	8	30	26	28	32	27	12
	Optimal.	$5 \Rightarrow 8 \Rightarrow 7 \Rightarrow 6 \Rightarrow 9 \Rightarrow 10 \Rightarrow 1 \Rightarrow 3 \Rightarrow 4 \Rightarrow 2$									
	Approx.	$5 \Rightarrow 8 \Rightarrow 7 \Rightarrow 6 \Rightarrow 9 \Rightarrow 10 \Rightarrow 1 \Rightarrow 3 \Rightarrow 4 \Rightarrow 2$									
Ex.7	*a*	130	60	62	65	120	135	140	150	155	68
	b	30	7	10	8	30	26	28	32	27	8
	Optimal.	$5 \Rightarrow 1 \Rightarrow 8 \Rightarrow 7 \Rightarrow 6 \Rightarrow 9 \Rightarrow 3 \Rightarrow 4 \Rightarrow 10 \Rightarrow 2$									
	Approx.	$5 \Rightarrow 1 \Rightarrow 8 \Rightarrow 7 \Rightarrow 6 \Rightarrow 9 \Rightarrow 3 \Rightarrow 4 \Rightarrow 10 \Rightarrow 2$									
Ex.8	*a*	12	23	11	26	16	31	19	21	44	25
	b	2	4	3	5	3.4	5	4.3	7	7.8	4.5
	Optimal.	$8 \Rightarrow 3 \Rightarrow 7 \Rightarrow 5 \Rightarrow 9 \Rightarrow 4 \Rightarrow 10 \Rightarrow 6 \Rightarrow 2 \Rightarrow 1$									
	Approx.	$8 \Rightarrow 3 \Rightarrow 7 \Rightarrow 5 \Rightarrow 9 \Rightarrow 4 \Rightarrow 10 \Rightarrow 6 \Rightarrow 2 \Rightarrow 1$									
Ex.9	*a*	65	120	135	140	150	155	55	100	122	60
	b	8	30	26	28	32	27	12	10	11	4
	Optimal.	$2 \Rightarrow 5 \Rightarrow 4 \Rightarrow 3 \Rightarrow 6 \Rightarrow 7 \Rightarrow 1 \Rightarrow 8 \Rightarrow 9 \Rightarrow 10$									
	Approx.	$2 \Rightarrow 5 \Rightarrow 4 \Rightarrow 3 \Rightarrow 6 \Rightarrow 7 \Rightarrow 1 \Rightarrow 8 \Rightarrow 9 \Rightarrow 10$									
Ex.10	*a*	32	42	55	65	72	80	83	84	95	100
	b	4	4.8	5.5	7.8	11	9.6	6.7	15	8.6	16

	Optimal.	$8 \Rightarrow 10 \Rightarrow 5 \Rightarrow 6 \Rightarrow 4 \Rightarrow 1 \Rightarrow 2 \Rightarrow 9 \Rightarrow 3 \Rightarrow 7$
	Approx.	$8 \Rightarrow 10 \Rightarrow 5 \Rightarrow 6 \Rightarrow 4 \Rightarrow 1 \Rightarrow 2 \Rightarrow 9 \Rightarrow 3 \Rightarrow 7$

Note: In Table.1, the symbol (a,b) denotes the oil capacity and efficiency of the vehicle respectively, and the symbol $\Rightarrow$ represents the oil supply relation.

Table2.The compared results of effectiveness with the heuristic algorithms ([4]).

	Exact	Heuristic.1		Heuristic.2		Greedy Heuristic	
	$S_{\max}$	$S_{\max}$	Appro. Ratio	$S_{\max}$	Appro. Ratio	$S_{\max}$	Appro. Ratio
Ex1	15.4766	15.3166	99.00%	15.4766	100%	15.4766	100%
Ex2	3.6033	3.5113	97.50%	3.5387	98.20%	3.4267	95.10%
Ex3	11.0236	10.968	99.50%	11.0236	100%	11.0236	100%
Ex4	11.9466	11.7532	98.40%	11.8345	99.10%	11.9466	100%
Ex5	12.723	12.3035	96.70%	12.4895	98.20%	12.723	100%
Ex6	12.124	11.8963	98.10%	11.9735	98.80%	12.124	100%
Ex7	12.7697	12.4763	97.70%	12.7697	100%	12.7697	100%
Ex8	9.4293	9.354	99.20%	9.3774	99.50%	9.4293	100%
Ex9	18.2772	18.2303	99.70%	18.2363	99.80%	18.2772	100%
Ex10	15.581	14.7718	94.80%	15.4277	99.01%	15.581	100%

From the results in *Table1* and *Table2*, we can see that the greedy heuristic algorithm can get exact solution for most of the examples, and for the example which couldn't find the exact solution, the algorithm also have an approximation ratio greater than 95%. Unfortunately, so far, we can neither prove nor give an approximate bound for our algorithm. From the aspect of complexity, the two heuristic algorithms already existed respectively has $O(n^2)$ and $O(n^3)$ of complexity, while the complexity of our heuristic algorithm is $O(n^4)$, but this difference have little effect on the algorithm effectiveness by using the fast computer. In conclusion, the greedy heuristic algorithm we design in this article has a better effectiveness.

5 Conclusion

In this paper, we introduce a kind of exploitation problem, which is called N-Vehicle Exploration Problem, and we take NVEP for short. And through studying the properties of the optimal solution of NVEP, we obtain a necessary condition for NVEP. Then, based on this property and combined with greedy idea, we design a kind of new heuristic algorithm, of which the computation complexity is $O(n^4)$. There have been two kinds of heuristic algorithms already existed in paper [4] for NVEP, respectively with $O(n^2)$ and $O(n^3)$ of complexity. In order to find the differences among these algorithms, we choose 10 numerical examples from paper [4] to compare the effectiveness of these algorithms. The compared results show that the greedy heuristic algorithm can get exact solution for most of the examples and have a better effectiveness.

References

[1] Rui Yang. A Problem with Oil Traffic and its Promotion.. Mathematical Communication, 1999, 9: 44-45 (in Chinese).

[2] Xiao-Ya Li, Jin-Chuan Cui. An Efficient Algorithm about a Kind of N Vehicle Problem. J. System Engineering, 2008, 23(4): 444-448 (in Chinese).

[3] Li-Li Liang. A Problem of Permutation Optimization. Journal of Guangxi University for Nationalities, 2006, 12(4), 72-76 (in Chinese).

[4] Xiao-Ya Li, Jin-Chuan Cui. Real-Time Algorithm Scheme for *n*-Vehicle Exploration Problem. Combinatorial Optimization And Applicationas, Proceedings, 2009, 5573: 287-300.

[5] Xi Xia, Jin-Chuan Cui. A Method of Estimating Computational Complexity Based on Input Conditions for N-vehicle Problem. Acta Mathmaticae Applicatae Sinica, English Series, 2010, 26: 1-12

[6] Yang-Yang Xu, Jin-Chuan Cui. A Kind of ε-approximation Algorithm for the Exploration Problem with N vehicles. Acta Mathematicae Applicatae Sinica, 2009, 32: 1036-1043 (in Chinese).

[7] Yang-Yang Xu, Jin-Chuan Cui. Multitask N-Vehicle Exploration Problem: Complexity and Algorithm. Journal System Science & Complexity, 2009, 20: 1-14.

[8] Jon Kleinberg, Eva Tardos. Algorithm Design. Addison Wesley: United States ED edition, 2005, 3.