

Liangjun Kea;Claudia Arch ettib;Zuren Feng

Ano de publicação - 2007

Ants can solve the team orienteering problem

Computers & Industrial Engineering

Volume 54, Issue 3, April 2008, Pages 648-665

ABSTRACT

O TOP envolve encontrar um conjunto de caminhos desde um ponto inicial até um ponto final tal que o total de prémios recebidos por visitar um subconjunto de locais é maximizado e o comprimento de cada caminho é restrito por um limite pré-especificado.

Neste artigo, uma optimização ACO da colónia é proposta para o TOP, quatro métodos (sequencial, determinístico-concorrente e aleatório concorrente e simultâneo,são propostos para construir candidatos a soluções na estrutura.Comparamos estes métodos de acordo com os resultados obtidos em problemas de literatura famosos. Finalmente comparamos o algoritmo com vários outros existentes.

Os resultados mostram que o nosso é promissor.

RESUMO DO ARTIGO

O projecto TOP envolve encontrar um conjunto de caminhos entre dois pontos de modo a maximizar a recompensa total arrecadada por visita a determinado subconjunto. O tempo para percorrer cada conjunto está limitado por um limite pré-especificado.

Este artigo fala-nos da optimização de uma colónia de formigas (problema de orientação de equipa).

No âmbito deste problema temos quatro métodos possíveis, sendo eles o método sequencial, determinista, aleatório concorrente e simultâneo concorrente.

Uma equipa de veículos tenta visitar um conjunto de pontos aos quais estão atribuídos recompensas, começando num ponto de partida e acaba num ponto final dentro de um prazo estabelecido. Um veículo visita um determinado ponto e recebe a recompensa no entanto todos os veículos que visitem esse ponto depois do primeiro não recebem qualquer tipo de recompensa. O algoritmo TOP é uma variante do caixeiro viajante. Existem apenas dois algoritmos exactos propostos até agora para o projecto TOP, o algoritmo de geração de colunas baseado e algoritmo de ramo preço baseado. Estes dois algoritmos são limitados em tamanho e tempo.

No projecto ACO, abordado por nós mais profundamente, o ponto-chave é a construção de soluções candidatas, isto é, uma formiga deve determinar qual veículo para mover e por onde mover.

ACO é uma classe base populacional mataheurísticos. Neste artigo é utilizada uma colónia de formigas que são guiadas por trilhos de feromonas e informação heurística para soluções iterativas. Depois de todos os trilhos de feromonas e os parâmetros inicializados, as formigas vão construindo soluções iterativamente até ser atingido o critério de paragem.

O processo principal de iteração baseia-se em dois passos, primeiro passo cada formiga constroi uma solução de acordo com a regra de transição no segundo passo é adoptado um procedimento de busca local para melhorar uma ou mais soluções, os valores das feromonas são actualizados de acordo com uma regra de actualização de feromonas.

Temos um grafo completo $G=(V,E)$ em que V é o conjunto de todos os vértices do

grafo $V=\{1,...,n\}$ e E o conjunto de arestas $E=\{(i,j) \mid i,j \in V\}$, cada vértice tem a si atribuído uma recompensa r_i , todos os caminhos começam no vértice 1 e acabam no vértice n . As recompensas nos pontos inicial e final são zero, isto é, $r_1=r_n=0$. Para cada aresta (i,j) é associado um custo não negativo c_{ij} onde c_{ij} é a distância entre i e j . TOP consiste encontrar caminhos m a começar no vértice 1 e acabar no vértice n de maneira a que a recompensa em cada vértice seja maximizada. Cada veículo só é recompensado se for o primeiro a passar nesse vértice. O tempo total de visita não pode exceder o tempo pré-especificado T_{max} .

Para cada veículo assumimos que existe uma proporcionalidade directa entre a distância percorrida e o tempo consumido pelo veículo, logo o T_{max} pode ser distância ou tempo, neste artigo T_{max} é o valor da distância máximo.

VARIÁVEIS DE DECISÃO

Sendo $y_{ik}=1$ ($i=1,...,n$; $k=1,...,m$) se o vértice i for visitado pelo veículo k e $y_{ik}=0$ caso contrário; e $x_{ijk}=1$ se corresponder a uma aresta visitada e $x_{ijk}=0$ caso contrário. Desde $c_{ij} = CJI$, apenas x_{ijk} ($i < j$) são definidos. Seja U um subconjunto de V .

MODELO DE PROGRAMAÇÃO LINEAR (INTEIRA)

$$\max \sum_{i=2}^{n-1} \sum_{k=1}^m r_i y_{ik} \quad (1)$$

$$\text{sujeito a } \sum_{j=2}^n \sum_{k=1}^m x_{ijk} = \sum_{i=1}^{n-1} \sum_{k=1}^m x_{ink} = m \quad (2)$$

$$\sum_{i < j} x_{ijk} + \sum_{i > j} x_{ijk} = 2y_{jk} \quad (j=2,...,n-1; k=1,...,m) \quad (3)$$

$$\sum_{k=1}^m y_{ik} \leq 1 \quad (i=2,...,n-1) \quad (4)$$

$$\sum_{i=1}^{n-1} \sum_{j>i} c_{ij} x_{ijk} \leq T_{max} \quad (k=1,...,m) \quad (5)$$

$$\sum_{i,j \in U, i < j} \leq |U|-1 \quad (U \subset V \setminus \{1,n\}; 2 \leq |U| \leq n-2; k=1,...,m) \quad (6)$$

$$x_{ijk} \in \{0,1\} \quad (1 \leq i < j \leq n; k=1,...,m) \quad (7)$$

$$y_{1k} = y_{nk} = 1, y_{ik} \in \{0,1\} \quad (i=2,...,n-1; k=1,...,m) \quad (8)$$

Restrições:

- (2) Começa em 1 acaba em n
- (3) Garante a conectividade do caminho
- (4) Todos os vértices (excepto 1 e n) podem ser visitadas no máximo uma vez.
- (5) Restrição de tempo
- (6) Sub-caminhos proibidos
- (7) (8) Requisito integrante em cada variável

RESULTADOS COMPUTACIONAIS

Os resultados do desempenho de ACO-TOP encontram-se em anexo.

O autores usaram um algoritmo em C++ e o hardware utilizado foi um pc com 3,0 GHZ Intel CPU.

As experiências computacionais foram feitas num conjunto de 387 casos de referencia divididos em sete sets, tendo cada um deles 32, 21, 33, 100, 66, 64 e 102 vertices. A recompensa de cada vertice é idêntica em todos as instâncias de cada conjunto tendo cada conjunto três grupos e cada um deles número diferente de veiculos. Um dos três grupos e caraterizado por um T_{max} diferente.

REFERÊNCIAS

O artigo apresentado tem 22 referência sendo a mais antiga datada de 1980 e a mais recente 2007.

ANEXOS

Resultados da comparação entre os quatro metodos

Group	Sequential			Deterministic-concurrent			Random-concurrent			Simultaneous		
	Best	Aver	Time	Best	Aver	Time	Best	Aver	Time	Best	Aver	Time
1.2	149.1	149.1	5.2	149.1	149.1	4.9	149.1	149.1	4.7	149.1	149.1	4.9
1.3	125.0	125.0	5.8	125.0	125.0	5.2	125.0	125.0	4.9	125.0	125.0	5.2
1.4	101.0	101.0	6.4	101.0	101.0	5.7	101.0	101.0	5.1	101.0	101.0	5.7
2.2	190.5	190.5	2.9	190.5	190.5	2.8	190.5	190.5	2.8	190.5	190.5	2.8
2.3	136.4	136.4	3.1	136.4	136.4	3.0	136.4	136.4	2.8	136.4	136.4	2.9
2.4	94.5	94.5	3.4	94.5	94.5	3.2	94.5	94.5	2.8	94.5	94.5	3.2
3.2	496.0	496.0	6.1	496.0	495.8	5.8	496.0	495.8	5.7	496.0	496.0	5.8
3.3	411.5	411.4	6.5	411.5	411.2	6.0	411.5	411.1	5.7	411.5	411.2	6.0
3.4	336.5	336.5	6.9	336.5	336.5	6.3	336.5	336.2	5.8	336.5	336.3	6.3
4.2	915.6	899.8	34.1	908.4	898.3	30.6	909.5	897.3	30.1	911.8	899.6	30.8
4.3	853.8	841.1	37.2	847.7	841.1	31.6	848.4	840.1	30.5	848.2	841.9	31.8
4.4	798.1	783.0	40.5	795.9	784.1	33.4	791.4	780.4	31.8	795.2	787.3	33.8
5.2	897.6	891.1	15.6	896.4	890.5	14.0	896.2	892.2	13.7	896.2	890.7	14.1
5.3	782.8	775.3	17.3	780.4	774.6	15.0	781.2	774.5	14.2	781.0	774.2	15.0
5.4	708.8	704.6	19.3	707.7	698.5	16.2	706.3	698.2	15.1	705.6	698.2	16.3
6.2	819.3	815.4	14.8	818.7	814.1	13.3	818.7	814.8	13.0	819.3	815.7	13.4
6.3	792.8	786.0	16.7	790.5	785.6	14.5	790.5	784.7	13.8	791.3	786.3	14.6
6.4	714.0	701.3	18.2	714.0	699.1	15.2	714.0	699.5	14.2	714.0	702.0	15.3
7.2	642.7	637.4	26.0	641.5	637.7	22.2	641.0	637.1	21.5	641.2	637.4	22.1
7.3	599.9	597.1	30.8	599.4	595.5	24.8	598.6	594.6	23.7	599.2	596.0	25.0
7.4	519.1	517.0	34.5	518.2	516.3	26.8	518.4	516.2	24.9	518.4	516.3	26.9

Comparação experimental com vários algoritmos existentes

	ACO-TOP	CGW	TMH	GTP	GTF	FVF	SVF
Set 1	7.9	15.4	NA	10.0	5.0	1.0	22.0
Set 2	3.8	0.9	NA	0.0	0.0	0.0	1.0
Set 3	8.5	15.4	NA	10.0	9.0	1.0	19.0
Set 4	51.1	934.8	796.7	612.0	324.0	121.0	1118.0
Set 5	25.2	193.7	71.3	147.0	105.0	30.0	394.0
Set 6	20.3	150.1	45.7	96.0	48.0	20.0	310.0
Set 7	44.7	841.4	432.6	582.0	514.0	90.0	911.0

Legenda:

CGW: a heurística de cinco passos propostos por Chao et al.(1996a);

TMH: a heurística busca tabu proposto por Tang e Miller-Hooks (2005);

GTP: a busca tabu com a estratégia de pena proposta pelo Archetti et al. (2007);

GTF: a busca tabu com a estratégia viável proposto por Archetti et al. (2007);

FVF: a busca rápida da vizinhança variável proposta por Archetti et al. (2007);

SVF: a busca bairro lenta variável proposta por Archetti et al.(2007) 0,1 Difere da FVF na parametrização.

Recompensa melhor (média) de cada grupo

Problem name	<i>n</i>	<i>m</i>	<i>T</i> _{max}	ACO-TOP	Previous best
p4.2.j	100	2	70.0	965	962
p4.2.p			100.0	1242	1241
p4.2.r			110.0	1288	1286
p4.2.s			115.0	1304	1301
p4.3.q	3	70.0	1252	1251	
p4.3.t		80.0	1305	1304	
p5.2.y	66	2	62.5	1645	1635
p7.2.i	102	2	90.0	580	579
p7.2.j		100.0	646	644	
p7.3.l	3	80.0	684	683	
p7.3.p		106.7	929	927	
p7.3.t		133.3	1118	1117	

