APLICAÇÃO DO ALGORITMO GENÉTICO DE CHAVES ALEATÓRIAS VICIADAS AO PROBLEMA DE ESCALONAMENTO DE TÉCNICOS DE CAMPO

Ricardo de Brito Damm
Departamento de Engenharia de Produção da Escola Politécnica, USP
rbdamm@usp.br

Débora Pretti Ronconi
Departamento de Engenharia de Produção da Escola Politécnica, USP
dronconi@usp.br

RESUMO

Um tema ainda pouco estudado na literatura é o problema de escalonamento de técnicos de campo, que consiste em programar um conjunto de tarefas que devem ser executadas por um grupo de técnicos. As tarefas estão distribuídas em uma região e têm diferentes prioridades e janelas de tempo. Os técnicos possuem habilidades distintas e diferentes horários de trabalho; cada tarefa é atendida por um único técnico e, como objetivo, procura-se principalmente maximizar o número ponderado de tarefas executadas em um dia, de acordo com as suas prioridades. O trabalho apresenta um modelo de programação linear inteira mista (PLIM) e, dada a complexidade do problema, foi aplicado um Algoritmo Genético denominado *Biased Random-Key Genetic Algorithms* (BRKGA), que utiliza chaves aleatórias para codificar e decodificar as soluções. Os resultados computacionais obtidos e as comparações com os limitantes superiores indicaram um bom desempenho do BRKGA.

PALAVARAS-CHAVE: programação de técnicos, roteirização, janelas de tempo, *Biased Random-Key Genetic Algorithms* (BRKGA).

1. Introdução

Um tópico frequentemente encontrado em empresas prestadoras de serviço é o problema de escalonamento de técnicos de campo (PETC), que consiste em associar um número de tarefas (em diversos locais de uma região, com diferentes prioridades e com janelas de tempo) a uma quantidade de técnicos (com diferentes horários de expediente e com habilidades distintas), que devem retornar para o local de origem no final do expediente. Cada tarefa é atendida por um único técnico.

Entre os primeiros autores que estudaram o PETC estão Tsang e Voudouris (1997) e Xu e Chiu (2001). Em 2007, a Sociedade de Pesquisa Operacional da França (*French Operational Society*) e a France Telecom lançaram este problema como um desafio e os trabalhos de Cordeau et al. (2010) e Hashimoto et al. (2011) foram premiados. Em outra importante publicação, Kovacs et al. (2011) estudou uma extensão da pesquisa dos dois artigos anteriores.

Recentemente, Pillac et al. (2012) analisaram a similaridade entre o PETC e o problema de roteamento de veículos com janelas de tempo (PRVJT). De fato, segundo Kovacs et al. (2011), Xu & Chiu (2001) e Pillac et al. (2012), o PETC é uma generalização do PRVJT, que é NP-difícil, e não poderá ser resolvido em tempo polinomial por métodos exatos.

O trabalho apresenta um modelo de programação linear inteira mista (PLIM) na seção 2 e, dada a complexidade do problema, um Algoritmo Genético foi aplicado ao problema (seção 3). A seção 4 apresenta os testes computacionais e a seção 5, as conclusões e futuros passos.

2. Descrição do problema e modelo de programação linear inteira mista

Neste trabalho, PETC será modelado da seguinte maneira: seja $J = \{1,..., n\}$ um conjunto de tarefas ou serviços e $K = \{1,...,m\}$ um conjunto de técnicos disponíveis para executá-los. Cada tarefa $i \in J$ tem duração de tempo estimada em p_i, uma prioridade w_i e deve ser iniciada e concluída dentro de um intervalo de horário específico $[e_i, l_i]$. As tarefas estão distribuídas dentro de uma cidade e c_{ij} será o tempo de deslocamento do local i para o local j, com $i, j \in J \cup 0$, onde 0 representa a origem ou a sede da empresa.

Cada técnico $k \in K$ tem seu horário de trabalho diário definido por $[a_k, b_k]$. A habilidade de um técnico k executar uma tarefa i é dada por s_{ik}, que pode ser 0 ou 1, onde 1 indica ser apto e 0, incapaz de executá-la. O objetivo será alocar o maior número ponderado de tarefas e, secundariamente, maximizar a ociosidade dos funcionários ao retornar à origem (no fim do dia).

O modelo tem quatro tipos de variáveis: y_{ik} é uma variável binária que assume 1 quando uma tarefa i foi associada a um técnico k e 0, caso contrário; x_{ijk} também é uma variável binária com valor 1 se uma tarefa j é executada imediatamente após a tarefa i pelo técnico k. A variável t_i indica o horário de início da execução da tarefa i e z_k, o tempo ocioso de cada técnico após retornar à base.

A seguir é apresentada a formulação linear inteira mista, baseada em Xu e Chiu (2001) para o problema estudado.

$$\max \sum_{i \in J} \sum_{k \in K} \frac{w_i s_{ik} y_{ik}}{MW} + \sum_{k \in K} \frac{z_k}{MZ} \qquad (1)$$

Sujeito a:

$$\sum_{k \in K} y_{ik} \leq 1 \qquad i \in J \qquad (2)$$

$$\sum_{j \in J \cup \{0\} \setminus \{i\}} x_{ijk} = \sum_{j \in J \cup \{0\} \setminus \{i\}} x_{jik} = y_{ik} \qquad k \in K, i \in J \qquad (3)$$

$$\sum_{i \in J} x_{0ik} \leq 1 \qquad k \in K \qquad (4)$$

$$e_i \leq t_i \leq l_i - p_i \sum_{k \in K} y_{ik} \qquad i \in J \qquad (5)$$

$$t_i + p_i + c_{ij} \leq t_j + M\left(1 - x_{ijk}\right) \qquad k \in K, i \neq j \in J \qquad (6)$$

$$a_k + c_{0j} \leq t_j + M\left(1 - x_{0jk}\right) \qquad k \in K, j \in J \qquad (7)$$

$$t_i + p_i + c_{i0} \leq b_k - z_k + M\left(1 - x_{i0k}\right) \qquad k \in K, i \in J \qquad (8)$$

$$0 \leq z_k \leq b_k - a_k \qquad k \in K \qquad (9)$$

$$x_{ijk}, y_{ik} \in \{0,1\} \qquad i, j \in J, k \in K \qquad (10)$$

$$t_i, z_k \in R^+ \qquad i \in J, k \in K \qquad (11)$$

Onde:

$$M > \max\left(\max_{i \in J} l_i, \max_{k \in K} a_k\right) + \max_{i, j \in J \cup \{0\}} c_{ij} \qquad (12)$$

$$MW = min\left(w_i s_{ik}\right) \qquad i \in J, k \in K, s_{ik} \neq 0 \qquad (13)$$

$$MZ = max\left(b_k - a_k\right) + \varepsilon \qquad k \in K \qquad (14)$$

Na função objetivo (1), o primeiro termo maximiza a soma ponderada das tarefas realizadas e o segundo, a soma dos tempos de disponibilidade dos técnicos após o retorno à base. Os denominadores têm duas funções: tornar adimensionais os valores somados e garantir que sempre será melhor executar uma tarefa, reduzindo a ociosidade de um técnico, e nunca o contrário. Ao mesmo tempo, o objetivo secundário procurará minimizar o tempo de trabalho externo dos técnicos e, consequentemente, reduzir o tempo de deslocamento total e os tempos de espera para o início da execução de uma tarefa.

A restrição (2) evita que uma tarefa seja designada a mais de um técnico e a (3) garante que haverá apenas uma antecessora e uma sucessora a uma tarefa programada para um técnico. Cada técnico deve deixar a origem, no máximo, uma vez, como assegura a restrição (4). A restrição (5) garante que as janelas de tempo das tarefas serão respeitadas. As restrições (6) a (8) vincula o início da execução de cada tarefa com o término da sua antecessora, os tempos de deslocamento e o horário de trabalho do técnico; ao mesmo tempo, essas restrições previnem o *subtour*. O tempo disponível (ou ocioso) máximo de um técnico k após o retorno à origem é determinado pela restrição (9). As restrições (10) e (11) estipulam os tipos de variáveis e as restrições (12) a (14), as constantes M, MZ e MW.

3. Algoritmo Genético com chaves aleatórias viezadas

O Algoritmo Genético com chaves aleatórias, denominado *Random-Key Genetic Algorithms* (RKGA), foi proposto por Bean (1994). Trata-se de um AG específico para problemas de otimização combinatória que garante a factibilidade das novas soluções geradas ao longo da busca. Alguns anos depois, foram introduzidos novos conceitos a esse AG, aumentando o protagonismo das melhores soluções (as mais aptas) de cada geração. Denominada por Gonçalves e Resende (2011) como *Biased Random-Key Genetic Algorithms* (BRKGA), essa nova versão do RKGA apresentou um melhor desempenho (Gonçalves et al., 2012).

No BRKGA, cada indivíduo da população é representado por um vetor com números reais aleatórios (chaves aleatórias) no intervalo [0; 1), que deverá ser decodificado por um algoritmo para transformá-lo em uma solução factível do problema.

A população de uma geração é dividida em dois grupos: um primeiro grupo menor com as melhores soluções (elite) e um segundo grupo maior com as demais soluções (não elite). A geração seguinte é formada por três grupos: todas as soluções de elite são copiadas, algumas novas soluções são geradas aleatoriamente (aumentando a diversidade da população) e o restante da população é formado por cruzamentos dos indivíduos da geração anterior.

O cruzamento é feito por um *crossover* parametrizado: dois cromossomos são selecionados, sendo um do grupo de elite e outro do grupo não elite. Para a escolha de cada gene do filho, é feito um sorteio onde a probabilidade de a solução de elite transmitir o seu gene é maior do que a da outra solução. Não há mutações após o cruzamento.

Duas versões do BRKGA foram desenvolvidas para o PETC. Na primeira, as chaves aleatórias foram utilizadas para selecionar um técnico para cada tarefa e construir o roteiro de cada técnico. Desse modo, os cromossomos terão uma chave aleatória para cada tarefa, que serão designadas para um técnico da seguinte maneira: se há dois técnicos capazes de realizar uma tarefa e a chave aleatória for um número no intervalo [0;1/2), então ela será associada ao primeiro técnico; caso seja um número entre [1/2; 1), será designada ao segundo. Um procedimento análogo é realizado quando uma tarefa tem 3 ou mais técnicos aptos. Uma vez distribuídas todas as tarefas, para construir a rota de cada técnico foi extraído um *índice* de cada chave aleatória, que é a porcentagem de distância entre o valor da chave aleatória e o início do subintervalo: por exemplo, se a chave aleatória tem valor 0,25, para o exemplo anterior, então ela está exatamente na metade do subintervalo [0;1/2) e tem índice de 50%. Desse modo, as tarefas de cada técnico são ordenadas em ordem crescente do índice e, da primeira à última, serão introduzidas no roteiro do técnico; caso não haja solução factível (por conflitos de janelas de tempo, etc.) para introduzir uma tarefa na rota, ela não será realizada. Essa versão foi denominada BRKGA-A.

Na segunda versão, foi criado um cromossomo também com uma chave aleatória para cada tarefa, que serão ordenadas em ordem crescente. As tarefas serão introduzidas nas rotas dos técnicos a partir da tarefa com menor chave aleatória até a tarefa com maior valor da chave aleatória. Para cada tarefa, todos os técnicos aptos são avaliados e o técnico que puder realizar aquela tarefa com menor tempo de deslocamento total no seu dia será escolhido. Essa versão foi chamada de BRKGA-STT.

4. Experimentos computacionais e resultados

Todos os códigos foram programados em linguagem C e os experimentos realizados em um computador Intel com 2.93 GHz e processador 16GB de memória RAM. Para o modelo PLIM utilizou-se o *software* ILOG CPLEX, versão 12.2, com os parâmetros *default*.

Foram construídas 260 instâncias: 20 para cada um dos casos apresentados na Tabela 1 (a segunda e terceira colunas da tabela apresentam, respectivamente, o número de tarefas e técnicos em cada caso). Os parâmetros dessas instâncias foram gerados por uma distribuição uniforme discreta dos seguintes valores: prioridade das tarefas de 1 (baixa) a 10 (alta); tempo de execução das tarefas de 30 a 120 min.; início da janela das tarefas de 8 a 16h; tamanho da janela das tarefas de 2 a 8h; início da janela dos técnicos de 6:30 a 11h; fim da janela dos técnicos somando 9h ao início da janela de tempo; habilidade dos técnicos para executar cada tarefa será 0 ou 1; todas as tarefas estão em uma região onde o tempo de deslocamento entre quaisquer dois locais (diâmetro) é menor ou igual a 1,5 hora.

A população inicial do BRKGA foi gerada aleatoriamente, com exceção de alguns cromossomos incluídos, que representam as soluções de heurísticas construtivas apresentadas na ONPCE XV (Damm e Ronconi, 2013).

Para os casos 1 a 4 (dimensões pequenas) foram encontradas as soluções ótimas pelo software Cplex e, para os casos maiores, não foram obtidas soluções ótimas em até uma hora de processamento. O BRKGA-A encontrou 95% das soluções ótimas dos casos 1 a 4 e o BRKGA-STT, 98%.

A tabela 1 apresenta a distância média percentual com relação aos limitantes superiores, calculados pela expressão:

$$GAP_{UB} = 100 \cdot \frac{BR - UB}{UB}$$

onde BR é o valor da função objetivo obtida pelo BRKGA e UB o valor da função objetivo do limitante superior.

Caso	Tarefas	Técnicos	BRKGA-A	BRKGA-STT
1	12	2	**-2,4**	**-2,4**
2	16	2	**-3,4**	**-3,4**
3	26	2	**-8,2**	**-8,2**
4	40	3	-9,0	**-8,9**
5	45	7	-5,0	**-4,4**
6	64	5	-11,6	**-11,2**
7	80	13	-5,7	**-4,8**
8	100	10	-12,0	**-10,9**
9	120	20	-5,1	**-4,0**
10	150	25	-5,0	**-3,4**
11	200	33	-5,6	**-3,7**
12	500	83	-4,4	**-3,1**
13	999	166	-3,9	**-2,7**
		Média dos casos 1-4 (%)	**-5,7**	**-5,7**
		Média dos casos 5-13 (%)	-6,5	**-5,3**

Tabela 1: Diferença percentual média entre os resultados obtidos e os limitantes superiores. Em negrito, os melhores valores obtidos.

Os resultados obtidos pelo BRKGA-STT em média superaram os resultados do BRKGA-A nos casos 4 a 13. Para dimensões médias e grandes (casos 5 a 13), os resultados do BRKGA-STT ficaram em média a 5,3% dos limitantes superiores calculados, o que parece um bom desempenho, uma vez que os valores ótimos das instâncias pequenas estão a uma distância média de 5,7% dos limitantes superiores. Nos casos em que há uma maior proporção de tarefas por técnicos (casos 3, 4, 6 e 8), os resultados obtidos ficaram em média mais distantes dos limitantes superiores.

5. Conclusões e perspectivas futuras

A principal contribuição deste trabalho foi a forma de aplicar o BRKGA ao PETC. A diferença percentual do desempenho entre as duas versões do BRKGA foi em média de 1% e, portanto, nota-se que as formas de codificar e decodificar as soluções tem um impacto significativo nos resultados. Outras versões do BRKGA foram desenvolvidas (por exemplo, adotando estratégias de busca local ou enumerações completas para determinar o roteiro de um técnico), obtendo desempenhos ligeiramente inferiores aos apresentados neste trabalho.

Como perspectivas futuras, pretendem-se estudar o problema com diferentes distribuições de pontos numa cidade (distribuição uniforme, agrupamento de clientes em clusters, etc.) e também desenvolver modelo multiobjetivo para o PETC.

Referências

Bean, J. C. (1994). Genetic algorithms and random keys for sequencing and optimization. *ORSA journal on computing*, 6 (2), 154-160.

Cordeau, J. F.; Laporte, G.; Pasin, F.; Ropke, S. (2010). Scheduling technicians and tasks in a telecommunications company. *Journal of Scheduling*, 13, 393-409.

Damm, R. B.; Ronconi, D. P. (2013). O problema de escalonamento de técnicos de campo. *XV Oficina Nacional de Problemas de Corte, Empacotamento, Dimensionamento de Lotes e Programação da Produção*, Caderno de resumos, 18–19.

Gonçalves, J. F.; Resende, M. G. (2011). Biased random-key genetic algorithms for combinatorial optimization. *Journal of Heuristics*, 17(5), 487-525.

Gonçalves, J. F.; Resende, M. G.; Toso, R. (2012). An experimental comparison of biased and unbiased random-key genetic algorithms. *Technical report, AT&T Labs Research*.

Hashimoto, H.; Boussier, S.; Vasquez, M.; Wilbaut, C. (2011). A GRASP-based approach for technicians and interventions scheduling for telecommunications. *Annals of Operations Research*, 183, 143-161.

Kovacs, A. A.; Parragh, S. N.; Doerner, K. F.; Hartl, R. F. (2011). Adaptive large neighborhood search for service technician routing and scheduling problems. *Journal of Scheduling*, 15, 579-600.

Pillac, V.; Guéret, C.; Medaglia, A. L. (2012). A parallel matheuristic for the technician routing and scheduling problem. *Optimization Letters*, 1-11.

Tsang, E.; Voudouris, C. (1997). Fast local search and guided local search and their application to British Telecom's workforce scheduling problem. *Operations Research Letters*, 20, 119-127.

Xu, J.; Chiu, S. Y. (2001). Effective Heuristic Procedures for a Field Technician Scheduling Problem. *Journal of Heuristics*, 7, 495-509.