XXIV ONPCE: Oficina Nacional de Problemas de Corte e Empacotamento, Planejamento e Programação da Produção e Correlatos

A biased random-key genetic algorithm for the service technician routing and scheduling problem with teams

José Angel Riveaux Merino[1]
Department of Production Engineering, Polytechnic School, University of São Paulo
Débora P. Ronconi[2]
Department of Production Engineering, Polytechnic School, University of São Paulo
Ricardo Damm[3]
Department of Production Engineering, Polytechnic School, University of São Paulo

The service technician routing and scheduling problem with team building (STRSPt) consists in finding the minimum cost vehicle routes for attending a set of tasks [1]. Each technician is specialized in different skills with different levels and each task demand one or more technician with the appropriated skills. To fulfill the skill requirements of the tasks, technicians can be grouped into teams. For the purpose of this work, it is assumed that teams are formed at the beginning of day and will continues until the end of day. Also, the service must start within the time windows associated to the task. A special case of the problem was also studied, that is the incorporation of lunch as a mandatory task for all teams. The lunch also have a time windows and an execution time.

This work continues the previous research of Damm et al [2] published in 2016, with competitive results of the BRKGA for the no-team version of the problem. The BRKGA is a genetic algorithm proposed by Gonçalves and Resender [3], based on the RKGA algorithm that represent each solution as a vector of random keys. It differs from RKGA in the way parents are selected for crossover, taking one mate from the elite population and the other for the non-elite. Toso and Resende provide a BRKGA public framework in C++ [4]. Its instantiation to some specific optimization problem requires exclusively the development of a class implementing the decoder for this problem. This is the only problem-dependent part of the tool.

For the STRSPt was developed three different decoders. The first one is the base decoder where a chromosome is made up by a $n + m$ alleles where the first n elements are associated with the tasks and the other m elements are associated with the technicians. Then technicians and task, are selected in the order of their keys to be added to a current solution. The other two decoder add a greedy strategy to selection of tasks: one is based on the difficulty of task, which is calculated from the number of technician reacquired to do the task (Decoder greedy 1) and the other in based in the change of objective function value after the insertion of task in the solution (Decoder greedy 2). For the team creation, we use a strategy based in the Kovacs idea to select the kernel tasks and create teams with the same algorithm proposed in their work [1].

Experiments were run in MacBook Pro (13-inch, M1, 2020), procesor M1. The Table 1 shows results of the BRKGA with the three decoders running an equals number of generations, three times for each instances. The instances, are the same used by Kovacs, et al [1] based on

[1] jangel.riveaux@gmail.com
[2] dronconi@usp.br
[3] rbdamm@gmail.com

Tabela 1: Decoder solution comparation

	Decoder base			Decoder greedy 1			Decoder greedy 2		
	minimum	average	maximum	minimum	average	maximum	minimum	average	maximum
C101_5x4	1,677.06	1,941.40	2,134.72	962.23	1,004.33	1,061.77	**879.88**	**892.75**	**905.39**
C101_6x6	1,792.64	1,857.21	1,921.43	992.32	1,064.31	1,108.90	**956.04**	**981.92**	**1,026.15**
C103_5x4	1,437.25	1,489.66	1,549.15	1,101.76	1,107.57	**1,111.77**	**1,097.39**	1,111.63	1,119.68
C103_6x6	1,444.13	1,568.42	1,649.36	1,188.37	1,218.14	**1,271.18**	**1,103.50**	**1,113.10**	1,127.75
C103_7x4	1,795.57	1,890.53	1,978.90	1,193.51	1,289.09	1,397.43	**1,191.70**	**1,230.33**	**1,259.89**
R101_5x4	**1,742.14**	**1,792.13**	**1,827.07**	1,913.47	1,950.05	1,981.56	1,752.67	1,857.14	1,940.54
R101_6x6	**1,784.14**	**1,805.78**	**1,828.58**	1,824.44	1,909.24	1,989.29	1,821.71	1,891.61	1,950.21
R101_7x4	**1,870.72**	**1,938.08**	**1,988.28**	2,303.51	2,339.63	2,361.56	1,881.90	2,103.43	2,347.86
C201_5x4	998.02	1,048.69	1,110.50	627.99	627.99	627.99	**611.57**	**611.57**	**611.57**
C201_6x6	1,054.31	1,163.16	1,221.66	**623.57**	**627.91**	636.60	624.39	628.24	**635.94**
C201_7x4	993.90	1,157.20	1,257.26	741.57	741.57	741.57	**689.55**	**699.31**	**704.19**
C203_5x4	1,023.16	1,219.77	1,321.69	732.92	741.14	745.26	**717.92**	**723.14**	**725.75**
C203_6x6	980.63	1,325.04	1,581.67	765.79	799.47	847.72	**761.17**	**794.64**	**811.38**
C203_7x4	1,141.47	1,160.29	1,180.55	855.10	872.30	**883.83**	**800.88**	**856.77**	885.18

the benchmark instances of Solomon (1987) for the vehicle routing problem with time windows (VRPTW) and the test instances for the ROADEF 2007 challenge.

It can be observed that the second greedy criteria gets the best average solution except for instances R101_5x4, R101_6x6, and R101_7x4.

This is a research in progress and the results are preliminary. We are incorporating a path-relinking strategy to the BRKGA to obtain a solution improvement and setting the BRKGA parameters based on the idea of Ribeiro, et al [5].

Referências

[1] A. A. Kovacs , S. N. Parragh, K. F. Doerner, and R. F. Hartl. Adaptive large neighborhood search for service technician routing and scheduling problems.*Journal of Scheduling*, 15 (5):579–600, 2012.

[2] Damm R. B., Resende, M. G. C., Ronconi, D. P. (2016). A biased random key genetic algorithm for the field technician scheduling problem. *Computers & Operations Research*, 75, 49–63. https://doi.org/10.1016/j.cor.2016.05.003

[3] Gonçalves J. F., & Resende, M. G. (2011). Biased random-key genetic algorithms for combinatorial optimization. *Journal of Heuristics*, 17 , 487–525.

[4] Toso R. F., & Resende, M. G. (2015). A C++ application programming interface for biased random-key genetic algorithms. Optimization Methods and Software, 30 , 81–93.

[5] Ribeiro, Celso C., Jose A. Riveaux, and Julliany S. Brandao. "Biased random-key genetic algorithms using path-relinking as a progressive crossover strategy."2021 *5th International Conference on Intelligent Systems, Metaheuristics & Swarm Intelligence.* 2021.