

Um Método Heurístico Para o Problema de Alocação de Pessoal

Marco Antonio M. Carvalho, André Gustavo dos Santos

Instituto Tecnológico de Aeronáutica – Praça Marechal Eduardo Gomes, 50, São José dos Campos, SP, Universidade Federal de Minas Gerais - Avenida Antonio Carlos, 6627, Belo Horizonte, MG

Resumo — Este trabalho apresenta o desenvolvimento de um método heurístico para o problema de alocação de pessoal, especificamente no subproblema de seleção de colunas para desempenhar todas as atividades existentes, modelado como uma versão multiobjetivo do *set covering problem* (SCP). Apresentamos justificativas para se trabalhar com a versão multiobjetivo, dado o contexto do problema, encontrando um conjunto de soluções eficientes, um apoio para decisão de um agente externo. Mostramos as idéias incorporadas ao algoritmo genético clássico, e os resultados comprovam as melhorias obtidas em termos da qualidade da solução e quantidade de pontos na solução Pareto-Ótima.

Palavras-chaves — inteligência artificial, metaheurísticas, alocação de pessoal, seleção de colunas, algoritmo genético.

I. INTRODUÇÃO

O problema de alocação de pessoal é classificado como um problema NP-difícil [13] e consiste em distribuir atividades ao pessoal disponível, ou mesmo designar atividades pré-definidas (treinamentos, exames médicos e folgas, entre outros), gerando jornadas (normalmente mensais) individuais para cada pessoa.

A distribuição deve ser feita de forma a otimizar a jornada de cada pessoa, minimizando assim os gastos com pessoal. Devem ser respeitadas ainda as legislações federais, sindicais e normas das entidades, que regulam a relação entre entidades e pessoal, de tal forma que todas as atividades sejam executadas pelas pessoas.

Tipicamente, o problema é tratado utilizando-se a geração de colunas, técnica aplicada a problemas lineares de grandes dimensões, no caso de não se dispor de todas as colunas *a priori* e seleção posterior das colunas geradas. Neste trabalho, colunas são jornadas de trabalho, contendo seqüências de atividades a serem executadas por uma pessoa.

Na geração de colunas, é gerado um conjunto de jornadas possíveis para cada pessoa, para posterior seleção do melhor subconjunto de jornadas que atenda a todas as atividades a serem executadas, de acordo com algum objetivo.

A seleção de colunas é um problema de recobrimento de conjuntos (*set covering problem* - SCP), um problema NP-Completo, o qual aparece como o modelo mais comumente empregado para formulação de problemas de otimização de grande porte.

A seleção de colunas é considerada o problema mestre, e a geração de colunas é considerada um subproblema. Neste trabalho consideramos que as colunas foram geradas anteriormente e nos concentramos na fase de seleção de colunas.

O problema de alocação de pessoal é um problema de programação inteira.

II. REVISÃO DA LITERATURA

Grande parte dos trabalhos que tratam o problema de alocação de pessoal, aborda o tema no contexto de serviços ininterruptos, sendo que a maior concentração é em serviços de transporte em massa e tarefas devem ser designadas a tripulantes, e dentre eles citamos abordagens em malhas aéreas [7], em malhas ferroviárias [4] e rodoviárias [8]. Em todos eles o problema de recobrimento de conjuntos (*set covering problem* - SCP) é usado como um subproblema. Como dito anteriormente, uma das etapas do processo de alocação de pessoal consiste em se resolver o SCP.

O SCP é tradicionalmente tratado na literatura em sua forma mono-objetivo. Os melhores algoritmos exatos para a versão mono-objetivo do SCP utilizam *branch-and-cut* combinado com outras técnicas [3]. O problema também já foi atacado por diversas técnicas heurísticas, e dentre elas citamos relaxação lagrangeana [1], *simulated annealing* [11] e algoritmos genéticos [2]. Entretanto, é encontrado em situações reais em sua versão multiobjetivo, por exemplo, quando usado como subproblema na solução do problema de alocação de pessoal.

Uma tendência recente é considerar mais de um objetivo em problemas de otimização, levando em conta não apenas otimizar um objetivo, mas também balancear outros fatores que influem na qualidade de uma solução, caracterizando assim o critério multiobjetivo. Porém, podem ser objetivos conflitantes, não havendo uma solução que atenda de forma ótima simultaneamente os dois critérios, e sim um conjunto de soluções ótimas, chamado conjunto Pareto-Ótimo ou conjunto de soluções eficientes [5]. Um algoritmo para esse problema deve então gerar todas as soluções eficientes já que a princípio, não há informação sobre como comparar as soluções sem privilegiar um objetivo específico. Posteriormente, as soluções são analisadas por algum agente de tomada de decisão, que escolherá a solução que melhor lhe satisfaz.

Diante de tantos objetivos importantes, não podemos nos limitar à tradicional forma de resolver o problema apenas diminuindo os custos diretos com a alocação das pessoal, mas devemos tratar o problema como multiobjetivo, como de fato ele ocorre na prática. Aqui tratamos o problema na forma bi-objetivo, podendo, por exemplo, ter como objetivos o custo da jornada de cada pessoa e a preferência da pessoa em

relação a ela. Porém, o algoritmo é facilmente extensível para mais de dois objetivos.

III. SELEÇÃO DE COLUNAS

A entrada do problema de recobrimento de conjuntos bi-objetivo é uma matriz binária A de dimensões $m \times n$, c e d , dois vetores n -dimensionais, onde c_j representa o custo da coluna j considerando-se o primeiro objetivo, e d_j representa o custo da coluna j considerando-se o segundo objetivo, assumindo $c_j > 0$ e $d_j > 0$. As colunas são jornadas de trabalho e as linhas são atividades a serem desempenhadas.

Dizemos que uma coluna j cobre a linha i se $a_{ij} = 1$ (onde a_{ij} representa o elemento da matriz A na linha i e na coluna j). Neste caso, a jornada j desempenha a atividade i , em outras palavras, a atividade i está incluída na jornada j . Caso contrário $a_{ij} = 0$. O problema consiste em selecionar um subconjunto de colunas de forma que todas as linhas sejam cobertas com um custo mínimo.

Neste trabalho, assume-se que as colunas geradas são viáveis quanto ao seu comprimento. O objetivo é então distribuir as tarefas em jornadas minimizando custos com mão de obra.

A. O Algoritmo Genético

Algoritmos genéticos são uma técnica de programação evolutiva utilizada em problemas de otimização complexos. A idéia por trás do algoritmo genético é emular o que existe na natureza: sobrevivência dos mais aptos. Operadores de *mutação* (*mutation*), *cruzamento* (*crossover*) e *seleção* (*selection*) são emulados. Estes operadores são aplicados a um conjunto de soluções em potencial (*população*) chamadas *cromossomos* ou *indivíduos*.

A codificação utilizada é a de um vetor binário, indicando a pertinência de cada coluna a uma solução, e a população inicial é gerada de forma aleatória, sem a utilização de nenhum conhecimento específico do problema. Para cada cromossomo, cada um dos bits recebe o valor 0 ou 1, com a mesma probabilidade.

Em versões multiobjetivo é comum se encontrar soluções melhores em certos objetivos e piores em outros. Formam-se então barreiras de soluções não dominantes entre si, mas que dominam outras barreiras de soluções. Esta classificação é utilizada para estabelecer um outro ranking das soluções [14] em barreiras, utilizado em conjunto com a função de avaliação para estabelecer a adaptação de cada cromossomo.

Para o processo de seleção é utilizado o método da roleta, o qual combina probabilidade e aleatoriedade, para a escolha de quais indivíduos serão mantidos para a próxima execução do algoritmo.

A implementação de *crossover* (ou cruzamento) utilizada combina o material genético de dois cromossomos selecionados anteriormente através do método de torneio binário, no qual são formadas duas filas com dois cromossomos selecionados aleatoriamente cada. O melhor cromossomo de cada fila é selecionado para participar do *crossover* de ponto, no qual um ponto aleatório dentro do cromossomo é selecionado e a partir dele o material genético é trocado para geração de um par de filhos.

Após a geração do par de filhos, estes são adicionados à população corrente, e poderão posteriormente substituir os

cromossomos da população dominados por eles, ou também serem eliminados caso sejam dominados..

O operador de mutação é o tradicional, e inverte o valor do gene submetido, alterando assim seu valor nas funções objetivo.

Após a aplicação dos operadores genéticos, uma avaliação da nova população é realizada, e a fim de conter o aumento descontrolado da população, é realizado um corte na mesma, mantendo para a próxima iteração do algoritmo apenas as três primeiras barreiras ou 40% dos indivíduos da população corrente.

B. Funções de Reparo

Após a aplicação dos operadores genéticos são utilizadas funções de reparo para tratar soluções que se tornaram inviáveis ou redundantes. Para isso, as funções de reparo incluem colunas a fim de cobrir linhas não cobertas, e também retiram colunas redundantes (que só cobrem linhas já cobertas). Para isso, ambas as funções de reparo verificam densidade de cada coluna, a razão entre os custos das colunas de acordo com os dois objetivos e o número de linhas cobertas, fazendo um ranking das colunas com base nas seguintes expressões:

$$c_j / \sum_{i=1}^m a_{ij} \quad (1)$$

$$d_j / \sum_{i=1}^m a_{ij} \quad (2)$$

$$(c_j + d_j) / \sum_{i=1}^m a_{ij} \quad (3)$$

Onde c_j representa o custo da coluna j considerando-se o primeiro objetivo, d_j representa o custo da coluna j considerando-se o segundo objetivo e o somatório representa a quantidade de linhas (enumeradas de 1 a m) do problema cobertas pela coluna j .

A Equação (1) dá preferência a colunas com custos baixos no primeiro objetivo que cobrem maior quantidade de linhas, (2) faz o mesmo para o segundo objetivo, e (3) faz uma combinação linear dos objetivos. Estas expressões são utilizadas em um esquema *round robin*: cada uma é usada durante um certo número de iterações. Este esquema permite explorar melhor o espaço de busca, corrigindo deficiências e evitando ótimos locais, no caso de aglomeração da população em apenas uma região da fronteira Pareto-ótima.

V. TESTES E RESULTADOS PARA SELEÇÃO DE COLUNAS

São apresentados os resultados obtidos pela fase de seleção de colunas, bem como comparações entre versões dos algoritmos desenvolvidos neste trabalho e comparações das versões finais com técnicas da literatura recente.

Todos os testes computacionais foram realizados em computadores com processadores AMD Duron 1.3 Gigahertz, 128 megabytes de memória RAM, sistema operacional SuSE Linux e compilador GCC (*Gnu Compiler Collection*).

Foram realizados testes com instâncias encontradas na biblioteca de instâncias numéricas da MCDM [10], na seção *MultiObjective Set Covering Problem* (MOSCP).

A. Resultados

Os resultados da fase de geração de colunas são apresentados como conjuntos de pontos, onde cada ponto representa os valores de cada solução. Como se trata da versão bi-objetivo do problema de recobrimento de conjuntos, os resultados são apresentados em um gráfico de dispersão, onde o valor em cada eixo representa o valor de cada objetivo da solução.

Para a verificação de dominância, para cada ponto, traça-se uma reta imaginária paralela a cada eixo; os pontos que se encontrarem dentro da área imaginária do primeiro quadrante são considerados dominados pelo ponto localizado na origem das retas, pois possuem valores piores em ambos os objetivos. O problema aqui abordado é um problema de minimização, portanto quanto mais os pontos se aproximarem dos eixos do gráfico, melhor serão em comparação aos outros mais distantes.

Foram realizados dois tipos de comparações: o primeiro compara versões do algoritmo desenvolvido neste trabalho, com objetivo de avaliar as principais melhorias implementadas, neste caso, o torneio binário e o esquema *round robin*; o segundo envolve sua comparação com os resultados apresentados em [12], cujo trabalho descreve a implementação e comparação de dez metaheurísticas aplicadas ao problema de recobrimento de conjuntos bi-objetivo.

As métricas de comparação de qualidade dos conjuntos de pontos gerados utilizadas são: comparação direta ponto a ponto, onde se verifica a quantidade de pontos dominados e/ou acrescentados; e a medida de distância média do conjunto de pontos, proposta em [6], onde o conjunto de referência é o conjunto gerado neste trabalho. A medida de distância assume o valor zero se em todos os objetivos o conjunto gerado [12] alcança os valores do conjunto de referência, caso contrário, a medida assume o valor do máximo desvio de valores.

A Tabela I mostra o resultado da execução do algoritmo com e sem a utilização de torneio binário na fase de *crossover*, para algumas entradas.

TABELA I MELHORIAS ALCANÇADAS UTILIZANDO TORNEIO BINÁRIO

Instância	Número de pontos		Pontos dominados		Distância média entre as versões
	Sem torneio binário	Com torneio binário	Sem torneio binário	Com torneio binário	
2scp11A	38	36	4	0	0.00140
2scp82B	8	13	4	0	0.04993
2scp101A	11	12	9	0	0.16783

A Fig. 1 ilustra os resultados da comparação a respeito da utilização do torneio binário.

Em todos os casos a utilização de torneio binário gerou um conjunto melhor de pontos, sendo melhor quanto maior a entrada. Nota-se que para a entrada 2scp11A foram gerados menos pontos, porém de melhor qualidade, dominando 4 dos pontos gerados sem o torneio.

A Tabela II mostra o resultado da execução do algoritmo com e sem a utilização do esquema *round robin* nas funções de reparo. Embora a utilização do *round robin* tenha deixado de gerar alguns pontos, o conjunto de pontos gerados é melhor, sempre dominando soluções geradas sem *round robin*.

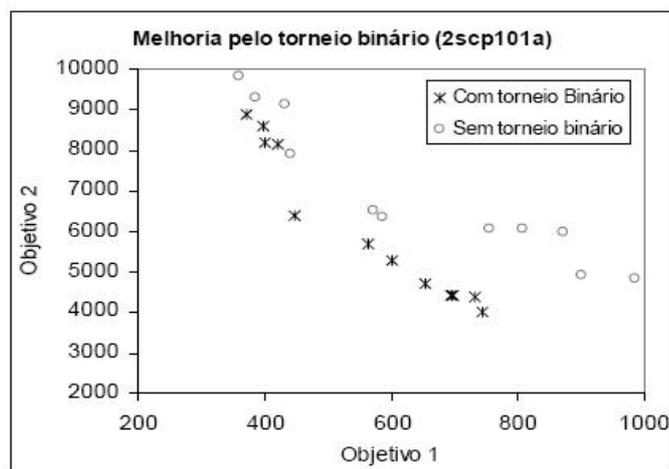


Fig. 1. Melhoria alcançada utilizando o torneio binário em relação ao método aleatório.

TABELA II MELHORIAS ALCANÇADAS UTILIZANDO ROUND ROBIN

Instância	Número de pontos		Pontos dominados		Distância média entre as versões
	Sem round robin	Com round robin	Sem round robin	Com round robin	
2scp11A	31	38	8	0	0.00322
2scp82B	16	13	7	1	0.01077
2scp101A	11	12	7	3	0.07491

A Fig. 2 ilustra o comportamento das versões do algoritmo na comparação.

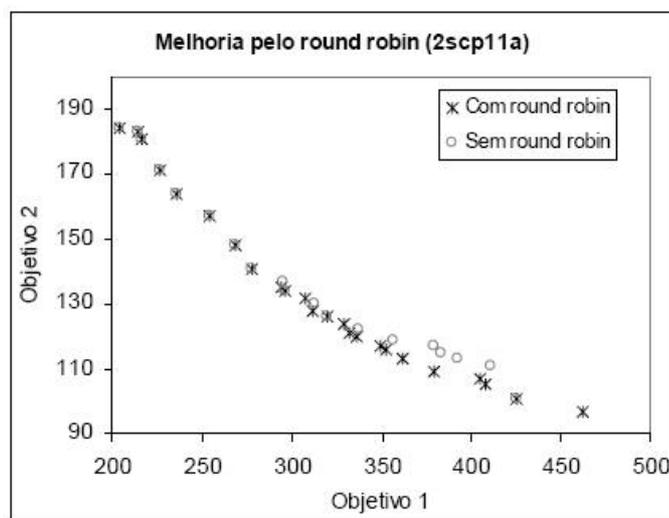


Fig. 2. Melhoria alcançada utilizando o esquema *round robin* em relação ao método tradicional de exploração.

Compararam-se os resultados para três instâncias relatadas em [12] e disponibilizadas em sua página web. Em seu trabalho, Jaskiewicz compara versões de algoritmos genéticos, *Simulated Annealing* métodos de busca local e um novo método proposto: *Pareto Memetic Algorithm*. Ele conclui que os melhores desempenhos são *Multiple-Objective Genetic Local Search* (MOGLS) e *Pareto Memetic Algorithm* (PMA), sendo as duas versões de algoritmos genéticos as que obtiveram pior desempenho.

Reuniram-se os resultados dos dois métodos de melhor desempenho, descartando os pontos dominados gerados para

cada série de instâncias, a Tabela III apresenta a comparação.

VI. CONCLUSÕES

TABELA III COMPARAÇÃO COM OS RESULTADOS DOS MÉTODOS MOGLS+PMA

Instância Representativa	Comparação Direta	Distância Média
2scp82B	Todos os pontos dominados ou iguais	0,01451
2scp101A	25 pontos não foram iguais ou dominados	-0,00097
2scp102A	Todos os pontos dominados ou iguais	0,98544

Dentre as séries testadas, a série 2scp101, representada pela instância 2scp101A teve um desempenho ligeiramente pior. Mas para as séries de densidade maior e objetivos conflitantes, o algoritmo genético implementado encontra resultados melhores. Pode ser observado que para a instância 2scp102A a distância média entre as barreiras foi quase 1,0 pela métrica utilizada. A Fig. 3 apresenta a primeira barreira de cada método, e a Fig. 4 apresenta uma ampliação da parte inicial das barreiras na comparação, destacando a maior diferença entre os as soluções geradas. Os valores nos eixos representam os valores de cada objetivo, e são meramente ilustrativos.

Apresentaram-se alterações no algoritmo genético clássico aplicando-o ao problema de recobrimento multiobjetivo, a fim de usá-lo em problemas de alocação de pessoal, embora possa ser utilizado em vários outros problemas onde a seleção de colunas surge como um subproblema.

O desempenho do algoritmo também foi testado com outras heurísticas e resultados recentes da literatura, mostrando um desempenho superior para algumas séries de instâncias. A incorporação de torneio binário e um esquema de reparo dos cromossomos utilizando uma técnica *round robin* de preferência de objetivos trouxe significantes melhorias ao algoritmo, corrigindo deficiências na exploração do espaço de soluções e tornando o método apresentado mais competitivo.

Marco Antonio M. Carvalho, mame@ita.br, André Gustavo dos Santos, andre@dcc.ufmg.br, Tel +55-12-81131711.

Este trabalho foi parcialmente financiado pelo CNPq, através do programa de bolsas de Iniciação Científica e programa de bolsas de Mestrado no país.

REFERÊNCIAS

- [1] Beasley, J. E., 1990, "A Lagrangean Heuristic for Set-Covering Problems", *Naval Research Logistics*, 37:151-164.
- [2] Beasley, J. E., Chu, P. C., 1996, "A genetic algorithm for the set covering problem", *European Journal of Operational Research*, vol.94, 1996, pp392-404.
- [3] Beasley, J. E. and Jornsten, K., 1992 "Enhancing an algorithm for set covering problems", *European Journal of Operational Research*, 58:293-300.
- [4] Caprara, A., Fischetti, M., Toth, P., Vigo, D., Guida, P. L., 1998, "Algorithms for railway crew management", *Mathematical Programming* 79, 125-141.
- [5] Chankong, V. and Haimes, Y. Y., 1983, "Multiobjective Decision Making: Theory and Methodology", North-Holland (Elsevier), New York.
- [6] Czyzak, P. and Jaszkiwicz, A., 1998, "Pareto Simulated Annealing - A Metaheuristic Technique for Multiple Objective Combinatorial Optimization", *Journal of Multicriteria Decision Analysis*, 7 34-47.
- [7] Ehrgott, M. and Ryan, D. M., 2003, "Constructing robust crew schedules with bicriteria optimization", Technical report, University of Auckland, To appear in *Journal of Multi-Criteria Decision Analysis*.
- [8] Fores, S., Proll, L. and Wren, A., 1996, "A column generation approach to bus driver scheduling", Bell, M H G (editor), *Transportation Networks: Recent Methodological Advances*, pp.195-208, Pergamon.
- [9] Hu, J., Johnson, E. L., 1999, "Computational results with a primal-dual subproblem simplex method. *Operations*", *Research Letters*, 25, 149-157.
- [10] International Society on Multiple Criteria Decision Making, 2005, <http://www.univvalenciennes.fr/ROAD/MCDM/ListMOSCP.html>.
- [11] Jacobs, L. W. and Brusco, M. J., 1993, "A simulated annealing-based heuristic for the set-covering problem", working paper, Operations Management and Information Systems Department, Northern Illinois University, Dekalb, IL 60115, USA.
- [12] Jaszkiwicz, A., 2004, "A Comparative Study of Multiple-Objective Metaheuristics on the Bi-Objective Set Covering Problem and the Pareto Memetic Algorithm", *Annals of Operations Research* 131, 135-138. Kluwer Academic Publishers.
- [13] Souza, M. J. F., Cardoso, L. X. T., and Silva, G. P., 2003, "Programação de Pessoal de Ônibus: Uma Abordagem Heurística", XXXV Congresso da Sociedade Brasileira de Pesquisa Operacional SBPO.
- [14] Srinivas, N. and Deb, K., 1994, "Multiple Objective Optimization using Nondominated Sorting in Genetic Algorithms", *Evolutionary Computation* 2(2), 221-248.

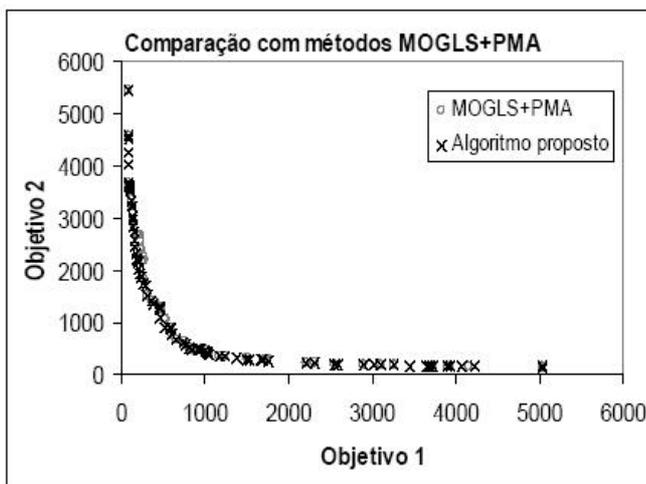


Fig. 3. Comparação entre as heurísticas para a entrada 2scp102A, mostrando todos os pontos gerados nas barreiras iniciais.

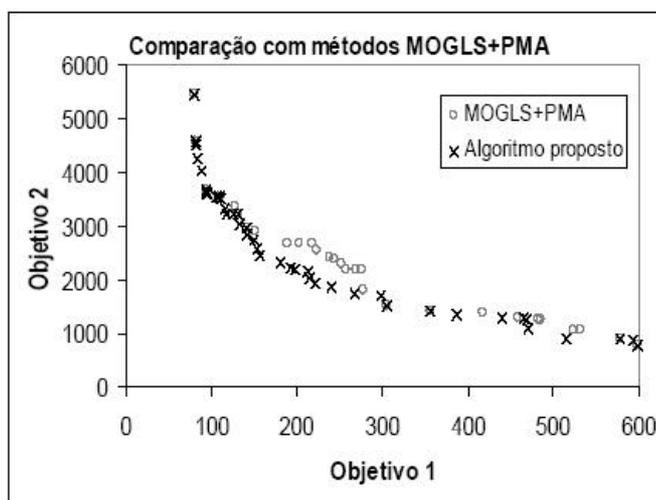


Fig. 4. Comparação entre as heurísticas para a entrada 2scp102A, mostrando apenas a parte inicial das barreiras.