

Evolução Diferencial Discreta para Programação de Turnos de Trabalho

Raimundo Ronilson Leal do Rosário¹, Angela Olandoski Barboza²,
Rosely Antunes de Souza³, Antônio Sérgio Coelho⁴

^{1,2,3} Universidade Tecnológica Federal do Paraná, Curitiba-PR, Brasil

⁴ Universidade Federal de Santa Catarina, Florianópolis-SC, Brasil

¹ ronilson@utfpr.edu.br

² aobarboza@yahoo.com

³ roselypr@gmail.com

⁴ coelho@deps.ufsc.br

Abstract. In this paper a model was developed to solve the shift scheduling problem with application in a Call Center company. This is a complex optimization problem due to the large number of restrictions and requirements. The shift scheduling problem for 5 weeks was divided into two problems: determining shifts for one week; assigning these shifts to employees for a period five weeks. For solving both problems was used Discrete Differential Evolution (EDD). The main objectives are: to optimize the number of telephone operators and the variation between times of arrival of these. The results of the EDD for the first problem were compared with those of a mathematical model of ILP and found that the results are equal to optimal or close to optimal. For the second problem we found the optimal result for the function considered the most important of the function bi-objective.

Keywords: Discrete Differential Evolution, shift scheduling, assigning, optimization.

1 Introdução

Devido à variação da demanda de chamadas nos Call Centers surgem vários problemas, dentre eles determinar a quantidade necessária de atendentes em cada turno para atender as demandas a cada momento e a designação dos turnos aos atendentes. Isto motivou o desenvolvimento deste trabalho, no qual, foi desenvolvido um procedimento para resolver o problema de programação de turnos de trabalho com aplicação em um Call Centers.

Segundo [1] geral são consideradas três etapas no desempenho da programação de funcionários para uma organização de serviço. A primeira etapa é a previsão da demanda para cada hora de trabalho de cada dia durante o próximo período de planejamento. A segunda etapa é traduzir as previsões de demanda em necessidade de funcionários para cada hora de trabalho do período de planejamento a fim de proporcionar um nível de serviço desejado ao cliente. A terceira etapa é atender as necessidades dos funcionários para cada jornada de trabalho do período de

planejamento, determinando os dias de trabalho, e um turno em cada dia de trabalho, para cada funcionário.

Neste artigo, o problema se refere à programação dos turnos de trabalho para o período de cinco semanas, ou seja, a determinação dos turnos de forma a atender a demanda de atendentes para cada hora do período de planejamento e a designação deste aos empregados para um período de planejamento de cinco semanas. Os tipos de turnos de trabalho devem satisfazer às restrições trabalhistas e também operacionais da empresa.

Programação de pessoal, ou escalas, é o processo de construção de horários de trabalho para seus funcionários, para que uma organização possa satisfazer a demanda por seus produtos ou serviços [2].

Os principais objetivos deste trabalho estão relacionados à redução de custos com salários, assim procura-se otimizar a quantidades de atendentes e a satisfação dos atendentes em relação a pouca variação do momento de entrada no trabalho de um dia para o outro.

Na literatura, existem diversos trabalhos que tratam de programação de horários e designação de turnos, dada a importância na busca de melhores soluções para o problema de escalonamento de pessoal. Muitos modelos e enfoques diferentes têm sido propostos devido às diferenças existentes entre as organizações.

Com o surgimento das metaheurísticas, tais como Algoritmos Genéticos, Busca Tabu e *Simulated Annealing*, abriu-se um novo horizonte na resolução desse problema. Embora tais métodos não garantam a obtenção da solução ótima, eles são providos de mecanismos para escapar de mínimos (máximos) locais.

Em [3] foi desenvolvido o método denominado SSAH (*Specialized Shift Assignment Heuristic*) para designar atendentes satisfazendo a demanda de chamadas, procurando atender as preferências e prioridades de horários dos atendentes. Em [4] foi proposto um operador de mutação eficaz e aplicam com sucesso ao problema programação de atendentes. Em [5] foi proposto uma solução para a elaboração e designação de jornadas de trabalho em uma central telefônica de atendimento. Utilizou a técnica de Simulação para determinar o número de atendentes necessários para satisfazer a demanda histórica. Em seguida desenvolveu um modelo de Programação Linear Inteira (PLI) para resolver o escalonamento das jornadas dos atendentes. Este modelo foi resolvido com o pacote computacional LINGO. Finalmente, é realizada a designação das jornadas aos atendentes, maximizando a satisfação destes com relação aos seus horários de trabalho. Em [6] foram apresentados modelos matemáticos e algoritmos de solução para uma família de problemas de programação. O principal objetivo é a minimização do número de funcionários necessários para executar todas as tarefas diárias no horizonte de planejamento. O problema é decomposto em duas etapas: a primeira etapa é definição da seqüência dos períodos de trabalho e folga para cada empregado, bem como a definição da tarefa diária a ser realizada em cada período de trabalho para cada funcionário, é formulado como um problema de cobertura. O segundo passo é formulado como um problema de viabilidade resolvido heurísticamente por uma seqüência de problemas de transporte. Apresentaram resultados computacionais em casos da vida real associada a um Call Center de emergência. Em [7] foi desenvolvido uma metodologia de escalonamento que combina Programação Linear, Busca Tabu e Simulação. Em um caso real, a aplicação da abordagem híbrida reduziu os custos

totais diários de pessoal de um Call Center em quase 18%. Em [8] foram descritos métodos simples para turnos de trabalho, métodos existentes para única habilidade em Call Center. Introduzem um novo método para programação de turnos em Call Centers multi-habilidades. O problema de tem como base um modelo de programação linear.

Neste trabalho, propomos um modelo de programação de turnos com intervalos de descanso previamente determinados e a designação destes aos atendentes. O problema é dividido em dois, a saber: primeiro, a otimização dos turnos de trabalho fechando um ciclo semanal; segundo, a designação destes turnos aos atendentes para o período de cinco semanas. Neste modelo usamos a metaheurística Evolução Diferencial Discreta para otimizar ambos os problemas.

O restante deste artigo está organizado como segue: A caracterização e a formulação do problema são tratadas na seção 2. O tópico relativo à metaheurística Evolução Diferencial Discreta é tratado na seção 3. O modelo proposto abordando a otimização da programação dos turnos de trabalho e da designação dos turnos aos atendentes de um Call Center é apresentada na seção 4. Os experimentos e resultados são exibidos na seção 5. Por último, nós concluímos nosso artigo na seção 6.

2 Caracterização e formulação do problema

Dados de um determinado Call Center foram utilizados para aplicação, neste trabalho. No Call Center as escalas dos operadores de telefonia são feitas manualmente com auxílio de planilhas do Excel. Cada mês é feita uma estimativa do número de chamadas utilizando dados do mês anterior, para estimar a quantidade necessária de atendentes. Em seguida, as escalas dos atendentes são elaboradas. Essas razões motivaram o desenvolvimento deste trabalho de forma a otimizar os turnos de trabalho e sua designação aos atendentes, cujo objetivo principal é a redução do número de atendentes. Aqui, turno de trabalho é definido como o período de trabalho diário de um atendente.

Os atendentes são contratados para trabalhar seis horas e vinte minutos por dia, com duas pausas de dez minutos cada e um intervalo de vinte minutos para descanso e alimentação. A primeira pausa, de dez minutos, deve satisfazer a seguinte restrição: início entre uma hora e duas horas após início do turno. A segunda pausa, de dez minutos, deve satisfazer a seguinte restrição: início entre uma hora e dez minutos, e duas horas e dez minutos antes do término do turno. O intervalo deve satisfazer as seguintes restrições: início entre uma hora e duas horas após o término da primeira pausa; e término entre uma hora e duas horas antes do início da segunda pausa.

O período de planejamento adotado foi o ciclo de uma semana. Como as demandas entre os dias úteis são similares, também entre os sábados e os domingos, considerou-se a demanda igual para todos os dias de semana, assim como para os sábados e os domingos entre si. Simplificando, desta forma, o ciclo para três momentos (dias), ou seja, dia útil, sábado e domingo. Este período de planejamento foi fragmentado de dez em dez minutos, compondo 432 fragmentos. Um turno é, portanto, composta por 38 fragmentos de dez minutos cada. O início de um turno pertence ao conjunto {00:00, 00:30, ..., 23:00, 23:30}. Devido a problemas com transporte após a meia-noite e

antes das 6 horas da manhã foram impostas restrições para início e fim de turno, nesse período. Mas, permitiu-se violação desses limites em até vinte minutos, para fim de turno de trabalho. As demandas utilizadas no problema são obtidas de um simulador, de acordo com o quadro 1.

Quadro 1: Demandas de um mês de funcionamento do Call Center.

Domingo			Sábado			Dia útil		
Tempo		Demanda	Tempo		Demanda	Tempo		Demanda
Início	Fim		Início	Fim		Início	Fim	
00:00	01:00	3	00:00	01:00	3	00:00	01:00	2
01:00	02:00	2	01:00	02:00	3	01:00	02:00	2
02:00	03:00	1	02:00	03:00	1	02:00	03:00	1
03:00	04:00	1	03:00	04:00	3	03:00	04:00	1
04:00	05:00	1	04:00	05:00	1	04:00	05:00	1
05:00	06:00	2	05:00	06:00	1	05:00	06:00	2
06:00	07:00	2	06:00	07:00	3	06:00	07:00	3
07:00	08:00	3	07:00	08:00	4	07:00	08:00	5
08:00	09:00	5	08:00	09:00	10	08:00	09:00	16
09:00	10:00	7	09:00	10:00	15	09:00	10:00	25
10:00	11:00	8	10:00	11:00	18	10:00	11:00	28
11:00	12:00	9	11:00	12:00	17	11:00	12:00	27
12:00	13:00	8	12:00	13:00	18	12:00	13:00	22
13:00	14:00	8	13:00	14:00	13	13:00	14:00	24
14:00	15:00	6	14:00	15:00	12	14:00	15:00	23
15:00	16:00	5	15:00	16:00	11	15:00	16:00	22
16:00	17:00	5	16:00	17:00	10	16:00	17:00	23
17:00	18:00	5	17:00	18:00	11	17:00	18:00	20
18:00	19:00	5	18:00	19:00	10	18:00	19:00	13
19:00	20:00	5	19:00	20:00	8	19:00	20:00	11
20:00	21:00	4	20:00	21:00	7	20:00	21:00	8
21:00	22:00	4	21:00	22:00	6	21:00	22:00	6
22:00	23:00	4	22:00	23:00	5	22:00	23:00	5
23:00	00:00	2	23:00	00:00	3	23:00	00:00	3

A designação dos turnos aos empregados será feito para um período de 5 semanas. As restrições de trabalho impostas, pela empresa, para esse período para cada empregado, são: deve ser designado para trabalhar em, exatamente, três sábados, um domingo e todos os dias de semana (dias úteis).

2.1 Formulação do Problema

A formulação matemática do problema foi adaptado da formulação original de Programação Inteira (PI) do problema de programação de turnos (*shift scheduling problem*) [9] e da formulação em uma aplicação [1]:

$$\text{Minimizar } \sum_n \sum_d \sum_t c_n x_{ndt} \quad (1)$$

$$\text{Minimizar } \sum_n \gamma_n \quad (2)$$

$$\text{Sujeito a: } \sum_n \sum_t (a_{ndtp} - b_{ndtp}) x_{ndt} \geq R_{dp} \text{ para cada fragmento } p, \text{ dia } d \quad (3)$$

$$\sum_d x_{ndt} \leq 1 \quad \text{para cada empregado } n, \quad (4)$$

$$\sum_{d_k} \sum_t x_{nd_k t} \leq W_{nd_k} \quad \text{para cada empregado } n, \text{ tipo} \quad (5)$$

$$x_{ndt} \text{ binário} \quad \text{para cada empregado } e, \text{ dia} \quad (6)$$

onde

- n → índice para os empregados
- d → índice para os dias
- t → índice para os turnos
- p → índice para os fragmentos de período de planeamento
- d_k → índice para os dias específicos
- c_n → custo de ter um empregado n para trabalhar
- γ_n → Quadrado da soma das variações dos início dos turnos do empregado n
- a_{ndtp} → 1, se o fragmento p é um fragmento do turno t , do dia d , designado ao empregado n ; 0, em caso contrário
- b_{ndtp} → 1, se o fragmento p é um fragmento de pausa ou de intervalo do turno t , do dia d , designado ao empregado n ; 0, em caso contrário
- x_{ndt} → 1 se o empregado n está designado para o dia d , no turno t ; 0, em caso contrário
- W_{nd_k} → número máximo de dias de trabalho do tipo de dia k (sábado, domingo, dia útil)
- R_{dp} → número de atendentes necessários para atender a demanda no fragmento p , do dia d

O objetivo do modelo é minimizar o custo da programação dos turnos de trabalho (1) e a variação do momento inicial dos turnos dos empregados (2), sujeito às restrições: a quantidade de empregados necessária está presente em cada fragmento de período de planeamento, considerando as pausas e intervalos (2), para cada empregado deve ser designado no máximo um turno por dia (4), quantidade máxima de certos dias de trabalho (por exemplo, máximo 3 sábados no período de planeamento) (5) e não existem atribuições parciais de atendentes aos turnos (6).

Diversas técnicas são utilizadas para otimizar a programação de turnos de trabalho conforme mencionado na seção anterior. Neste artigo, utiliza-se algoritmo Evolução Diferencial (ED) [10], [11]. O problema abordado, neste artigo, tem característica discreta, assim, será utilizado o algoritmo Evolução Diferencial Discreta.

3 Evolução Diferencial

Diversas técnicas são utilizadas para otimizar a programação de turnos de trabalho conforme mencionado na introdução deste artigo. Aqui, utilizamos o algoritmo Evolução Diferencial (ED) [10],[11], considerado um Algoritmo Evolutivo, pois emprega operadores como cruzamento, mutação e seleção. O ED clássico é um

algoritmo de otimização global estocástico, baseado em população, que pode ser usado para minimizar funções não-lineares e não-diferenciáveis de espaço contínuo, com parâmetros de valores reais. Assim, como outros Algoritmos Evolutivos, ED inicia com uma população de indivíduos, seguindo-se dos operadores de mutação, *crossover* e seleção, respectivamente, até que um critério de término seja satisfeito.

Um i -ésimo indivíduo da população de NP indivíduos na geração g tem n cromossomos, ou seja,

$$X_i(g) = [x_{i,1}(g), x_{i,2}(g), \dots, x_{i,n}(g)], \quad i = 1, 2, \dots, NP \quad (7)$$

onde, $x_{i,j}(g)$ são os cromossomos (componentes) do i -ésimo indivíduo.

Na operação de mutação, para cada indivíduo i (indivíduo alvo) da população, são escolhidos três outros indivíduos denominados $X_{r1}(g)$, $X_{r2}(g)$ e $X_{r3}(g)$, com índices aleatórios $r1, r2$ e $r3 \in \{1, 2, \dots, NP\}$ mutuamente diferentes, isto é, $i \neq r1 \neq r2 \neq r3$. Um indivíduo mutante, $V_i(g+1)$, é gerado pela combinação destes três indivíduos, de acordo com:

$$V_i(g+1) = X_{r1}(g) + F[(X_{r2}(g) - X_{r3}(g))] \quad (8)$$

onde F é um fator de escala.

Na operação de *crossover*, um indivíduo chamado indivíduo tentativa, $U_i(g+1)$, cujos cromossomos é determinado, de acordo com:

$$u_{i,j}(g+1) = \begin{cases} v_{i,j}(g+1), & \text{se } rand_{i,j} \leq CR \text{ ou } j = I_{rand} \\ x_{i,j}(g), & \text{se } rand_{i,j} > CR \text{ ou } j \neq I_{rand} \end{cases} \quad (9)$$

onde CR é a taxa de *crossover*, $rand_{i,j}$ é um número escolhido aleatoriamente do intervalo $[0, 1[$ e I_{rand} é um número inteiro escolhido aleatoriamente do conjunto $\{1, 2, \dots, n\}$ e j é a ordem do cromossomo.

A seleção é efetuada de acordo com a formulação a seguir:

$$X_i(g+1) = \begin{cases} U_i(g+1), & \text{se } f(U_i(g+1)) \leq f(X_i(g)) \\ X_i(g), & \text{em caso contrário} \end{cases} \quad (10)$$

As vantagens de ED são: a estrutura simples, a velocidade na obtenção das soluções e a robustez.

4.1 Evolução Diferencial Discreta (EDD)

O principal obstáculo na aplicação de ED em problemas combinatórios é devido à sua natureza contínua. Com relação à resolução de problemas de otimização combinatória com características discretas, novas abordagens de ED discretas têm sido propostas na literatura. Outra dificuldade seria a de que em problemas de otimização combinatória, os vetores diferenciais não têm uma correspondência clara com as direções no espaço de busca.

O operador aritmético de reprodução utilizado no ED é simples, porém, a maneira pela qual está definido, torna praticamente impossível a aplicação do ED original a outros espaços [12]. Outro problema que se encontra é que, no ED original não se consegue manter o fechamento, quando da aplicação no domínio discreto. Assim, não pode ser usado em problemas de otimização discreta diretamente [13].

Alguns trabalhos envolvendo aplicações de EDD em problemas de *scheduling* podem ser citados, tais como: *Flowshop Scheduling* [14], *Permutation Flowshop Scheduling* [15], *Bi-Criteria No-Wait Flowshop Scheduling* [16], *Production*

Scheduling [17], *Task Scheduling in Heterogeneous Computing Systems* [18], *Resource Planning and Scheduling of Payload for Satellite* [19], *Multi-Objective Flowshop Scheduling* [20] e *Blocking Flowshop Scheduling* [21].

4 Modelo Proposto

O problema é dividido em dois problemas, a saber: primeiro, determinação dos turnos de trabalho para o período de uma semana e, segundo, designação desses turnos aos empregados para o período de 5 semanas.

Na determinação dos turnos de trabalho o problema é tratado para o ciclo de uma semana. As demandas para dias de semana (dias úteis) possuem padrões similares, assim, por simplificação, uma semana será reduzida a um dia útil, cujas demandas são as médias aritméticas das demandas dos dias de semana. Reduzindo a quantidade de restrições e de variáveis, pois reduz um problema com restrições e variáveis para 7 dias, para um problema com 3 dias. O Segundo problema é composto de um ciclo de 5 semanas. Para a resolução de ambos os problemas será utilizado Evolução Diferencial Discreta.

4.1 Determinação dos Turnos de Trabalho via EDD

Assim, como outros Algoritmos Evolutivos, EDD inicia com uma população de indivíduos, seguindo-se dos operadores de mutação, *crossover* e seleção, respectivamente, até que um critério de término seja satisfeito.

Codificação dos indivíduos e População Inicial. Uma solução do problema é composta por diversos tipos de turnos possíveis do período de planejamento. Assim, cada indivíduo (*i*) representa uma programação semanal, cada cromossomo (*c*) representa um turno e seus genes (*g*) representam a caracterização do turno. Aqui, os cromossomos possuem 5 genes, onde: o primeiro indica se o turno faz parte ou não da programação do período de planejamento; o segundo representa o momento inicial do turno; o terceiro representa o início da primeira pausa; o quarto representa o início do intervalo e o quinto representa o início da segunda pausa. Assim, o primeiro gene é um valor binário, o segundo é um valor pertencente ao conjunto $\{1, 2, \dots, 432\}$ que representa um dos 432 fragmentos do período de planejamento (uma semana simplificada em dias úteis, sábado e domingo) e o terceiro, quarto e quinto genes são valores pertencentes ao conjunto $\{1, 2, \dots, 38\}$ que representa um dos 38 fragmentos de um turno. Para a população inicial (geração $g = 0$) gerou-se valores aleatórios para as componentes de cada indivíduo dentro da variação dos limites do problema.

Computação do Fitness. Pretende-se minimizar a quantidade diária de turnos de trabalho que é equivalente a somatória dos valores do primeiros genes de cada cromossomo.

Mutação. Uma população mutante de indivíduos é gerada a partir de uma população chamada população alvo. Para um indivíduo alvo $Y_i(\mathbf{g})$, três indivíduos $X_{r1}(\mathbf{g})$, $X_{r2}(\mathbf{g})$ e $X_{r3}(\mathbf{g})$ da população alvo são escolhidos aleatoriamente, distintos entre si e de $Y_i(\mathbf{g})$. Um indivíduo mutante $V_i(\mathbf{g} + \mathbf{1})$, é gerado de acordo com $V_i(\mathbf{g} + \mathbf{1}) = X_{r1}(\mathbf{g}) \oplus F \otimes [X_{r2}(\mathbf{g}) \ominus X_{r3}(\mathbf{g})]$. Os símbolos \oplus , \otimes e \ominus substituem os símbolos tradicionais de adição, multiplicação e subtração, respectivamente. Para as variáveis inteiras (segundo, terceiro quarto e quinto genes), definimos: \ominus é a diferença entre os genes do c -ésimo cromossomo do indivíduo X_{r2} pelos genes do c -ésimo cromossomo do indivíduo X_{r3} ; \otimes é a multiplicação da constante F pelo resultado da operação \ominus , arredondando para o número inteiro abaixo; \oplus adição de X_{r1} com o resultado da operação \otimes , limitando o resultado ao intervalo de factibilidade, ou seja, valor fora do intervalo iguala-se ao extremo mais próximo. Para as variáveis binárias (primeiro gene), definimos: \ominus é o operador binário AND, ou seja, retorna 1 se ambos os operandos são iguais a 1, e 0 caso contrário; \otimes mantém o resultado da operação \ominus ; \oplus é o operador binário OR, ou seja, retorna 1 quando pelo menos um dos operandos é 1, e 0 caso contrário.

Crossover. Um indivíduo chamado indivíduo tentativa $U_i(\mathbf{g} + \mathbf{1})$ é produzido a partir dos genes do indivíduo mutante $V_i(\mathbf{g} + \mathbf{1})$ e dos genes do indivíduo alvo $Y_i(\mathbf{g})$. Os cromossomos do indivíduo tentativa são formados de acordo com a operação (9).

Seleção. A operação de seleção utilizada, neste trabalho, é a mesma da versão original de ED, de acordo com a operação (10).

Parâmetros. Algumas pesquisas mostram que valores fixos de F e CR podem levar à convergência prematura e pior desempenho [22]. Para resolver este problema em [23] foi descrita uma estratégia de ajustamento adaptativo, de acordo com:

$$F = F_0 \cdot e^{-a_1 \left(\frac{G}{G_{max}} \right)} \quad (11)$$

$$CR = CR_0 \cdot e^{-a_2 \left(\frac{G}{G_{max}} \right)} \quad (12)$$

onde G_{max} e G são, respectivamente, as gerações evolutivas máxima e atuais; F_0 e CR_0 são, respectivamente, o valor inicial de fator de escala e a taxa de *crossover*; a_1 e a_2 são constantes positivas. Utilizamos a base 2, pois, empiricamente, trouxe melhores resultados que a base e .

Busca Local. Um aumento adequado na aleatorização é fundamental para a obtenção de melhorias significativas no funcionamento de ED [24]. A aleatorização foi empregada no EDD de forma que fossem inseridos movimentos de busca local. Para isso, introduziu-se uma mutação extra aplicada ao terceiro, quarto e quinto genes. Nesta mutação novos genes são determinados da mesma maneira como foram gerados na população inicial. O fator desta mutação chamado ME, é dado por:

$$ME = ME_0 \cdot 2^{-a_3 \left(\frac{G}{G_{max}} \right)} \quad (13)$$

onde ME_0 é uma constante definida empiricamente. Aleatorização para modificar os genes binários foi introduzida após o *crossover*. Caso o gene binário seja 1 aplica-se a taxa denominada $turn_{out}$ que é a probabilidade do gene ser permutado pelo 0 (o turno passa a não pertencer a programação) e, caso o gene seja 0 aplica-se a taxa denominada $turn_{in}$ que é a probabilidade do gene ser permutado pelo 1 (o turno passa a pertencer a programação). As taxas estão definidas de acordo com:

$$turn_{out} = turn_{out_0} \cdot 2^{-a_4 \left(\frac{G}{G_{max}} \right)} \quad (14)$$

$$turn_{in} = turn_{in_0} \cdot 2^{-a_5 \left(\frac{G}{G_{max}} \right)} \quad (15)$$

onde, $turn_{out_0}$ e $turn_{in_0}$ são constantes definidas empiricamente.

4.2 Designação dos Turnos aos Empregados via EDD

As estruturas dos EDDs para a determinação dos turnos e para a designação destes aos empregados possuem algumas similaridades. Assim, alguns partes da abordagem pode ser suprimida.

Codificação dos indivíduos e População Inicial. Uma solução do problema é composta pelos turnos determinados na seção anterior designados aos empregados no período de planejamento de 5 semanas. Assim, cada indivíduo (i) representa uma programação de cinco semanas, cada cromossomo (c) representa um empregado designado aos turnos (chamaremos de *tour*) no período de 5 semanas e seus genes (g) representam a caracterização do empregado designado a um *tour*. Aqui, os cromossomos possuem 5 genes, onde: o primeiro, indica se o empregado faz parte ou não da programação do período de planejamento; o segundo, representa uma das 50 possíveis combinações das restrições de sábado e domingo (cada empregado deve trabalhar exatamente 3 sábados e um domingo); o terceiro, representa qual turno de sábado está designado ao empregado; o quarto, representa qual turno de domingo está designado ao empregado; o quinto, representa qual turno de dia útil está designado ao empregado. Para a população inicial (geração $g = 0$) gerou-se valores aleatórios para as componentes de cada indivíduo dentro da variação dos limites do problema.

Computação do Fitness. Pretende-se minimizar a quantidade total de empregados e a variação entre os inícios dos turnos de um mesmo empregado, para que haja pouca variação de um dia para outro em relação ao momento de início de trabalho. Desta forma, este problema de designação dos turnos aos empregados possui duas funções para minimização. Assim, temos um problema de otimização bi-objetivo a ser resolvido. Segundo [25] o Método da Soma Ponderada é a técnica mais utilizada para transformar um problema multi-objetivo para um problema com um único objetivo. Esta é a técnica utilizada neste problema. Assim, a função *fitness* para o indivíduo i é a somatória do número de empregados adicionado à somatória do quadrado das variações iniciais entre os dias de trabalho. Para simplificação, considerou-se que todos os dias úteis o empregado inicia sempre no mesmo horário (o mesmo turno para os dias de semana), assim como para os sábados e os domingos, não necessariamente iguais entre si, de acordo com:

$$F_i = \sum_e x_e + w \sum_e [(ITS_e - ITD_e)^2 + (ITS_e - ITU_e)^2 + (ITD_e - ITU_e)^2] x_e \quad (16)$$

onde x_e é 1 se o empregado é designado à programação do período de planejamento e 0 caso contrário, w é o peso que é dado a função de variação de início de turno, ITS_e é o início do turno de sábado para o empregado e , ITD_e é o início do turno de domingo para o empregado e , ITU_e é o início do turno de dia útil para o empregado e .

Mutação, Crossover e seleção. As operações de mutação, crossover e seleção são iguais as do EDD para a determinação de turnos.

Parâmetros. Empiricamente, a taxa de cruzamento que obteve melhores resultados foi para o valor fixo de $CR = 0,8$. Para a mutação utilizou-se a mesma estratégia de ajustamento adaptativo do EDD para determinação de turnos.

Busca Local. Aqui, também, foi introduzida aleatorização para melhorias no funcionamento deste EDD. Para o primeiro gene foi similar ao do EDD anterior e para os demais genes de forma aleatória, ou seja, na mutação o valor é substituído por qualquer valor dentro dos limites do problema, ou seja, na operação de mutação novos genes são determinados da mesma maneira como foram gerados na população inicial, com taxas de mutação iguais para todos esses genes, de acordo com:

$$T = T_0 \cdot 2^{-a_6 \left(\frac{G}{G_{max}}\right)} \quad (17)$$

onde T_0 é uma constante definida empiricamente. Para os genes binários a aleatorização foi introduzida de forma similar ao do EDD anterior. Desta forma, caso o gene binário seja 1 aplica-se a taxa denominada $empreg_{out}$ que é a probabilidade do gene ser permutado pelo 0 (o empregado passa a não ter turnos designados a ele) e, caso o gene seja 0 aplica-se a taxa denominada $empreg_{in}$ que é a probabilidade do gene ser permutado pelo 1 (o empregado passa a ter turnos designados a ele). As taxas estão definidas de acordo com:

$$empreg_{out} = empreg_{out_0} \cdot 2^{-a_7 \left(\frac{G}{G_{max}}\right)} \quad (18)$$

$$empreg_{in} = empreg_{in_0} \cdot 2^{-a_8 \left(\frac{G}{G_{max}}\right)} \quad (19)$$

onde, $empreg_{out_0}$ e $empreg_{in_0}$ são constantes definidas empiricamente.

5 Experimentos e Resultados

Para validação do modelo proposto com aplicação de EDD é feita a comparação dos resultados do modelo proposto para os dois problemas com os de dois modelos matemático de Programação Linear Inteira (PLI), para cada problema.

5.1 Experimentos e Resultados: Determinação de Turnos de Trabalho

Para a implementação do modelo PLI para a determinação de turnos de trabalho foi desenvolvido um “Gerador de Turnos” que tem por finalidade gerar diversos turnos considerando as restrições de horário. Para a resolução do modelo foi utilizado o serviço de recursos de otimização *on line* NEOS Server (*Network Enabled Optimization System*). Este serviço é um portal para *softwares* de otimização e serviços pela *Internet* e foi desenvolvido pela Divisão de Ciência da Computação e Matemática do *Argonne National Laboratory*. O arquivo para envio ao site deve ser em formato GAMS (*General Algebraic Modeling System*). Entre os aplicativos de otimização oferecidos pelo NEOS, escolheu-se, para a resolução, o MOSEK e o XPRESS. Os resultados encontrados na resolução do problema foram os seguintes: 23 turnos de trabalho para domingos, 61 para dias úteis e 42 para sábados.

Na resolução do problema por EDD, os parâmetros que mostraram melhores resultados, foram:

- 5 indivíduos;
- $CR_0 = F_0 = turn_{out_0} = 0,1$;
- $ME_0 = turn_{in_0} = 0,01$;
- $a_1 = a_2 = a_3 = a_4 = a_5 = 3$.

Foram adotados (empiricamente) 400 cromossomos para cada indivíduo da população. Para todos os genes binários da população inicial foi adotado o valor 1. Como critério de parada 75.000 gerações.

Foram feitas 10 execuções do EDD. Obteve-se o valor igual a 126 para a função objetivo em 4 execuções. Em outras 5 execuções obteve-se o valor igual 127 e nas outras 2 execuções encontrou-se o valor 128. Na figura 1 é mostrado o gráfico da relação entre o *fitness* e o número de gerações da execução de EDD que obteve o melhor resultado. O resultado 126 para a função objetivo foi encontrado na geração 86.522.

O EDD foi programado em linguagem *object Pascal* no DELPHI 7 e executado em um microcomputador Pentium IV, 2.8GHz e 1GB de RAM. O tempo médio das 10 execuções foi igual a 159,6 segundos.

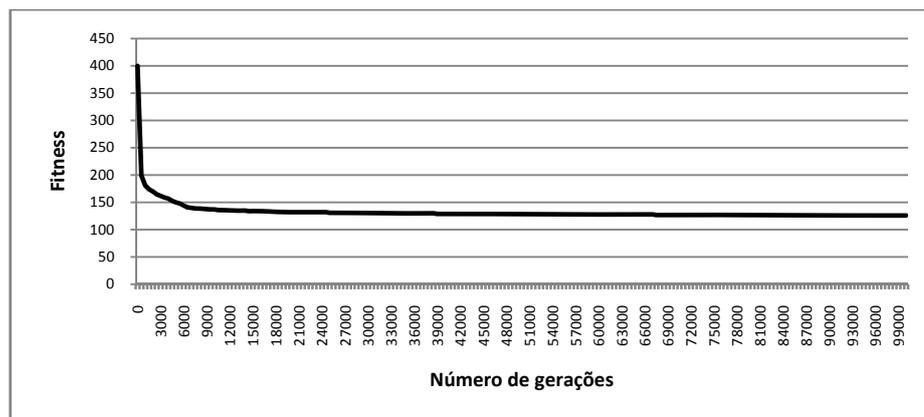


Figura 1: relação entre fitness e gerações

5.2 Experimentos e Resultados: Designação dos Turnos aos Empregados

O valor adotado para w na função a ser minimizada nesse problema foi igual a 20.000^{-1} , determinado empiricamente.

Para a resolução do modelo PLI, também, foi utilizado o serviço de recursos de otimização *on line* NEOS Server (*Network Enabled Optimization System*). Parte do arquivo de resposta consta as seguintes mensagens “*Out of Memory in New or GetMem*” e “*Out of Memory while executing line 2505*”, ou seja, falta de memória para armazenamento de parte das equações de restrição do problema.

Foram adotados, também, 400 cromossomos para cada indivíduo da população. Para todos os genes binários da população inicial, também, foi adotado o valor 1. Como critério de parada 25.000 gerações.

Na resolução do problema por EDD para designação de turnos aos empregados, os parâmetros que mostraram melhores resultados, foram:

- 5 indivíduos;
- $CR_0 = 0,8$ e $a_2 = 0$ (ou seja, valor fixo para CR)
- $empreg_{out_0} = 0,1$;
- $F_0 = T_0 = empreg_{in_0} = 0,01$;
- $a_1 = 5$;
- $a_6 = a_7 = a_8 = 3$.

Foram feitas 10 execuções do EDD. Em 7 execuções obteve-se o resultado ótimo (115 empregados) para a função que determina a quantidade de empregados. Em outras 3 execuções obteve-se o valor igual 116. Entre as 7 execuções, a que obteve melhor resultado em relação a outra função que está relacionada a variação entre os inícios dos turnos de um mesmo empregado foi de 1,98. Para esclarecimento em relação a esta segunda função a ser minimizada, ou seja, em relação a minimização da variação inicial entre os turnos de um mesmo empregado, foi feito um histograma. A figura 2 mostra o histograma da frequência de empregados em relação a máxima variação entre os inícios de turnos.

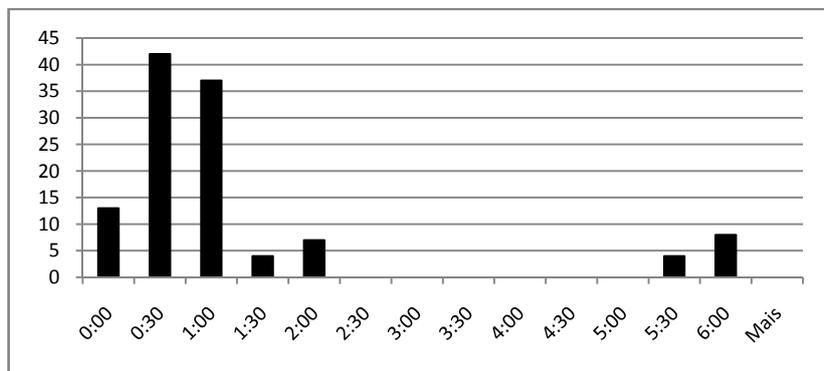


Figura 2: Frequência de empregados em relação a variação máxima entre inícios de turnos de um mesmo empregado

Verifica-se que vários empregados são designados a turnos em que o momento de entrada no trabalho (início de seu turno) tem pouca variação e dentre eles 13 empregados são designados a turnos cuja entrada no turno é igual para todos os dias e 79 empregados cuja variação é de até 60 minutos. A variação máxima é de 6 horas, neste caso, vale lembrar que a variação entre dois turnos consecutivos em um mesmo dia é de 30 minutos, diferente apenas entre os dois primeiros que é de 6 horas em que 8 empregados tiveram designação em ambos os turnos, justificando parte das maiores variações encontradas.

6 Conclusões

Neste artigo, foi descrito um procedimento para resolver o problema de programação de turnos durante um período de 5 semanas de trabalho em um *Call Center*. O problema foi dividido em dois problemas: (i) determinação dos turnos de trabalho para o ciclo de uma semana e (ii) designação dos turnos aos empregados para o período de 5 semanas. Para o primeiro problema resolveu-se por um modelo de PLI pelo site NEOS e por EDD comparando-se os resultados. O EDD encontrou resultado ótimo em algumas execuções e encontrados resultados próximos do ótimo em outras execuções. Para o segundo problema tentou-se a resolução também por PLI pelo site NEOS, obtendo-se como resposta falta de memória para parte de dados de várias restrições do problema. Assim, resolveu-se o problema apenas por EDD, justificando sua aplicação.

Os resultados foram considerados satisfatórios, uma vez que encontrou-se resultado ótimo para uma das funções da função bi-objetivo e bons resultados relacionados a outra função da função bi-objetivo, pois vários empregados foram designados a turnos cujos inícios entre dias diferentes sofreram pouca variação, ou seja, procurando a satisfação dos empregados em relação a variação do momento de entrada diária no trabalho.

Conclui-se que o modelo proposto com aplicação do EDD para o problema de programação de turnos de trabalho em *Call Center* foi adequada, pois gerou resultados satisfatórios em tempo computacional aceitável, uma vez que encontrou-se, para o primeiro problema, em alguns testes, o resultado ótimo e em outros testes, resultados próximos do ótimo e para o segundo problema resultados, também ótimos e próximos do ótimo.

Referências Bibliográficas

1. Hojati M., Patil A. S.: An Integer Linear Programming-Based Heuristic for Scheduling Heterogeneous, Part-Time Service Employees. *European Journal of Operational Research*, vol. 209, pp. 37--50 (2011)
2. Ernst A. T., Jiang H., Krishnamoorthy M., Owens B., Sier D.: An Annotated Bibliography of Personnel Scheduling and Rostering. *Annals of Operations Research*, vol. 127, pp. 21--144 (2004)

3. Thompson G. M.: Assigning Telephone Operators to Shifts at New Brunswick Telephone Company. *Interfaces*, vol. 27, pp. 1--11 (1997)
4. Yoshimura K., Nakano R.: Genetic Algorithm for Information Operator Scheduling. The 1998 IEEE International Conference on Evolutionary Computation Proceedings, pp. 277--282 (1998)
5. Barboza A. O.: Aplicação de Algumas Técnicas da Pesquisa Operacional na Otimização de Horários de Atendentes em Central Telefônica. Dissertation, UFPR, Curitiba (2000)
6. Caprara A., Monaci M., Toth P.: Models and Algorithms for Staff Scheduling Problems. IV ALIO/EURO Workshop, Abstracts Applied Combinatorial Optimization. Pucón, Chile, (2002)
7. Saltzman R. M.: A Hybrid Approach to Minimize the Cost of Staffing a Call Center. *International Journal of Operations and Quantitative Management*, vol. 11, n. 1, pp. 1--14 (2005)
8. Bhulai S., Koole G., Pot A.: Simple Methods for Shift Scheduling in Multiskill Call Centers. *Manufacturing & Service Operations Management*, vol. 10, n. 3, pp. 411--420 (2008)
9. Dantzig G. B.: A Comment on Edie's "Traffic Delays at Toll Booths". *Operations Research*, vol. 2, n. 3, pp. 339--341 (1954)
10. Price K., Storn R.: Differential Evolution: A Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces. Berkeley: Technical Report TR-95-012, International Computer Science Institute (1995)
11. Price K., Storn R.: Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *Journal of Global Optimization*, vol. 11, pp. 341--359. (1997)
12. Pampará G., Engelbrecht A. P., Franken N.: Binary Differential Evolution. 2006 IEEE Congress on Evolutionary Computation, pp. 1873--1879 (2006)
13. Deng C., Zhao B., Yang Y., Deng A.: Novel Binary Differential Evolution Algorithm for Discrete Optimization. 2009 Fifth International Conference on Natural Computation, pp. 346--349 (2009)
14. Onwubolu G., Davendra D.: Scheduling Flow Shops Using Differential Evolution Algorithm. *European Journal of Operational Research*, vol. 171, pp. 674--692 (2006)
15. Pan Q. K., Tasgetiren M. F., Liang Y. C.: A Discrete Differential Evolution Algorithm for the Permutation Flowshop Scheduling Problem. *Computers & Industrial Engineering*, vol. 55, pp. 795--816 (2008)
16. Pan, Q. K.; Wang, L. & Qian, B.: A Novel Differential Evolution Algorithm for Bi-Criteria No-Wait Flowshop Scheduling Problems. *Computers & Operations Research*, vol. 36, pp. 2498--2511 (2009)
17. Jun Z., Li L. Q., Wei W.: Models and Algorithms of Production Scheduling in Tandem Cold Rolling. *Acta Automatica Sinica*, vol. 34, n. 5, pp. 565--573 (2008)
18. Kang Q., He H.: A Novel Discrete Differential Evolution Algorithm for Task Scheduling in Heterogeneous Computing Systems. Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics, pp. 5006--5011 (2009)
19. Li J.: Resource Planning and Scheduling of Payload for Satellite with a Discrete Binary Version of Differential Evolution. 2009 IITA International Conference on Control, Automation and Systems Engineering, pp. 62--65 (2009)
20. Qian B., Wang L., Huang D. X., Wang W. L., Wang X.: An Effective Hybrid DE-based Algorithm for Multi-Objective Flowshop Scheduling with Limited Buffers. *Computers and Operations Research*, vol. 36, n. 1, pp. 209--233 (2009)
21. Wang L., Pan Q. K., Suganthan P. N., Wang W. H., Wang Y. M.: A Novel Hybrid Discrete Differential Evolution Algorithm for Blocking Flowshop Scheduling Problems. *Computers & Operations Research*, vol. 37, pp. 509--520 (2010)
22. Noman N., Iba H. Accelerating Differential Evolution using an Adaptive Local Search. *IEEE Transaction on Evolutionary Computation*, vol. 12, n. 1, pp. 107--125 (2008)

23. Zhang M., Zhao S., LV J., Qian Y.: Multi-phase Urban Traffic Signal Real-Time Control with Multi-objective Discrete Differential Evolution. 2009 International Conference on Electronic Computer Technology, pp. 296--300 (2009)
24. Neri F., Tirronen V.: Recent Advances in Differential Evolution: A Survey and Experimental Analysis. Artificial Intelligence Review, vol. 33, pp. 61--106 (2010)
25. Osyczka, A.: Multicriterion Optimization in Engineering with Fortran Programs. Ellis Horwood Limited, England (1984)