

Método híbrido *Clustering Search* aplicado ao problema de roteamento periódico de veículos para coleta de lixo

Eliseu Araújo, Kelly Poldi e Antônio Chaves

UNIFESP – Rua Talin, 330 São José dos Campos/SP

Resumo — A classe de problemas de roteamento de veículos comumente possui o objetivo de minimizar o custo associado ao atendimento de um grupo de consumidores. Desta classe advém o Problema de Roteamento Periódico de Veículos (PRPV) que deve gerar um conjunto de rotas para cada dia tendo os custos globais minimizados e de modo a respeitar todas as restrições envolvidas. Neste trabalho é estudado o problema de coleta de lixo modelado com um PRPV. Para resolver este problema, é proposto um método híbrido chamado *Clustering Search* voltado exclusivamente ao PRPV, que procura combinar metaheurísticas e heurísticas de busca local em que a pesquisa é intensificada somente em áreas do espaço de busca que merecem atenção especial. Nos testes computacionais, um caso real de Ponte de Lima, um município de Portugal é analisado, e também algumas instâncias clássicas da literatura.

Palavras-Chave — Problema de roteamento periódico de veículos, coleta de lixo, *Clustering Search*.

I. INTRODUÇÃO

A velocidade do crescimento populacional acarreta verdadeiros problemas no ambiente urbano, afetando a qualidade de vida de milhões e milhões de pessoas em todo o mundo. Um dos problemas advindos do crescimento populacional é a acelerada produção de lixo urbano sólido, principalmente nas últimas décadas.

O destino do lixo urbano sólido é um problema cuja solução se torna cada vez mais difícil. Em Nova York, cidade do mundo que mais produz lixo, a média diária de lixo produzido é de 13 mil toneladas [6]. É natural então perguntar-se como gerenciar da forma mais eficiente possível, como no caso de Nova York, 1,5 quilo de lixo *per capita* por dia.

Um mau gerenciamento desse lixo pode comprometer o meio ambiente, a saúde da população no entorno, além dos custos significativos. Segundo [9] temos três fatores decisivos pra o estudo do gerenciamento desse lixo:

- A grande quantidade de lixo gerada, como o exemplo de Nova York, sendo que o índice brasileiro *per capita* está em torno de 0,5 a 1 kg/habitante/dia;
- Os impactos ambientais e à população no entorno onde os resíduos são depositados, principalmente

causados pelo chorume. Um líquido de odor forte e alto potencial de contaminação resultante do processo de putrefação dos resíduos, que podem contaminar o solo, rios e lençóis freáticos;

- Altos gastos financeiros do gerenciamento do lixo urbano sólido. No Brasil, os serviços de limpeza respondem, em média, por 7% a 15% do orçamento dos municípios. Sendo que, aproximadamente 50% dos custos totais de gerenciamento são de coleta e transporte.

A ideia deste trabalho foi modelar um caso real do problema de coleta de lixo na cidade de Ponte de Lima, em Portugal, como um problema de roteamento periódico de veículos (PRPV), atendendo à diversas restrições, como a capacidade de cada veículo disponível, tempo da rota aceito por cada veículo, entre outras, e aplicar o método *Clustering Search* no problema modelado, visando a construção de rotas para cada veículo com o objetivo de reduzir o custo total do processo (distância total percorrida pelos veículos) respeitando todas as restrições impostas.

Para tanto, foi executada uma revisão bibliográfica da literatura a respeito do PRPV e dos processos de coleta de lixo, do tratamento das restrições e das diferentes abordagens utilizadas para resolução do PRPV. Além disso, foi apresentada a necessidade de um estudo da técnica *Clustering Search*, e de sua aplicação ao PRPV para, por fim, desenvolver um algoritmo *Clustering Search* (CS) para o PRPV. Após o desenvolvimento, foram realizados testes computacionais com dados reais do município de Ponte de Lima, apresentando resultados satisfatórios, reduzindo em 30% a distância percorrida pelos veículos em relação à solução adotada atualmente no município.

Este artigo é organizado como se segue. A seção 2 apresenta uma formulação matemática para o problema de coleta de lixo. A seção 3 descreve as ideias básicas e componentes conceituais do CS. A seção 4 descreve o procedimento da heurística proposta para resolver o problema. Os resultados computacionais obtidos pela aplicação do método proposto estão presentes e discutidos na seção 5. Por fim, as conclusões são explanadas na seção 6.

II. FORMULAÇÃO MATEMÁTICA

O problema de coleta de lixo do município de Ponte de Lima, localizado em Portugal, opera com 5 veículos de diferentes capacidades, há 994 contêineres de lixo distribuídos de forma não uniforme pelo município para

serem atendidos e as coletas são feitas em 6 dias da semana. O objetivo central é traçar rotas para cada veículo com diferentes frequências de coleta nos contêineres e distribuição nos dias da semana de forma a realizar tal tarefa com custos mínimos.

Em [1] é apresentada uma formulação matemática para o problema de coleta de lixo, modelando-o como um PRPV. A Tabela 1 apresenta os índices, parâmetros e conjuntos do modelo. Uma agenda é definida como os dias num dado período de tempo que um dado cliente deve ser visitado. Assume-se que y_{ir} é a variável que realiza a atribuição do cliente i para a agenda r (recebendo valor 1 se o cliente i é visitado pela agenda r) e x_{ijkl} é a variável que indica se o veículo k visita o cliente j imediatamente após visitar o cliente i no dia l (recebendo valor 1 se a aresta ij estiver na rota). A formulação completa é apresentada pelas equações 1 a 10.

TABELA I NOTAÇÃO

i, j, h	Cliente
k	Veículo
l	Dia
r	Agenda
L	Horizonte de planejamento (dias)
N	Número de clientes
P	Número de depósitos
K_l	Número de veículos disponíveis no dia l
d_{ij}	Distância entre os clientes i e j (km)
q_i	Quantidade total de lixo do cliente i (Kg)
s_i	Duração do serviço no cliente i (minutos)
t_{ij}	Duração da rota de i a j (minutos)
Q_k	Capacidade do veículo k (Kg)
T_l	Duração máxima da rota no dia l (minutos)
C_i	Número de agendas possíveis para o cliente i
a_{rl}	Constante que indica um se o dia l pertence à agenda r
\mathcal{L}	Dias do horizonte de planejamento, $\mathcal{L} = \{1; \dots; L\}$
\mathcal{U}	Locais, $\mathcal{U} = \{v_0; v_1; \dots; v_{N+P}\}$, onde v_0 corresponde à garagem
\mathcal{U}_c	Clientes, $\mathcal{U}_c = \{v_1; v_2; \dots; v_N\}$
\mathcal{U}_p	Locais de descarga (depósitos) $ \mathcal{U}_p = P$, $\mathcal{U}_p = \{v_{N+1}; v_{N+2}; \dots; v_{N+P}\}$
K_l	Veículos disponíveis no dia l , $ K_l = K_l$

C_i Conjunto de agendas possíveis para o cliente i , $|C_i| = c_i$

$$\text{minimizar} \sum_{i \in \mathcal{U}} \sum_{j \in \mathcal{U}, j \neq i} \sum_{l \in \mathcal{L}} \sum_{k \in K_l} d_{ij} x_{ijkl} \quad (1)$$

Sujeito a:

$$\sum_{r \in C_i} y_{ir} = 1; i \in \mathcal{U}_c \quad (2)$$

$$\sum_{j \in \mathcal{U}, j \neq i} \sum_{k \in K_l} x_{ijkl} - \sum_{r \in C_i} a_{rl} y_{ir} = 0; i \in \mathcal{U}_c; \quad (3)$$

$l \in \mathcal{L}$

$$\sum_{i \in \mathcal{U}, i \neq h} x_{ihkl} - \sum_{j \in \mathcal{U}, j \neq h} x_{hjkl} = 0; \quad (4)$$

$$h \in \mathcal{U} \setminus \{v_0\}; l \in \mathcal{L}; k \in K_l$$

$$u_{ikl} - u_{jkl} + Q_k x_{ijkl} \leq Q_k - q_j; \quad (5)$$

$$i, j \in \mathcal{U} \setminus \{v_0\}; j \neq i; q_i + q_j \leq Q_k; l \in \mathcal{L};$$

$$k \in K_l$$

$$q_i \leq u_{ikl} \leq Q_k; i \in \mathcal{U} \setminus \{v_0\}; l \in \mathcal{L}; k \in K_l \quad (6)$$

$$\sum_{j \in \mathcal{U} \setminus \{v_0\}} x_{0jkl} \leq 1; l \in \mathcal{L}; k \in K_l \quad (7)$$

$$\sum_{i \in \mathcal{U}} \sum_{j \in \mathcal{U}, j \neq i} (t_{ij} + s_i) x_{ijkl} \leq T_l; l \in \mathcal{L}; k \in K_l \quad (8)$$

A função objetivo (1) minimiza a distância total percorrida pelos veículos em todos os dias da agenda planejada. As restrições (2) asseguram que uma agenda adequada é atribuída para cada cliente. As restrições (3) garantem que os clientes são visitados somente nos dias correspondentes em suas agendas. As restrições (4) garantem que a rota seja gerada adequadamente, assegurando que toda vez que o veículo chega a um determinado cliente, ele em seguida deixa tal cliente. As restrições (5) e (6) garantem que não haja repetições na rota. As restrições (7) asseguram que no máximo, cada veículo seja utilizado uma vez por dia. O limite de tempo para cada dia é assegurado nas restrições (8). As restrições (9) asseguram que todos os veículos passem por um depósito antes de, finalmente, se dirigirem à garagem. A restrição (10) impõe que as variáveis x e y sejam binárias

III. CLUSTERING SEARCH

O PRPV pertence à classe de problemas NP-difícil [13]. Sendo assim, os métodos heurísticos são as técnicas propostas mais utilizadas para a sua resolução. Dentre os métodos utilizados destacam-se Algoritmos Genéticos [16],

Greedy Randomized Adaptive Search Procedure (GRASP) [10], Busca Tabu [11], entre outros.

Este trabalho é modelado como um PRPV e para buscar boas soluções para o problema de coleta de lixo foi utilizado o método Clustering Search (CS).

O CS [2] é um método híbrido que busca combinar metaheurísticas e heurísticas de busca local, em que a busca é intensificada somente em regiões do espaço de busca que merecem atenção especial (regiões promissoras). O CS introduz uma inteligência e prioridade para a escolha de soluções para aplicar a busca local, ao invés de escolher aleatoriamente ou aplicar busca local em todas as soluções. Consequentemente, é esperado uma melhora no processo de convergência com uma diminuição no esforço computacional pois há uma aplicação mais racional das heurísticas.

O CS se atém a localizar áreas de busca promissora construindo-as em *clusters*. Um *cluster* é definido por um centro, c , que é geralmente, inicializado aleatoriamente e, posteriormente, ele tende progressivamente a pontos promissores no espaço de busca. O número de clusters, NC , deve ser fixada a priori.

O CS pode ser explicitado em quatro partes conceitualmente independentes: a metaheurística (SM), o componente de agrupamento (IC), um módulo de análise (AM) e a busca local (LS).

A Fig. 1 apresenta um fluxograma conceitual do CS.

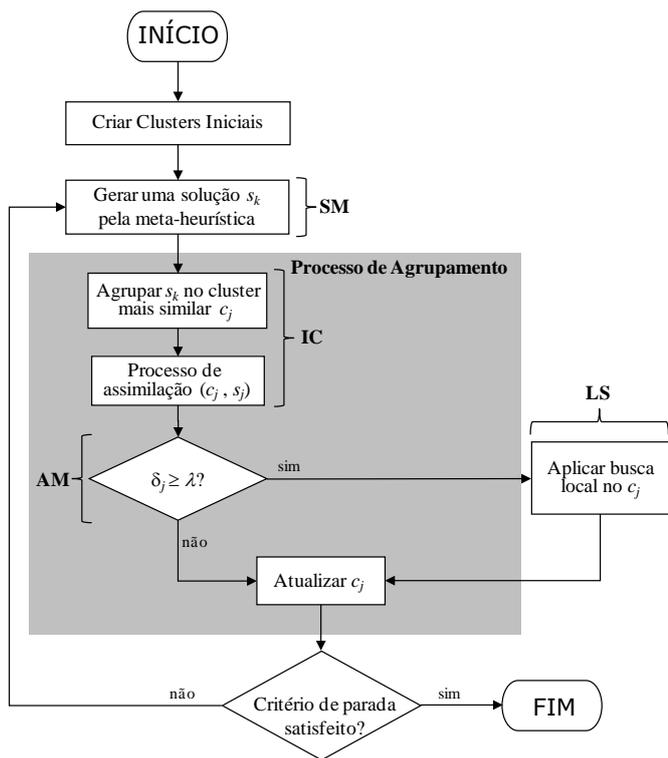


Fig. 1: Componentes do CS
 Fonte: [19]

O componente SM pode ser implementado por qualquer algoritmo de otimização que gera soluções diversificadas do espaço de busca. Ela trabalha com um

gerador de soluções, explorando o espaço de busca através de uma manipulação de um conjunto de soluções, de acordo com sua estratégia de busca específica.

O componente IC procura reunir soluções similares dentro de grupos, mantendo um centro de cluster representativo delas. Uma métrica de distância, Δ , é definida, a priori, permitindo uma medida de similaridade para o processo de agrupamento. Por exemplo, em otimização combinatorial, a similaridade pode ser definida como o número de movimentos necessários para alterar uma solução até o centro do *cluster* [20]. O processo de assimilação é aplicado no centro mais próximo c_i , considerando a nova solução gerada s_k . O método path-relinking [19] pode ser usado para gerar um série de pontos, memorizando o melhor ponto avaliado para ser o novo centro.

O componente AM examina cada cluster, em intervalos regulares, indicando um provável *cluster* promissor. Uma densidade de cluster, δ_j , é uma medida que indica o nível de atividade dentro do cluster j . Para simplificar, δ_j pode contar o número de soluções geradas pelo SM e agrupada em c_j . δ_j obtém um número exato *threshold*, λ , que é um valor que atingido pela contagem em δ_j , sinaliza que o cluster pode então ser melhor investigado para acelerar o processo de convergência.

Por último, o componente LS é um módulo de pesquisa interno que provém a exploração de um região supostamente promissora, representada por um *cluster*, intensificando a busca nessa região.

IV. CLUSTERING SEARCH APLICADO AO PROBLEMA DE ROTEAMENTO PERIÓDICO DE VEÍCULOS PARA COLETA DE LIXO

Para aplicar o método CS ao PRPV para coleta de lixo, fez-se necessário desenvolver uma representação para soluções possíveis.

Soluções para o problema foram representadas através de uma matriz, dividindo-se as rotas para cada dia em um vetor, e dividindo-se cada vetor em números aos quais aqueles que são negativos representam os veículos e os números seguintes a ele até o próximo negativo representam sua rota. Foi armazenado também um vetor que determina a frequência a ser usada de acordo com a demanda de cada cliente.

A Fig. 2 exibe um exemplo de solução para um PRPV com 2 veículos atendendo a 5 pontos:

DIA	ROTA					
1	-1	2	4	-2	1	3
2	-1	3	5	1	-2	
3	-1	1	4	-2		
4	-1	-2	1	4		
5	-1	2	3	-2	1	
6	-1	2	3	1	-2	4

Clientes	$c1$	$c2$	$c3$	$c4$	$c5$
Frequência	6	3	4	4	1
Agenda	19	13	17	18	2

Fig. 2: Exemplos de representação

A Fig. 2 é uma representação de um problema com 2 veículos e 5 clientes. No qual, no primeiro dia da semana, o veículo 1 visita o cliente 2, em seguida o cliente 4 e vai, então, para o depósito e após isso para a garagem. O segundo veículo visita o cliente 1, em seguida o 3 e então se dirige ao depósito e garagem. Nos dias seguintes, a representação segue de maneira análoga.

A função objetivo do PRPV (11) é a soma total da distância do ponto inicial (garagem) ao primeiro cliente, da distância entre cada cliente da rota, e por fim, da distância de retorno ao ponto inicial (do último cliente até o depósito e deste até a garagem). No algoritmo são determinadas penalidades que são somadas as soluções que não satisfazem alguma das restrições impostas pelo problema (peso além da capacidade de cada veículo ($\alpha * Ep$) ou tempo total da rota acima do permitido ($\beta * Et$)). Representa-se a distância inicial por D_i , a distância do cliente i ao cliente seguinte(j) por D_{ij} e a distância de retorno por D_r .

$$\sum_i (D_i + D_{ij} + D_r + (\delta * Ep) + (\beta * Et)) \quad (11)$$

O método CS começa com um número fixo de *clusters* ($NC=10$) que são inicializados com soluções aleatórias. A metaheurística *Simulated Annealing* (SA) [17] foi responsável pela geração de soluções que são agrupadas no CS. Neste caso, foi usado o algoritmo implementado por [15]. Usamos a solução s_k do SA a cada n (número de clientes) vizinhos para aumentar a diversidade entre as soluções enviadas para o processo de agrupamento.

Em cada iteração do CS, uma solução s_k é agrupada no cluster mais próximo C_i , que é o cluster que minimiza a distância entre a solução e o centro do cluster. O volume δ_j é acrescentado em uma unidade e o centro c_j deverá ser atualizado com novos atributos de s_k (processo de assimilação).

O processo de assimilação usa o método *Path-relinking*. O procedimento inicia-se através do cálculo da diferença entre o centro c_j e a solução s_k , $\Delta(c_j, s_k)$, ou seja, o conjunto dos movimentos necessários para atingir o s_k de c_j . O caminho das soluções é gerado, ligando c_j e s_k . Em cada passo, o procedimento move-se e examina todos os movimentos $m \in \Delta(c_j, s_k)$ da solução corrente s e seleciona aquele com melhor custo, aplicando então o melhor movimento a solução s . O conjunto de movimentos disponíveis é atualizado. O procedimento termina quando 50% das soluções no caminho foram analisados. O novo centro c_j é a melhor solução ao longo do caminho. Neste trabalho, um movimento é trocar uma agenda de c_j por uma agenda de s_k , mudando a rota do cliente com a nova programação.

Depois de realizar a assimilação, é preciso realizar uma análise do volume δ_j , verificando se este *cluster* pode ser considerado promissor. Um *cluster* é considerado promissor quando seu volume atinge o limiar λ . O valor de λ foi definido como 5.

O método 2-Opt [9] é implementado como uma busca local do CS, intensificando a busca na vizinhança de um *cluster* promissor C_j . O 2-Opt consiste em 2-mudanças ao longo de uma rota, excluindo dois arcos e substituindo-os por outros dois arcos para formar uma nova rota. Este método continua enquanto há melhoria na rota através deste movimento. O centro c_j é atualizado se a nova solução é melhor do que o anterior.

IV. RESULTADOS COMPUTACIONAIS

Os testes computacionais foram executados em um PC Intel Core i5 de 2.67 GHz com 2 GB de memória RAM.

Nos testes utiliza-se a instância proposta por [1], sendo a representação real da coleta de lixo do município de Ponte de Lima em Portugal. Esta instância contém: cinco veículos, cada qual com sua capacidade e tempo máximo previsto para utilização; uma garagem, de onde partem os veículos para seguirem suas rotas e depois depositarem o lixo no depósito que estiver mais próximo; dois depósitos, a serem visitados no final da rota e antes dos veículos retornarem à garagem; e 51 clientes, cada um com sua exigência de quantidade de lixo a ser recolhida e frequência específicas, os quais serão visitados no máximo seis dias na semana.

No segundo conjunto de testes, utiliza-se instâncias clássicas para o PRPV, introduzidas por [7], que tem características similares com a instância proposta por [1]. Cada veículo contém uma capacidade máxima e uma duração máxima da rota, uma garagem que também serve como depósito, e cada cliente tem uma demanda e frequência específica e será visitado em um número fixo de dias na semana.

Os resultados computacionais obtidos pelos métodos (CS e SA) são apresentados nas Tabelas 2 e 3, que possui as seguintes colunas: nome da instância testada, valor em quilômetros da solução utilizada na prática atualmente, os dados das instâncias (número de clientes, depósitos, veículos e dias), a melhor solução encontrada (coluna Melhor), solução média (coluna Média), o desvio percentual (coluna Dev) e o tempo computacional (coluna Tempo) em segundos. O desvio percentual é calculado como $Dev = 100\% (Média - Melhor) / Melhor$.

TABELA II RESULTADOS COMPUTACIONAIS DOS MÉTODOS CS E SA

Instancia	Solução corrente	Dados				CS			SA			Tempo(s)
		Clientes	Depósitos	Veículos	Dias	Melhor	Média	Dev (%)	Melhor	Média	Dev (%)	
<i>Ponte de Lima</i>	2389	51	2	5	6	1680,80	1729,84	2,92	1748,10	1852,08	5,95	147,89

Os resultados mostraram a eficácia do CS. Para a instância, o CS encontrou a melhor solução, reduzindo a distância percorrida em 30% em relação à solução atual adotada no município. Além disso, cabe ressaltar que o número de rotas também foi reduzido, na solução atual são necessárias 26 rotas enquanto que na solução apresentada pelo CS existem 25 rotas.

Nós podemos observar que o SA sem o processo de agrupamento do CS obteve soluções piores em termos de qualidade que o CS (a melhor solução encontrada pelo CS é cerca de 1% melhor que a melhor solução encontrada pelo SA).

O algoritmo CS foi robusto, obtendo soluções médias próximas das melhores soluções de acordo com o desvio percentual. Os tempos computacionais dos algoritmos foram competitivos, obtendo boas soluções em um tempo razoável.

A Tabela 3 reporta uma comparação com as melhores soluções encontradas por [7](coluna CGL), que propôs um algoritmo de Busca Tabu. O CS apresentou uma solução melhor para a instância PR01. E para as outras instâncias, as soluções obtidas pelo CS foram próximas a melhor solução da Busca Tabu. Nós podemos observar que o desvio médio da melhor solução da Busca Tabu foi 8,4%.

TABELA III RESULTADOS COMPUTACIONAIS DAS INSTÂNCIAS DE CORDEAU

Instancia	CGL	Tempo(s)	Dados				CS			SA			Tempo(s)
			Clientes	Depósitos	Veículos	Dias	Melhor	Média	Dev (%)	Média	Média	Dev (%)	
<i>PR01</i>	2234,23	231,6	48	1	2	4	2216,68	2295,31	3,55	2313,51	2404,44	7,62	163,51
<i>PR02</i>	3836,49	587,4	96	1	4	4	3865,33	4188,51	8,36	4058,06	4400,92	14,71	435,38
<i>PR03</i>	5277,62	1117,2	144	1	6	4	5773,70	6197,19	7,33	5983,94	6409,58	21,45	966,85
<i>PR04</i>	6072,67	1694,4	192	1	8	4	6855,78	7160,21	4,44	7027,29	7365,42	21,49	1897,86
<i>PR05</i>	6769,80	2197,2	240	1	10	4	7329,56	7577,34	3,38	7469,34	7746,62	14,43	2904,54
<i>PR06</i>	8462,37	3058,2	288	1	12	4	9323,27	9986,68	7,12	9651,80	10363,50	22,47	3485,45
<i>PR07</i>	5000,90	577,8	72	1	3	6	5171,39	5256,42	1,64	5416,63	5486,81	9,72	326,54
<i>PR08</i>	7183,39	1443	144	1	6	6	7956,61	8117,48	2,02	8139,61	8313,64	15,73	934,43
<i>PR09</i>	10507,34	2739	216	1	9	6	11706,61	12204,50	4,25	12092,93	12626,41	20,17	2614,09
<i>PR10</i>	13629,25	4293,6	288	1	12	6	14575,61	15286,48	4,88	14852,95	15585,71	14,35	3248,73
<i>Média</i>	6897,41						7477,45	7827,01	4,70	7700,61	8070,31	16,19	

A Fig. 3 ilustra uma distribuição dos tempos de execução, ou *time-to-target (TTT) plot*, para a instância de Ponte de Lima. O experimento consiste em executar o SA e o CS 100 vezes sobre a instância. Cada execução é independente da outra e interrompe quando uma solução com custo menor ou tão boa quanto o valor de uma solução alvo é encontrada. Nós observamos visualmente que o CS dominou SA.

Por exemplo, a probabilidade do CS encontrar uma solução menor ou tão boa quanto um valor alvo em menos de 10 segundos é aproximadamente 40%, em 20 segundos, 70% e em 30 segundos é 90%. Todas as soluções do SA foram encontradas com mais de 30 segundos.

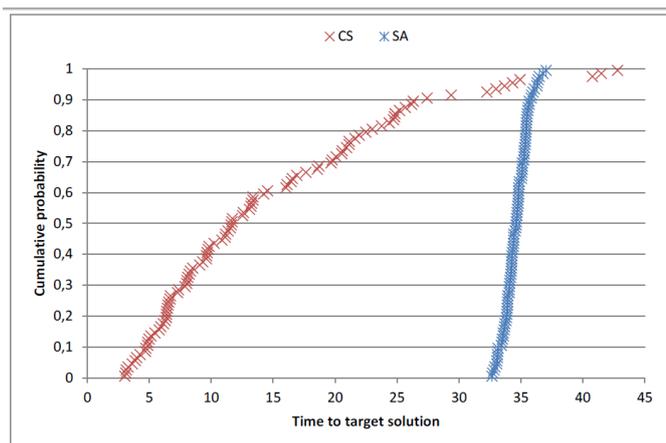


Fig. 3 – Distribuição da probabilidade cumulativa

VI. CONCLUSÃO

Os resultados obtidos mostram que modelar o problema de coleta de lixo como um PRPV e resolvê-lo com o método *Clustering Search* (CS) é uma iniciativa extremamente válida, visto que levou a resultados melhores que os previamente conhecidos para o problema tratado. Além de resultados finais 30% melhores, o que possibilita reduzir significativamente os gastos do processo real, o tempo de cálculo foi reduzido consideravelmente em relação à abordagem realizada anteriormente a este trabalho, que foi realizada por [1] e que é adotada no município de Ponte de Lima.

Como trabalhos futuros, propõe-se implementar uma nova estrutura de solução ao invés da relatada neste trabalho, visando melhorar os tempos computacionais. Pretende-se também propor um novo método de agrupamento de soluções e novas metaheurísticas geradoras de soluções, como Busca Tabu e Algoritmos Genéticos.

REFERÊNCIAS

- [1] Bianchi-Aguiar, T.; Carravilla, M. A.; Oliveira, J. F. (2012), Municipal waste collection in Ponte de Lima, Portugal – A vehicle routing application. *OR Insight*, 25, 185-198
- [2] Chaves, A.A.; Lorena, L.A.N. (2010) Clustering search algorithm for the capacitated centered clustering problem. *Computers & Operations Research*, v. 37, p. 552-558.
- [3] Christofides, N.; Beasley, J. E. (1984), The Period Routing Problem, *Networks* 14, 237 – 256.
- [4] Christofides, N; Mingozzi, A.; Toth, P. (1979), The vehicle routing problem. In Christofides, Mingozzi, Toth, Sandi, editors. *Combinatorial Optimization*, 315-338, John Wiley & Sons.
- [5] Carvalho, D. B.; Oliveira, G. A.; Souza, M. J. F. (2003), Método de pesquisa em vizinhança variável aplicado à resolução do problema de roteamento de veículos, *XXXV SBPO*, 676-684.
- [6] Coelho, M. A.; Soares, L. T. (2001), Geografia geral: o espaço natural e socioeconômico. Moderna, 4ª ed., 427.

- [7] Cordeau, J.F.; Gendreau, M.; Laporte, G. (1997), A tabu search heuristic for periodic and multi-depot vehicle routing problems. *Networks*, 30, 105–119.
- [8] Croes, G. (1958), A method for solving traveling salesman problems. *Operations Research*, 6, 791–812.
- [9] Cunha, V.; Caixeta F.J.V. (2002), Gerenciamento da Coleta de Resíduos Sólidos Urbanos: Estruturação e Aplicação de Modelo Não-Linear de Programação por Metas, *Gestão & Produção*, 9(2):143–161.
- [10] Feo, T. A.; Resende, M. G. C. (1995), Greedy randomized adaptive search procedures. *Journal of Global Optimization*, v. 6, 109-133.
- [11] Glover, F. (1986), Future paths for integer programming and links to artificial intelligence. *Computers and Operations Research*, 5, 553-549.
- [12] Glover F.; Laguna M.; Martí, R. (2000). Fundamentals of scatter search and path relinking. *Control and Cybernetics*, 39, 653-684.
- [13] Golden, B. L.; Chao, I. M.; Wasil, E. (1995), An Improved Heuristic for the Period Vehicle Routing Problem. *Networks*, 25-44.
- [14] Gonçalves, L. B.; Ochi, L. S.; Martins, S. L. (2005), Heurísticas GRASP para um Problema de Roteamento Periódico de Veículos. Acesso em: www.ic.uff.br/PosGraduacao/Dissertacoes/274.pdf. Access: 04/29/2012.
- [15] Higino, W.; Araujo, E.; Marcel, H.; Poldi, K.; Chaves, A.A. (2012) Metaheurísticas Simulated Annealing e Pesquisa em Vizinhança Variável aplicadas ao Problema de Roteamento Periódico de Veículos para Coleta de Lixo. In: XVI CLAIO / XLIV SBPO, Rio de Janeiro
- [16] Holland, J. H. (1975), *Adaptation in natural and artificial systems*. Michigan: University of Michigan Press, 211.
- [17] Kirkpatrick, S.; Gelatt, D. C.; Vecchi, M. P. (1983), *Optimization by simulated annealing*. *Science*, v. 220, n. 4598, 671-680.
- [18] Oliveira A. C. M.; Lorena, L. A. N. (2007), Hybrid Evolutionary Algorithms and Clustering Search. *Hybrid Evolutionary Systems: Studies in Computational Intelligence*, n. 75, 81-102.
- [19] Oliveira, A. C. M.; Chaves, A. A.; Lorena, L.A.N. (2013), Clustering Search. *Pesquisa operacional*, v.33 (1), 1-17.
- [20] Toth, P.; Vigo, D. (2002), *The vehicle routing problem*. Society for Industrial and Applied Mathematics. 367 pp.