GRASP Aplicado à Distribuição de Veículos: Uma Visão Computacional

Carlos Louro[1], Vânia Machado[2], Manuel Leite[3], João Negreiros[4]

louro.carlos@sapo.pt, vbessamachado@gmail.com, mcostaleite@clix.pt, c8057@isegi.unl.pt

Universidade Lusófona, Campo Grande 375, 1749-024, Lisboa, Portugal

Resumo: Demonstração da técnica de Investigação Operacional GRASP na determinação do número mínimo de contentores a serem utilizados no transporte, via marítima, de um conjunto aleatório de veículos Volkswagen (considerou-se nove modelos). Assim, é introduzido a técnica GRASP na perspectiva de construção de soluções por um critério de optimização combinatória. Discussão da sua implementação técnica e respectiva análise dos resultados obtidos adereçam a segunda parte deste artigo.

Palavras-chave: Investigação Operacional, Computação, Metaheurística, GRASP, Optimização Combinatória.

1. Introdução

A Investigação Operacional (IO) é a ciência que utiliza modelos matemáticos, estatísticos e algoritmos computacionais na resolução de problemas complexos mas sempre com o objectivo de melhorar ou optimizar a sua melhor solução. De acordo com Alvarez-Valdes [2006], IO significa investigação das operações, ou seja, uma tomada de decisões utilizando uma abordagem científica. Desde os anos 50, o avanço tecnológico possibilitou o desenvolvimento do ramo da ciência computacional/optimização combinatória e que apresenta como estudo base a maximização/minimização da função objectivo, associada a um conjunto de restrições. Actualmente, os modelos são abrangentes bem como os seus algoritmos de obtenção de soluções aproximadas, em detrimento da procura de soluções óptimas e exactas.

De facto, existem várias classificações possíveis para problemas de optimização e, algumas delas, apresentam métodos exactos e deterministicos em termos de resolução. Outras levam à necessidade de métodos não exactos (heurísticas), uma vez que a sua formulação e/ou resolução exacta levaria a uma complexidade intratável. No entanto, as heurísticas são reconhecidas como soluções limitadas e fornecem sempre o mesmo desfecho quando iniciadas do mesmo ponto de partida. Logo, surge a necessidade de heurísticas mais eficazes, flexíveis e dinâmicas, que se adaptem facilmente a estruturas paralelas direccionadas à optimização de um problema. Formalmente, essas heurísticas são designadas Metaheurísticas (geram procedimentos internos de modo a explorar inteligentemente a instância do problema e o seu espaço de soluções). Segundo Alvarez-Valdes [2006], uma das técnicas mais importantes na obtenção deste tipo de soluções aproximadas denomina-se Greedy Randomized Adaptive Search Procedures - GRASP.

GRASP é uma metaheurística que permite a construção de soluções melhoradas e cuja aleatoriedade é utilizada intensamente, mas levando em conta critérios de optimização relativos á solução parcial. Na prática, um procedimento GRASP faz a amostragem em pontos do espaço de soluções por um critério de optimização. A sua estratégia é diminuir a distância percorrida entre soluções iniciais e óptimos locais para que se possa descobrir uma maior quantidade de óptimos locais dentro de toda a vizinhança a explorar. Á medida que a complexidade dos problemas crescem, o respectivo planeamento, controlo e procura de sequências realizáveis com uma exploração eficiente de recursos escassos torna-se uma tarefa de investigação estimulante e atractiva. A determinação da melhor solução em problemas de optimização em conjuntos, é aqui explorado e o objectivo passa por, dado um conjunto de restrições económicas, determinar o número mínimo de recursos a serem utilizados de modo a minimizar a função objectivo. Propõem-se, assim, a utilização do GRASP na determinação da melhor solução, para o seguinte problema de optimização combinatória.

2. Caso de Estudo

No transporte de automóveis (considerou-se nove modelos da marca Volkswagen), surge a necessidade imperiosa de exportar veículos para outros países. Em virtude da distância e da localização geográfica do porto de origem e destino, o transporte de veículos é efectuado, normalmente, por cargueiros contentores. Todavia, os veículos a serem transportados têm diversos pesos e tendo como restrição primária o limite da capacidade de cada um dos contentores. Considerou-se que a capacidade de peso de cada contentor rondaria os 16000 Kg. Assim, o problema consiste na distribuição de n veículos de forma a minimizar o número m de contentores a utilizar.

Constata-se que se trata de um problema com uma forte componente combinatória. Quantificada a proximidade de um determinado número de contentores, utiliza-se a metaheurística GRASP cujo objectivo é gerar procedimentos de procura de vizinhanças que proporcionem melhores soluções. Considerou-se, como segunda restrição, que os n veículos apresentam diferentes pesos.

2.1. BinPacking

Este problema, em particular, espelha-se na abordagem do Bin Packing unidimensional cujo objectivo é determinar o menor número de contentores de igual capacidade e onde a soma dos pesos dos n veículos nunca é superior à capacidade do contentor. Todavia, este problema também poderá ser analisado de uma forma não combinatória, não havendo qualquer estudo na alocação dos veículos (distribuir os n veículos pelos contentores até a respectiva capacidade do contentor ser preenchida). Contudo e muito provavelmente, o resultado gerado estaria longe da

solução óptima, o que levaria a um custo maior no transporte dos veículos. Assim, ao se aplicar a este tipo de problema métodos algorítmicos que nos permitam optimizar soluções, poder-se-á diminuir os custos e o tempo de estudo no envio dos mesmos, com o objectivo de minimizar os custos e maximizar os proveitos. [Bouleimen, K., Lecocq, 2003]

Formalmente, este problema consiste nos seguintes três itens: (1) Transporte de *n* veículos da marca Volkswagen por *m* contentores. (2) A cada veiculo está associado um peso e a cada contentor uma capacidade (16000Kg). (3) O objectivo é automatizar a distribuição dos veículos de modo a minimizar o número de contentores a utilizar.

$$\left\lceil \sum_{i=1}^{n} \frac{Peso\,[i]}{m} \right\rceil = \delta\ Contentores$$

Formula 1 – Peso[i] representa o peso de cada automóvel, *m* indica a capacidade de peso de cada contentor, *n* significa o número de carros a transportar e *δ* revela o número de contentores necessários à solução do problema.

Já anteriormente referido, este problema de optimização tem uma forte componente combinatória, pelo facto de existir uma grande variedade de combinação na distribuição dos veículos nos respectivos contentores, o que justifica a utilização de metaheurísticas. Assim, determinar a solução óptima significa verificar qual o menor número de contentores a utilizar (representado por δ na formula acima). Mais, terá de se verificar que δ é uma solução possível de se alcançar.

Considerando estas restrições, a colocação dos veículos nos contentores será aleatória, não havendo qualquer estudo prévio na distribuição dos mesmos. Esta primeira solução heurística obtida (solução inicial) representa o resultado da colocação dos *n* veículos nos respectivos *m* contentores. Note que esta solução é sempre a mesma quando iniciada do mesmo ponto de partida. Utilizando heurísticas mais flexíveis, é possível criar soluções que se adaptem mais facilmente a estruturas paralelas direccionadas à optimização global de um problema, através de diferentes procedimentos de procura local.

O GRASP procura a obtenção de soluções aproximadas com o objectivo de indicar quais os veículos que serão distribuídos nos determinados contentores mas satisfazendo a restrição de capacidade. Segundo Bouleimen e Lecocq [2003], o GRASP divide-se em duas fases: Construção e pesquisa local. Na fase de construção, a solução inicial é construída elemento a elemento tendo em conta o peso dos veículos e a sua aleatoriedade. Assim, (1) atribui-se a cada veículo, passível de inserção o valor do seu peso, (2) ordena-se os veículos por ordem crescente ou decrescente conforme o valor da função objectivo (neste caso particular, ordena-se decrescentemente em virtude da minimização da função objectivo) e (3) insere-se sucessivamente os veículos, por ordem decrescente, até que se obtenha uma solução inicial e viável.

Trocando-se este determinismo da escolha ordenada por uma aleatoriedade construída veículo a veículo, considerou-se um conjunto de veículos aleatórios designada Lista Restrita de Candidatos (LRC). Obtida a LRC, surge a necessidade de se poder limitar a variação de escolha, permitindo um maior rigor na solução encontrada. Quanto menor a variação, maior o rigor da solução obtida. Assim, a variação da LRC poderá variar entre um (totalmente rigoroso) e o número de veículos a colocar (totalmente aleatório). Deste modo, a fase de construção resume-se a: (1) Colocar num determinado contentor um veículo da LRC. (2) Construir a melhor solução, sendo a variação rigorosa com alguma aleatoriedade. (3) Distribuir todos os veículos pelos contentores.

A partir da solução inicial gerada, apresenta-se a fase de pesquisa local (explorar a vizinhança até encontrar um óptimo local). Note que nesta fase não são permitidas soluções piores que a inicial. A estratégia do método é, portanto, diminuir a distância percorrida entre soluções iniciais e óptimos locais para que se possa descobrir uma maior quantidade de óptimos locais dentro de toda a vizinhança [Viana e Sousa, 2000, Sprecher e Drexl, 1998].

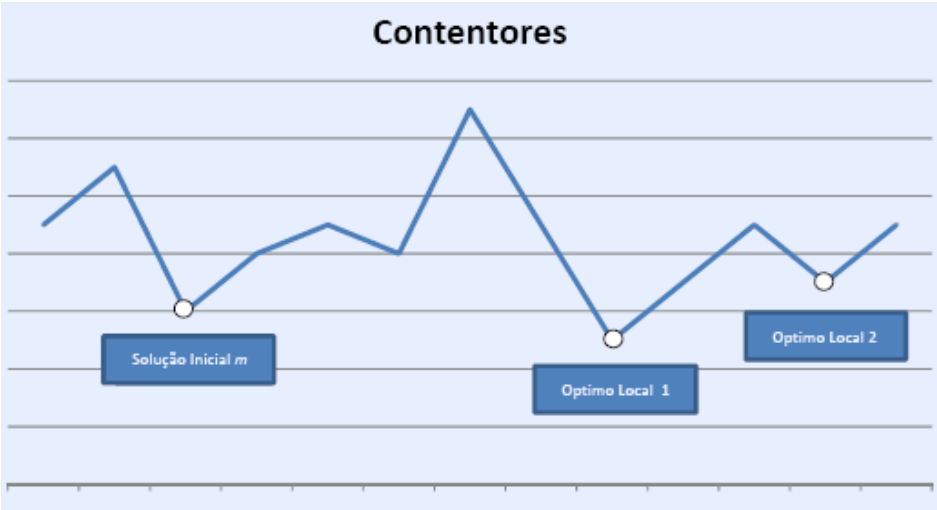


Figura 1 – O GRASP não permite soluções piores que as iniciais.

Como ilustra a figura acima, dada uma solução inicial *m*, o procedimento GRASP só considera na sua vizinhança um óptimo local *m-n*, $\forall$ n $\geq$ 1. No entanto, partindo de uma solução inicial *m*, o procedimento nunca considera soluções piores que a inicial. Assim, o único óptimo local a considerar é somente o óptimo *local 1*, descartando todas as outras soluções piores que a solução inicial como, por exemplo, o óptimo *local 2*. A principal vantagem que o GRASP admite baseia-se na forma totalmente aleatória em que trabalha e na independência entre cada iteração. O critério de paragem poderá ser determinado pelo número de iterações do processo ou por um número de iterações pretendido.

´ Ao invés, o Simulated Annealing (SA), por exemplo, é uma metaheurística para optimização combinatória que consiste num procedimento de procura local probabilística, possibilitando soluções piores que as iniciais e atribuindo as mesmas probabilidades locais.

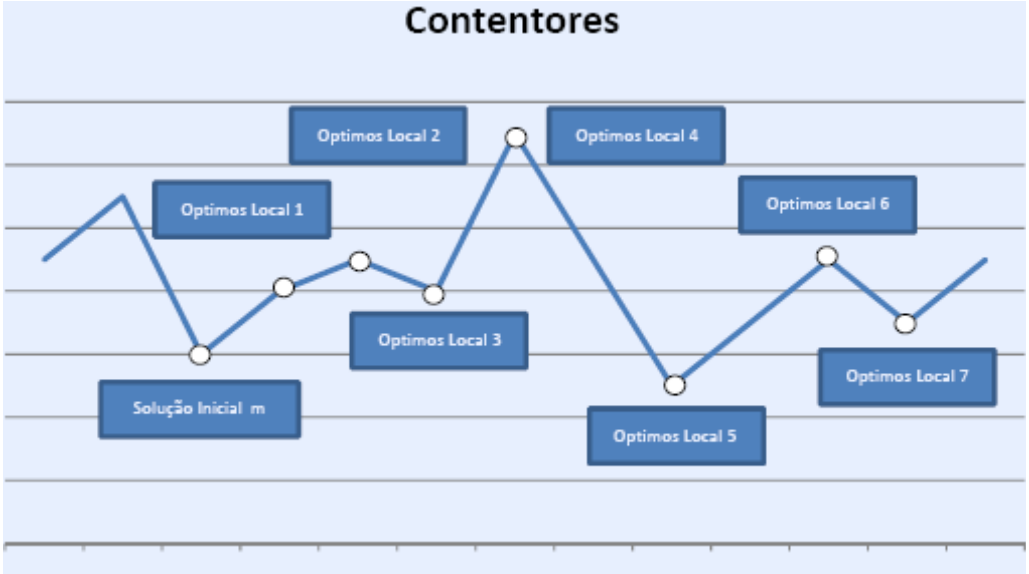


Figura 2 – O SA permite soluções piores que as iniciais.

Como se ilustra na figura acima, dada uma solução inicial *m*, a técnica SA consiste num procedimento de procura local mas possibilitando soluções piores que a inicial e ao qual atribui uma probabilidade de melhor solução. Por exemplo: Considere na sua vizinhança, um óptimo local *m-n* tal que $\forall$ $n \geq 1$, mas atribuindo uma probabilidade a cada solução encontrada. Assim, partindo da solução inicial *m*, é determinado o óptimo *local 1* cuja solução é pior que a inicial. No entanto, esta solução parcial é considerada, sendo atribuído uma probabilidade de melhor solução. Na procura de um novo óptimo local, este processo sempre considera a solução anterior. Determinado agora o óptimo *local 2*, ao mesmo é atribuído novamente uma probabilidade de melhor solução. Este procedimento repete-se sucessivamente até se determinar a melhor solução para o problema. De um modo genérico, o procedimento SA considera óptimos locais com soluções piores que a inicial, com o objectivo de a partir desta determinar óptimos locais com soluções melhores que a inicial. Talvez mais eficaz que a técnica GRASP na obtenção de soluções óptimas mas menos eficaz no período de tempo da obtenção das mesmas. [Bouleimen e Lecocq, 2003].

3. Implementação Computacional

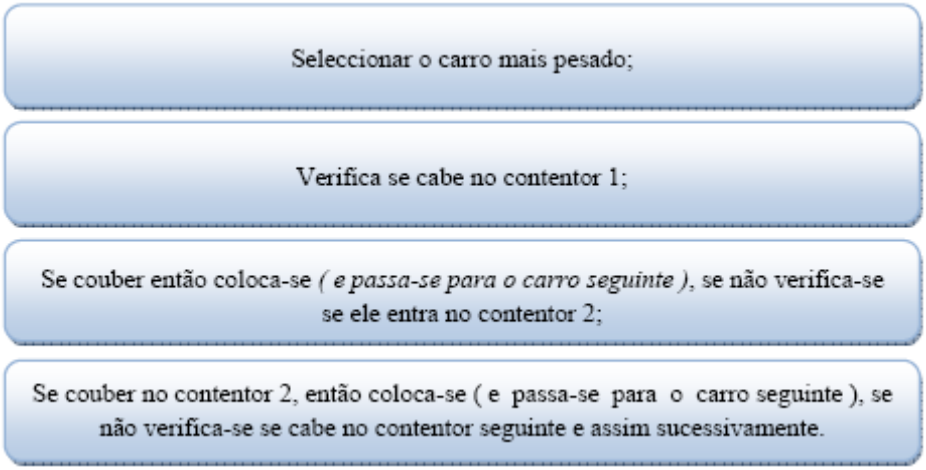
A execução deste algoritmo foi efectuada em MATLAB® 7. Foi criada uma interface onde possibilita o utilizador, entre outros, indicar o número dos modelos dos veículos a serem transportados. A interface permite também a visualização gráfica da melhor solução encontrada bem como o espaço ocupado e disponível dos contentores. A mesma apresenta ainda estatísticas

finais que permitem visualizar o tempo dispendido por cada processo, a média do peso por contentor, a média do peso por carro e o custo total do transporte.

Em suma, o algoritmo GRASP apresenta o seguinte procedimento: (1) Para i=1 até N, construir a solução inicial e realizar a pesquisa na vizinhança da solução inicial (actualizando a solução sempre que a vizinha for melhor). (2) Devolver a melhor solução encontrada.

3.1. Método de Construção e Procura Local

Já referido anteriormente, a implementação do problema começa pela construção da solução inicial. Portanto, começa-se por ordenar os veículos por pesos o que significa, por exemplo, que o veiculo mais pesado fica com o número 1 e o veiculo mais leve com o número *n-m* (último número). Ordenados os pesos, elabora-se o seguinte processo:



Numa visão sumária, tem-se: Seleccionar veículo a veículo e verificar se há espaço para todos os *n* veículos serem arrumados nos contentores. Sempre que um veículo permita ser colocado num contentor, o espaço disponível no contentor é actualizado e sucede-se para o veículo seguinte. Naturalmente, realiza-se este processo até não haver mais automóveis. Construída a solução inicial, utiliza-se a heurística GRASP de modo a gerar a segunda solução inicial.

Computacionalmente, sempre que se estuda a colocação de um veículo num contentor cria-se uma lista de veículos possíveis (Lista Restrita de Candidatos) a serem colocados. Por exemplo: Quando se inicializa este procedimento, em vez de se seleccionar o primeiro veículo (o mais pesado), selecciona-se os primeiros dez carros e garante-se que qualquer um destes é uma alternativa viável para ser alocado no contentor. Desta forma, a escolha de um dos veículos é aleatória. Todavia, o tamanho da LRC é um factor muito importante que influencia todo o procedimento em causa. Se o tamanho da LRC for 1, então está-se na presença de um processo totalmente rigoroso (não existindo qualquer aleatoriedade na distribuição dos veículos), o que originaria soluções sempre iguais. Caso o tamanho da LRC seja demasiado elevado, tem-se um processo completamente aleatório na construção de soluções iniciais. No entanto, o que se

pretende é um processo mais inteligente, ou seja, algo que se adapte à quantidade de veículos que se está a colocar.

Em primeiro lugar, é fundamental determinar uma percentagem do número de veículos a colocar nos contentores e, em segundo lugar, calcular o máximo entre essa percentagem e o tamanho da LRC. Na realização deste projecto, foi considerado 10% como o valor da percentagem e 5 como o tamanho da LRC. Por exemplo: Imagine o leitor que tem 10 veículos para arrumar no contentor. Matematicamente, 10% de 10 é 1. No entanto, não se pretende que um seja o tamanho da LRC por não haver qualquer aleatoridade de escolha. Assim, assigna-se um máximo entre 1 e 5, no que se traduz do tamanho da LRC (5, e não 1). Considere agora que restam quatro veículos para arrumar num contentor. Uma vez mais, 10% de 4 é 0.4, isto é, zero. O máximo entre 0 e 5 será 5. Entretanto, a LRC não pode ter tamanho 5, pois não há 5 veículos mas apenas 4. Presumivelmente, determinou-se o mínimo entre 5 e o número de veículos por colocar (neste caso, 4).

A construção desta solução vizinha apresenta como objectivo principal melhorar a solução inicial. Para tal, escolhe-se um contentor aleatoriamente e retira-se todos os veículos encontrados dentro deste. A construção de uma nova solução vizinha será essencial em virtude do processo de retirada de veículos dos contentores não ter sido ainda concluído. Repare que apenas se analisou a retirada dos veículos, mas não a recolocação dos mesmos. Considerou-se, em termos de programação MATLAB®, que todos os contentores esvaziados e respectivos veículos foram guardados numa variável auxiliar designada carrosFora. Nesta fase computacional, é importante realizar o refresh dos dados, isto é, actualizar a variável do contentor agora vazio (novo volume total disponível) e os veículos que foram retirados (ficam momentaneamente por arrumar). Imagine, por exemplo, que o contentor esvaziado seria o número 10. Por vantagem de programação e não resultar qualquer prejuízo para o processo, o contentor 11 passa-se a chamar 10, o 12 passa-se a designar 11 e assim sucessivamente.

Nesta fase, selecciona-se cada viatura aleatoriamente e tenta-se efectuar a sua recolocação. Caso não seja possível colocá-los nos contentores que já estavam a ser utilizados, cria-se um novo contentor. Obviamente, uma vez que o objectivo é minimizar o número de contentores, o ideal é não criar um novo contentor. Construídas as soluções iniciais e as vizinhas, é possível agora programar a função principal do GRASP. Após o cálculo de uma determinada solução inicial, calculam-se várias soluções vizinhas, repetindo-se o processo N vezes ou até se encontrar uma solução que confira o mesmo número de contentores que a solução óptima. Neste caso particular, foi considerado o cálculo de 50 soluções vizinhas, repetindo-se o processo 500 vezes.

3.2. Exemplo

Considere, também, que o número de veículos a distribuir pelos contentores é 1000. O objectivo é determinar o número mínimo *m* de contentores a utilizar no transporte dos veículos. Em termos da seguinte interface, existe a possibilidade de seleccionar os *n* veículos por modelos ou por um número total de veículos. Caso a selecção seja efectuada por um número total de veículos, será atribuído um número de veículos a cada modelo aleatoriamente. Analogamente, os valores dos pesos são valores aleatórios que variam de modelo para modelo.

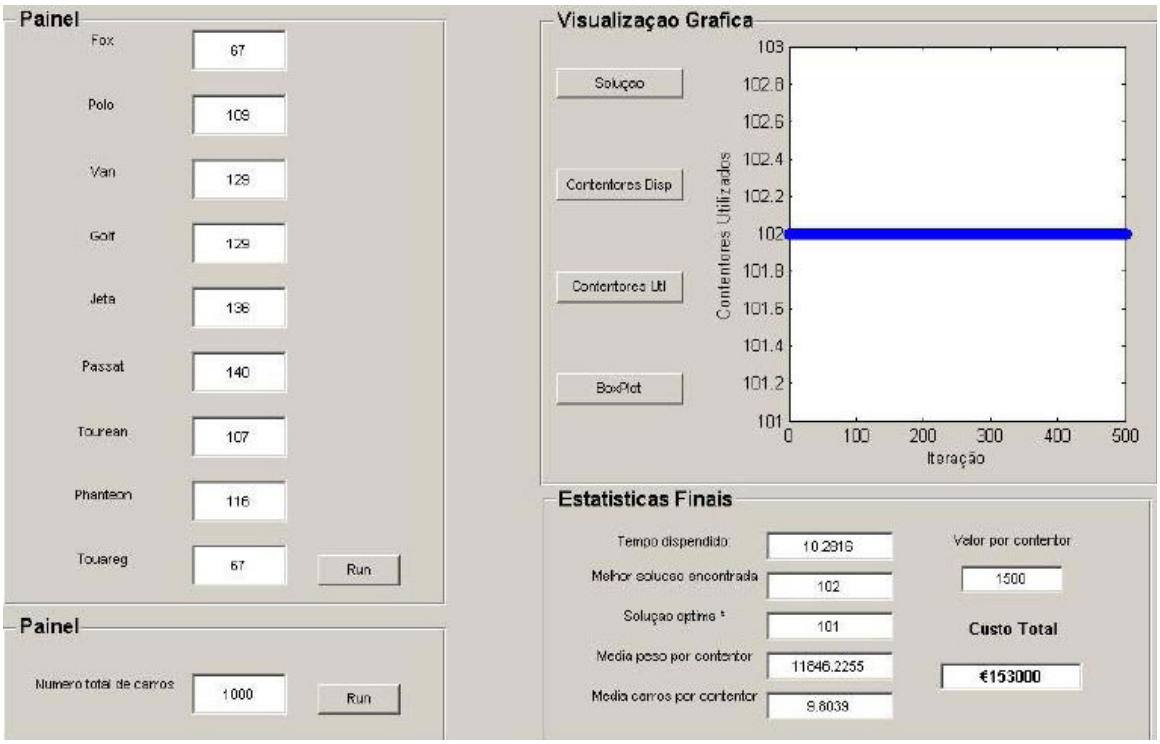


Figura 3 - Interface proposta para a técnica GRASP.

Após a introdução do número de veículos no respectivo painel, é possível visualizar a interpretação gráfica dos resultados. Os mesmos possibilitam ao utilizador da aplicação obter uma noção gráfica da distribuição dos veículos bem como o peso ocupado e disponível em cada contentor. Em termos de estatísticas finais, pode-se visualizar medidas interessantes. Resumidamente, são disponibilizados quatro gráficos. A figura 4 (esquerda) apresenta a melhor solução encontrada, isto é, o número mínimo de contentores a utilizar. Analisando-o, verifica-se que foram efectuadas 500 iterações e cuja solução foi sempre de 102 contentores. A figura à direita representa o respectivo peso por preencher para cada contentor. Assim, pode-se concluir que dos 102 contentores utilizados, só o ultimo é que ainda tem um peso significativo disponível. Todos os outros estão parcialmente ou totalmente preenchidos.

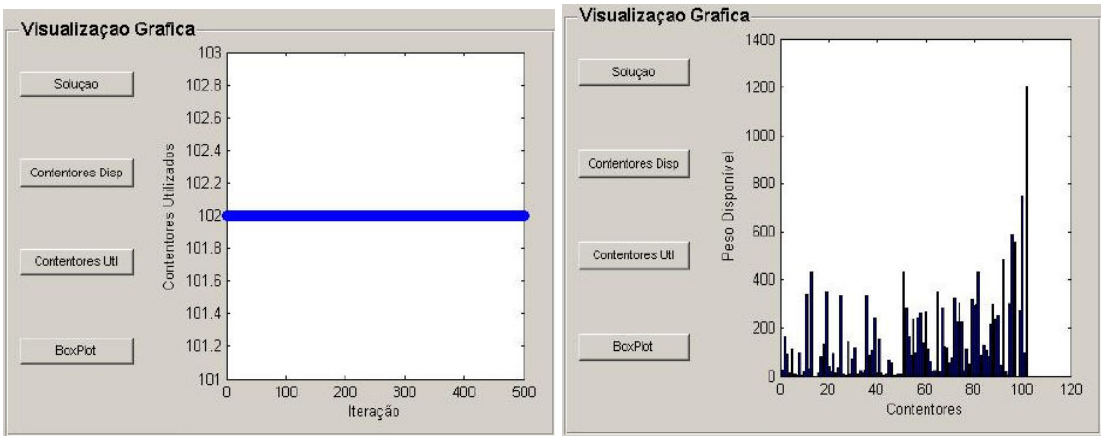


Figura 4 - Solução final em função do número de iteracções (esquerda) e representação gráfica do espaço disponível por cada contentor (direita).

Na figura 5 (esquerda), é disponibilizado a Box-Plot que possibilita a verificação do valor mediano de peso por contentor, 1.19×10^4 Kg, mas persistindo uma boa variação dos dados (os pontos fora das extremidades são outliers). Foi ainda implementada a visualização da estatística descriptiva com o objectivo de fornecer uma análise mais detalhada (figura 5, direita). Esta mostra qual a melhor solução, o tempo dispendido para cada processo iterativo, a melhor solução encontrada, qual a solução óptima do problema, a média do peso por cada contentor, a média dos carros por contentores e o custo total no transporte dos veículos.

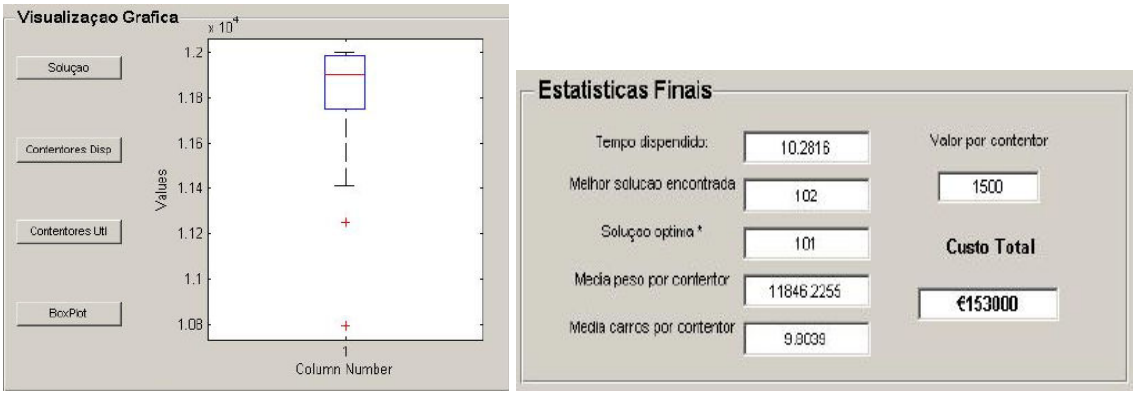


Figura 5 – Box-Plot do peso alocado por contentor (esquerda) e a respectiva estatística descriptiva (direita).

Todavia, ainda falta determinar qual a distribuição dos veículos pelos contentores. Assim, é criada um relação em Excel® que mapeia qual a distribuição dos veículos nos respectivos contentores (ver figura seguinte).

	A	B	C	D
1	Número	Peso	Modelo	Contentor
2	1	2049	Touareg	4
3	2	2011	Phanteon	10
4	3	2010	Touareg	2
5	4	1998	Phanteon	11
6	5	1992	Touareg	9
7	6	1985	Touareg	1
8	7	1944	Touareg	3
....	.....	.....	.....	.....
....	.....	.....	.....	.....
....	.....	.....	.....	.....
....	.....	.....	.....	.....
....	.....	.....	.....	.....
94	93	668	Lupo	5
95	94	643	Lupo	8
96	95	623	Lupo	4
97	96	616	Lupo	4
98	97	601	Lupo	3
99	98	598	Lupo	8
100	99	590	Lupo	6
101	100	588	Lupo	10
102	101	585	Lupo	6

Folha1 Folha2 Folha3

Pronto

Figura 6 - Distribuição dos veículos nos respectivos contentores

4. Conclusão

Esta pesquisa apresenta uma abordagem metaheurística a um problema real e existente em empresas de construção automóvel e companhias de transporte naval. Tendo sido implementada a técnica GRASP, esta abordagem resulta do facto deste problema em particular ter uma forte componente combinatória, de tal forma que uma abordagem deterministica poderia demorar semanas a encontrar a solução óptima.

É precisamente o binómio entre o tempo dispendido e a qualidade de solução a chave deste problema. Por um lado, pretende-se apresentar uma solução de qualidade em tempo útil. Por outro lado, quanto mais tempo se utilizar, mais possibilidades há-de melhorar essa mesma solução. É neste balanço que enceta esta metodologia pois tenta reconciliar uma solução de grande qualidade num factor de decisão fundamental para qualquer gestor.

Em termos computacionais, o GRASP mostrou flexibilidade do ponto de vista da implementação e adaptabilidade dos seus parâmetros neste tipo de problema. Deste modo, houve a necessidade de adequar os argumentos à dimensão do problema em questão, ou seja, quer se tenha um problema com 100 viaturas (pequena empresa) ou 1000 viaturas (grande importador de automóveis), existe uma necessidade imperiosa de a aplicação se adaptar convenientemente a este facto.

Foi também observado que se pode ajustar os parâmetros de modo a obter soluções iniciais de grande qualidade na primeira fase do processo. Contudo, o processo de pesquisa local (segunda fase) torna-se mais moroso à medida que pretende garantir que as soluções vizinhas eram melhores que a actual. Desta situação, resultou a decisão de permitir a geração de

melhores soluções iniciais, associado a uma pesquisa local menos intensa. Como se comprova pelos resultados, foi sempre garantida uma solução de grande qualidade em tempo reduzido, sendo que, nos casos em que não foi alcançado o limite inferior, foi obtida uma solução muito próxima.

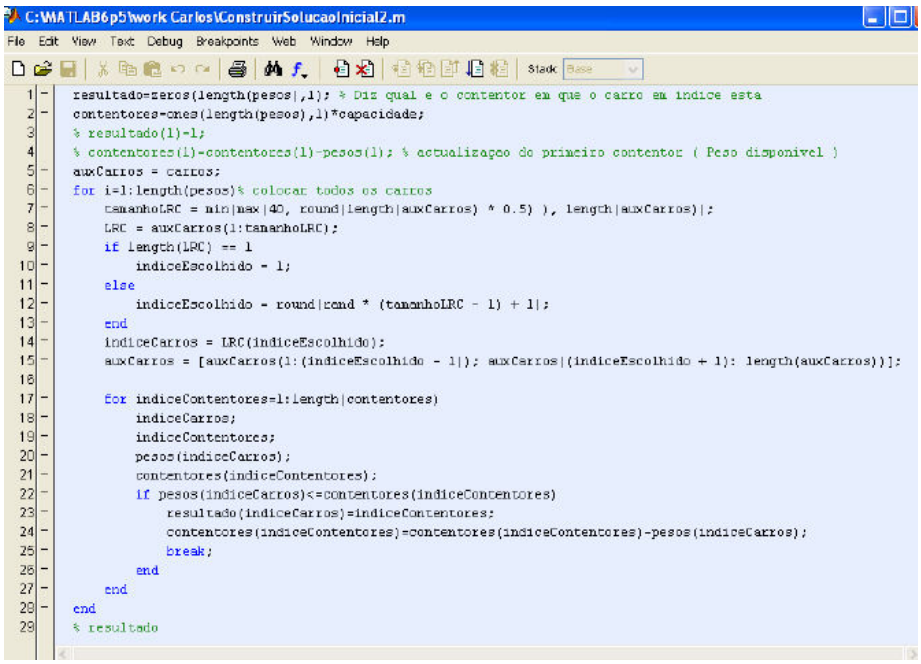


Figura 7 – Visão parcial do código programado (cerca de 400 linhas no seu total) em MATLAB®.

Por último, é gerado um ficheiro em Excel®, o qual é dirigido ao decisor da empresa (possibilidade de apresentar um plano de execução baseado em custos de transporte mais baixos). De facto, este output contém informação que é extremamente útil ao decisor, operador e controlador de viaturas para que estes possam tomar conhecimento e coordenar a correspondência entre viaturas versus contentores.

Bibliografia

[1] Alvarez-Valdes, R., Crespo, E., Tamarit, J. M., Villa, F. (2006). A scatter search algorithm for project scheduling under partially renewable resources.

[2] J. Blazewicz, J.K. Lenstra, A.H.G. Rinnooy Kan (1983). Scheduling subject to resource constraints: Classication and complexity. Discrete Applied Mathematics.

[3] Bouleimen, K., Lecocq, H. (2003). A new excient simulated annealing algorithm for the resource-constrained project scheduling problem and its multiple mode version. European Journal of Operational Research.

[4] Sprecher, A., Drexl, A. (1998). Solving multi-mode resourceconstrained project scheduling by a simple, general and powerful sequencing algorithm. European Journal of Operational Research.

[5] Viana, A., Pinho de Sousa J. (2000). Using methaheuristics in multiobjective resource constrained project scheduling. European Journal of Operational Research.

[6] Woo Kim, K., Gen, M. e Yamazaki, G, (2003). Hybrid genetic algorithm with fuzzy logic for resource-constrained project scheduling. Applied Soft Computing.